



Domótica aplicada a la prevención de enfermedades Proyecto de grado – Año 2016

Ingeniería en Informática

Proyecto de grado enfocado, por medio de un sistema domótico, a asistir a las personas en el manejo diario del control del estado de salud.

Natino, Miguel Carlos – Díaz Martínez, Facundo Adrián



Título: Ingeniero en Informática

Profesor Guía: Ing. Rivera Bernsdorff, Fernando

Miembros del Tribunal Evaluador:

Nombre:

Firma:

Nombre:

Firma:

Nombre:

Firma:

Fecha de exposición del Trabajo Final:



Dedicatoria:

Es difícil poder llegar a agradecer a todas las personas que me ayudaron y me dieron fuerzas, ganas y voluntad para poder cumplir con este objetivo.

En primer lugar, a mis Padres, mi hermana Ana, mis sobrinos Benjamín y Emma, por el apoyo diario, amor, comprensión, consejos y ayuda en momentos difíciles. Gracias a ellos soy lo que soy.

A mis amigos y compañeros, por todo el apoyo y ayuda que me brindaron. Cuando los necesite siempre estuvieron, y sin pedir nada a cambio, y eso lo valoro muchísimo. Por eso nunca los voy a olvidar.

A todo el personal de la Universidad, profesores, en especial a nuestro guía de tesis, Ing. Fernando Rivera Bernsdorff. Gracias por su apoyo, orientación y experiencias brindadas durante todo el cursado para poder llegar a este objetivo.

Le agradezco a mi amigo y compañero Facundo, con el que transitamos estos años de cursado, realizando prácticos, estudiando para parciales, finales, etc. Sos una gran persona y gracias por estar tanto en las buenas como en las malas. Te deseo lo mejor hoy y siempre.

¡Gracias Totales!

Miguel Carlos Natino

Dedico este proyecto a mi padre, quien incansablemente luchó hasta el final, y hoy gracias a él pude inspirarme para finalizar mi carrera.

A mi madre, quien siempre estuvo para darme todo el amor y el consejo necesario para guiarme durante mi vida.

A mi novia Magdalena quien me apoyo y alentó para llegar a esta instancia.

A mis hermanos quienes fueron un gran apoyo emocional durante el tiempo en que escribía esta tesis.

A mis amigos y compañeros que siempre estuvieron presentes en cada momento de mi carrera y de mi vida.

A nuestro profesor tutor Ing. Fernando Rivera Bernsdorff, quien con su guía y continuo apoyo permitió que este proyecto pueda ser finalizado y presentado.

A mis maestros quienes nunca desistieron al enseñarme, a ellos que continuaron depositando su esperanza en mí.

A todos los que me apoyaron para escribir y concluir esta tesis.

Para ellos es esta dedicatoria de proyecto de grado, es a ellos a quienes se las debo por su apoyo incondicional.

Facundo Adrián Díaz Martínez



Índice de Imágenes	9
Índice de Tablas	10
Abstract	11
1 - INTRODUCCIÓN.....	12
Breve descripción del problema y su importancia.....	14
Carácter del problema.....	14
Motivación para abordarlo resaltando la importancia de resolverlo	16
Que pasos se realizarán.....	17
Criterios de éxito	20
Salida del producto al mercado	20
Desarrollo de la solución	20
Desafíos técnicos	21
2 – ESTADO DE LA CUESTIÓN.....	22
Datos y cifras sobre el cáncer (según OMS)	22
Fundamentos teóricos y prácticos que sustentan las soluciones escogidas	24
Hardware	24
Software.....	25
Que trabajos del tipo propuesto han sido o están siendo realizados (si existen)	26
Técnicas y herramientas que fueron requeridos en el tratamiento del problema	26
Temas pendientes de resolución.	31
3 - DEFINICIÓN DEL PROBLEMA	33
El cáncer en la sociedad.....	33
Usuarios involucrados	34
Objetivos.....	34
Requisitos	36
Suposiciones	36
Restricciones.....	36
Alcance	36
Análisis FODA	37
4 - SOLUCIÓN PROPUESTA	41
Análisis de Porter	42
Amenaza de entrada de nuevos competidores.....	42
Rivalidad de los Competidores.....	42
Poder de negociación de los proveedores.....	42
Poder de negociación de los compradores.....	42
Amenaza de ingreso de productos sustitutos	43
Procedimiento lógico del sistema	43



Hardware	43
Conexiones físicas realizadas	48
Software.....	55
Documentación Técnica	59
Product backlog	61
Historias de Usuario	63
Sprint backlog.....	70
Diagrama Entidad – Relación.....	73
Formulación del Proyecto.....	75
Alcance del Proyecto	75
Estructura de Desglose del Trabajo (EDT)	75
Diccionario de la EDT.....	76
Recursos Necesarios.....	81
Recursos Humanos del Proyecto	82
Tiempo del Proyecto.....	84
Tareas	84
Diagrama de Gantt	90
Adquisiciones del Proyecto	97
Riesgos del Proyecto	100
Matriz de Probabilidad e Impacto	102
Comunicaciones del Proyecto	103
Calidad del Proyecto.....	103
Integración del Proyecto	104
Evaluación Económica - Financiera del Proyecto	104
Costos del Proyecto	104
Costos de Recursos Humanos.....	104
Costos de Adquisiciones.....	105
Costos de Mantenimiento	106
Costos Totales	106
Flujo de Fondos	107
Inversión Inicial.....	109
Plan de Pago	109
Beneficios del Proyecto	109
Beneficios cualitativos.....	109
Beneficios cuantitativos.....	110
Alternativas Tecnológicas	110



Alternativas de Hardware	110
Alternativas de Software	111
Elección de Alternativas.....	112
Arquitectura de procesamiento y comunicación de la propuesta tecnológica.....	114
5 - RESULTADOS O VERIFICACIÓN EXPERIMENTAL.....	115
Experimentos y pruebas realizadas.....	115
Software.....	115
Seguridad.....	115
Funcionalidad	117
Eficiencia	118
Portabilidad	119
Conectividad.....	119
Hardware	119
Montaje:.....	119
Tiras reactivas:.....	120
Conectividad:.....	121
Codificación Arduino:	122
Resultados obtenidos	123
Software.....	123
Funcionalidad	123
Eficiencia	124
Portabilidad	125
Conectividad.....	125
Montaje:.....	125
Tiras Reactivas:.....	126
Conectividad:.....	126
Codificación Arduino:	126
Calidad de los resultados en comparación con otras soluciones similares.....	126
Evaluación general de procedimientos, recursos y facilidades	126
Proyecto	126
Producto.....	127
Dificultades no previstas encontradas y sus soluciones	128
Software.....	128
Hardware	129
6 - CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN.....	131
7 - BIBLIOGRAFÍA.....	132



8 - ANEXOS	134
Domótica	134
Beneficios de la domótica.....	134
Características del sector	136
Enfermedad.....	136
Conceptos	136
Enfermedad.....	136
Salud	137
Cáncer.....	137
Datos y cifras	137
El problema	138
¿Cuál es la causa del cáncer?	138
Factores de riesgo del cáncer.....	139
Modificación y prevención de los riesgos	139
Estrategias de prevención	139
Detección temprana.....	140
Respuesta de la OMS.....	140
Dietas	141
Alimentos alcalinos y pH	141
Lista de alimentos alcalinos.....	142
Lista de alimentos neutros	143
Lista de alimentos ácidos	143
pH	144
La escala del pH	144
Homeostasis y regulación del nivel de pH en el cuerpo.....	145
¿Cómo se mide el pH?.....	145
Sangre.....	145
Orina.....	145
Saliva	145
¿Que origina un pH ácido?	146
Como afecta un pH ácido a nuestro organismo	147
Sintomatología de un pH ácido	148
Otto Heinrich Warburg.....	148
Lenguajes de programación.....	150
Ruby	150
Paradigma	151



La flexibilidad de Ruby.....	151
Los Bloques.....	151
Ruby y el Mixin	152
La apariencia visual de Ruby	152
Ruby on Rails	152
¿Qué es Rails?	153
¿Para qué se utiliza?.....	153
¿Cuál es su Origen?	153
Base de datos	153
PostgreSQL.....	153
Requerimientos de Instalación	153
Usando RVM.....	153
Instalando Postgres	154
Configurando Postgres	154
Creando la aplicación en Rails	154
Levantar la aplicación.....	155
Hardware	156
Arduino.....	156
Elementos de un sistema domótico	157
Elementos del prototipo del sistema domótico.....	158
ESP-8266.....	158
Conexionado	158
Características Técnicas	159
Usando el Firmware AT	159
Utilización de los comandos.....	159
Listado de Comandos	160
Fuente para Protoboard.....	162
Sensor de ultrasonidos – HC-SR04	163
Características	163
Pines de conexión.....	163
Sensor de Color - TCS3200	164
Características	164
Uso de RGB en HTML y en lenguajes de programación	165
Servo.....	165
Heroku	166
Dynos.....	167



Principales características.....	167
Seguridad (SSL)	168
Metodología para la gestión del proyecto.....	170
¿Por qué utilizar PMBOK?	170
Análisis de Porter	172
ANMAT	175



Índice de Imágenes

Imagen # 1: Técnicas y herramientas utilizadas.....	26
Imagen # 2: Sensor de Ultrasonidos HC-SR04.....	27
Imagen # 3: Sensor de Color TCS3200.....	27
Imagen # 4: Módulo Wifi ESP-8266.....	28
Imagen # 5: Fuente de Alimentación.....	29
Imagen # 6: Servo.....	29
Imagen # 7: Placa Arduino UNO.....	30
Imagen # 8: Factores que influyen en la calidad de vida de la persona.....	35
Imagen # 9: Análisis FODA.....	38
Imagen # 10: Sensor Distancia HC-SR04 (Conexión).....	44
Imagen # 11: Servo (Tomar tira reactiva).....	45
Imagen # 12: Servo (Analizar tira reactiva).....	45
Imagen # 13: Servo (Expulsar tira reactiva).....	46
Imagen # 14: Modos de Conexión ESP-8266.....	47
Imagen # 15: Módulo ESP-8266.....	48
Imagen # 16: Conexión Física del Sensor de Color.....	48
Imagen # 17: Conexión Física del Sensor de Color (Circuito).....	49
Imagen # 18: Conexión Física del Sensor de Ultrasonidos HC-SR04.....	49
Imagen # 19: Conexión Física del Sensor de Ultrasonidos HC-SR04 (Circuito).....	50
Imagen # 20: Conexión Física del Módulo Wifi ESP-8266.....	50
Imagen # 21: Conexión Física del Módulo Wifi ESP-8266 (Circuito).....	51
Imagen # 22: Conexión Física del Servo.....	51
Imagen # 23: Conexión Física del Servo (Circuito).....	52
Imagen # 24: Conexión Física del Sistema Completo.....	52
Imagen # 25: Conexión Física del Sistema Completo (Circuito).....	53
Imagen # 26: Sistema Completo.....	54
Imagen # 27: Vista “Home” del Sistema.....	56
Imagen # 28: Dietas específicas según pH.....	56
Imagen # 29: Detalle de una dieta seleccionada.....	57
Imagen # 30: Agregar producto al carrito de compras.....	57
Imagen # 31: Carrito de Compras.....	57
Imagen # 32: Pago de una Orden de Compra.....	59
Imagen # 33: Metodología SCRUM.....	60
Imagen # 34: Diagrama Entidad Relación (DER).....	74
Imagen # 35: Estructura de Desglose del Trabajo (EDT).....	76
Imagen # 36: Diagrama de GANTT.....	96
Imagen # 37: Arquitectura de procesamiento y comunicación.....	114
Imagen # 38: Codificación – Seguridad.....	115
Imagen # 39: Seguridad – Validación de Modelos.....	116
Imagen # 40: Seguridad – Encriptación.....	117
Imagen # 41: Tiras Reactivas.....	120
Imagen # 42: pH de componentes.....	121
Imagen # 43: Rendimiento en Conectividad.....	125
Imagen # 44: Beneficios de la domótica.....	135
Imagen # 45: Dietas por pH.....	144
Imagen # 46: Elementos de un Sistema Domótico.....	157
Imagen # 47: Conexión ESP-8266.....	159
Imagen # 48: Fuente de Alimentación.....	162
Imagen # 49: Sensor Ultrasónico HC-SR04.....	163



Imagen # 50: Sensor de Colores TCS3200.....	164
Imagen # 51: Servo.....	165
Imagen # 52: Alternativas de Dynos.....	168
Imagen # 53: Áreas de Conocimiento PMBOK.....	171
Imagen # 54: Esquema de las 5 fuerzas de Porter.....	173

Índice de Tablas

Tabla # 1: Product Backlog del Sistema.....	62
Tabla # 2: Resumen de Historias de Usuario.....	71
Tabla # 3: Resumen de Tareas por Historia de Usuario.....	73
Tabla # 4: Cantidades de RRHH.....	83
Tabla # 5: Matriz de Responsabilidades.....	83
Tabla # 6: Duración Total del Proyecto.....	84
Tabla # 7: Desagregación de Tareas.....	89
Tabla # 8: Matriz de Probabilidad e Impacto.....	102
Tabla # 9: Costos de Recursos Humanos.....	104
Tabla # 10: Costos de Adquisiciones.....	105
Tabla # 11: Costos de Adquisiciones (Arduino).....	105
Tabla # 12: Costos de Adquisiciones.....	106
Tabla # 13: Costos Totales.....	106
Tabla # 14: Flujo de Fondos.....	107
Tabla # 15: Verduras Alcalinas.....	142
Tabla # 16: Frutas Alcalinas.....	142
Tabla # 17: Otros Alimentos Alcalinos.....	142
Tabla # 18: Alimentos Neutros.....	143
Tabla # 19: Alimentos Ácidos.....	143
Tabla # 20: Comandos ESP-8266.....	162



Abstract

El siguiente proyecto de grado, se realizó aplicando diferentes conocimientos y aprendizajes adquiridos a lo largo de toda la carrera de Ingeniería en Informática, en donde el enfoque principal fue la aplicación de conceptos de domótica, desarrollo de sistemas y gestión de proyectos.

En este documento se trata la problemática del creciente número de enfermedades cancerígenas a nivel mundial. La importancia de este problema es tratar de ayudar a las personas, y concientizar a las mismas en su forma de vida diaria, para prevenir enfermedades a corto/mediano o largo plazo.

El objetivo principal fue realizar un sistema informático basado en la domótica, que abarque tanto software como hardware, capaz de mejorar la calidad de vida.

Uno de los grandes desafíos que tuvimos, fue el tratar de realizar un sistema domótico de bajo costo, en donde no se conocen en el mercado equipos que realicen la misma funcionalidad de la que se propone. Esto a su vez, requiere de una mayor investigación tanto en aspectos legales, técnicos y operativos para que el producto sea aceptado y de buena calidad.

La metodología que se implemento en este proyecto fue SCRUM, debido a su agilidad y adaptación a la realización de proyectos domóticos, y a su vez, se gestiono el trabajo en base a la guía de las buenas prácticas del PMBOK.

La implementación de este proyecto de grado, permitió que el usuario final adquiriera un producto de fácil uso, económico y que sea útil en su rutina diaria de control de la salud. Los resultados obtenidos fueron acordes a lo planificado, por lo que podemos concluir que el proyecto fue exitoso en cuanto a las expectativas que se tenían del mismo.

Concluimos que el tema en cuestión necesita de mayor investigación e inversión para poder realizar un producto que pueda tener salida al mercado. A los fines de este proyecto, se realizo un prototipo que podrá ser usado de base para futuras investigaciones, cumpliendo así con el alcance de este trabajo.

Palabras claves:

- Domótica
- Prototipo
- Programación
- Servidor
- Ruby on Rails
- Arduino
- SCRUM
- PMBOK
- Enfermedad
- Salud



1 - INTRODUCCIÓN

En este primer capítulo se pondrá en conocimiento aquellos conceptos y aspectos básicos del proyecto, con el objetivo de ofrecer un panorama general de los propósitos del mismo. Se incluye: una introducción del problema señalando sus particularidades, y también se definirán criterios de éxito y el contenido del informe.

El objetivo de este capítulo es establecer conocimientos básicos al lector, para la correcta comprensión de lo detallado en el presente documento.

El proyecto está motivado por la idea de crear un sistema informático, abarcando tanto software como hardware, capaz de mejorar la calidad de vida de las personas. Cabe aclarar que en todo el documento se habla de sistema domótico siendo el mismo un prototipo funcional, el cual debería estar suficientemente perfeccionado en todos los aspectos requeridos, y alcanzar metas que se pensaron para poder salir al mercado como un producto final.

La asistencia a los usuarios para controlar su estado de salud en su día a día, el otorgamiento de recomendaciones y la facilitación de su cumplimiento, son la base de la cual parte este proyecto de grado.

Uno de los principales aspectos que preocupa en la actualidad, es el creciente déficit de salud que poseen las personas, causado por el crecimiento del estrés laboral, la malversación de alimentos, las dietas mal realizadas entre otros, por lo cual decidimos integrar avances tecnológicos con la rama de la medicina, para así concientizar a las personas sobre lo perjudicial que pueden ser estos hábitos, y también ayudar de una manera práctica a incluir nuevas costumbres en la vida diaria para prevenir enfermedades.

Partiendo de los problemas previamente mencionados, decidimos indagar en cuanto a las consecuencias de los mismos: el nombre que más se hacía presente es el de la enfermedad del cáncer. Recopilando información de fuentes médicas por distintos medios, como internet, consultas personales, bibliografía y estudios médicos, llegamos a la conclusión que estos problemas desembocan, en gran parte, en esta enfermedad. Este descubrimiento fue el que permitió continuar con el segundo paso: indagar sobre las causas del cáncer.

Para poder cumplir con lo antes especificado basamos el proyecto en los descubrimientos del Dr. Otto Heinrich Warburg (Friburgo de Brisgovia, 8 de octubre de 1883 - Berlín, 1 de agosto de 1970), el cual expuso su tesis llamada “La causa primaria y la prevención del cáncer” que fue premiada en 1931 con un premio Nobel.

Warburg desarrolló grandes avances en el campo de la medicina, en especial en lo que respecta a la mutación celular. Asegura que el principal motivo por el cual una célula pasa de un estado normal a mutar y convertirse en una célula cancerígena, es por la falta de oxígeno. En su tesis expone que el principal motivo de que un cuerpo se encuentre en un estado no saludable, es porque se encuentra acidificado. Un cuerpo en estas condiciones provee a las células de un ambiente carente de oxígeno, por lo cual es necesario que muten para sobrevivir.

Resumiendo, Otto Heinrich Warburg expuso en su tesis que un ambiente acidificado es sinónimo de un ambiente carente de oxígeno en nuestro cuerpo. Como comentamos



anteriormente, un ambiente ácido no permite la presencia del oxígeno, elemento esencial para la supervivencia de las células. En cambio, un ambiente alcalino realiza lo contrario, permite el oxígeno. Warburg asegura que privar a una célula de 35% de su oxígeno durante 48 horas puede convertirlas en cancerígenas.

Como existe un ambiente alcalino y un ambiente acidificado, también podemos hablar de un ambiente neutro, el cual tiene un pH de 7. El pH es un coeficiente que indica el grado de acidez o de alcalinidad de una sustancia. Un pH de 5.5 indica acidificación, lo cual permite que las células del cáncer se propaguen, ya que dichas células pueden sobrevivir sin oxígeno. Un tejido cancerígeno es un tejido ácido, en cambio, un tejido sano es un tejido alcalino.

En su obra "El metabolismo de los tumores" (Warburg, 1931), Warburg demostró que todas las formas de cáncer se caracterizan por dos condiciones básicas:

- Ambiente ácido.
- La falta de oxígeno.

También descubrió que las células cancerígenas son anaerobias (no respiran oxígeno) y no pueden sobrevivir en presencia de altos niveles del mismo. En cambio, sobreviven gracias a la glucosa, siempre y cuando el entorno esté libre de oxígeno. Por lo tanto, el cáncer no sería nada más que un mecanismo de defensa que tienen ciertas células del organismo, para continuar con vida en un entorno ácido.

Entendiendo la teoría de Warburg, podemos aplicar conocimientos informáticos para ayudar a las personas a regular su pH.

La idea principal del trabajo a realizar, es proveer a las personas un sistema informático automatizado que sea capaz de indicar a las mismas de una manera práctica y sencilla cómo regular estos niveles de acidez y alcalinidad de nuestro organismo, lo cual tendría como consecuencia inmediata la prevención del cáncer, logrando así aumentar la calidad de vida.

El sistema contará tanto de hardware como de software. El hardware será el encargado de realizar las mediciones del pH a través de la orina de las personas. Una vez obtenida la medición y enviada a un servidor web. El software será el encargado de transformar este dato en información para el usuario, y así otorgarle una serie de pautas y conductas a seguir para poder equilibrar su organismo.

Para mejorar aún más la experiencia del usuario, se facilitará la obtención de los materiales para poder seguir los pasos indicados por el sistema. Esto será posible gracias a que los materiales faltantes podrán ser seleccionados, lo cual generará una orden de compras en un supermercado determinado, y los mismos serán entregados en el hogar del usuario mediante servicios prestados por el centro de abastecimiento.



Breve descripción del problema y su importancia

El pH es una medida de acidez o alcalinidad en una disolución como expusimos anteriormente. La escala del pH varía del 0 al 14, considerándose un pH neutro el correspondiente al valor 7.

Cuando el pH baja del 7, se considera un pH ácido (con exceso de iones de hidrógeno, H⁺), y cuando aumenta de 7 se considera un pH alcalino.

Un cuerpo saludable es un cuerpo con índices de alcalinidad, por esto es necesario y de suma importancia verificar a menudo nuestro nivel de pH en el cuerpo. Por lo general, las personas no se realizan controles médicos de forma seguida, y esto puede ser un inconveniente a futuro, ya que si lo harían se podrían detectar déficit en algunos aspectos de nuestra vida diaria que perjudiquen la salud, y por lo tanto la calidad de vida.

La importancia de este problema es tratar de ayudar a las personas, y concientizar a las mismas en su forma de vida diaria, para prevenir enfermedades a corto/mediano o largo plazo.

Carácter del problema

Para llevar una vida saludable, el equilibrio que existe en el pH es muy importante, y esto es uno de los problemas más graves que se tiene en la actualidad. Existen varias causas por las cuales este nivel de pH no se encuentra balanceado:

Una de las tareas principales del cuerpo humano es procesar los distintos materiales que ingresan a nuestro organismo con fines distintos, dependiendo tanto de la sustancia, como del órgano encargado de procesarla. Este trabajo continuo genera desechos que deben ser eliminados. El organismo no puede eliminar el 100% de estos residuos por el estilo de vida moderno que llevamos. Existen irregularidades presentes a la hora de establecer un horario adecuado de descanso, lo que produce alteraciones al correcto metabolismo del hombre.

No descansar lo suficiente, sumado al estrés diario que tiene una tendencia creciente conforme pasa el tiempo, produce que nuestro cuerpo cree más desechos de los que puede eliminar.

- *“Estado de cansancio mental provocado por la exigencia de un rendimiento muy superior al normal; suele provocar diversos trastornos físicos y mentales.”* (Oxford, 2016)
- *“El estrés es un estado de sobre esfuerzo, generalmente psíquico, al que está sometido el organismo en todos sus planos y que puede llevarnos a la pérdida, parcial o total, de la salud. Asimismo, entendemos que la salud es un estado de bienestar físico, psíquico y social; lo que llamamos biopsicosocial. Por lo tanto, diremos que la salud no es sólo la ausencia de enfermedad, sino el equilibrio de bienestar antes descrito.”* (Dory Martínez, 2016)



- *“El estrés es la tensión provocada por situaciones agobiantes, que originan reacciones psicósomáticas o trastornos psicológicos a veces graves.”* (Real Academia Española, 2016)

El espacio laboral y cultural en el que se han desarrollado enfermedades crónicas, influye de manera decisiva al encontrarse rodeado por el estrés. Esta situación puede ser temporal - la cual llamaríamos aguda y duraría menos de tres meses, o crónica, que duraría más de tres meses y es más difícil de eliminar.

La segunda razón es el déficit nutricional. La mayor parte de los alimentos que consumimos son de naturaleza ácida, produciendo una sobre acidificación de las células, tejidos, órganos y finalmente la sangre.

Al decir alimento ácido no se refiere a su sabor, sino a que los desechos que producen son ácidos. Los cereales y las carnes son (en su gran mayoría) ácidos; las frutas y vegetales son alcalinos. Las frutas cítricas pueden tener un sabor ácido, pero contienen minerales alcalinos, por lo cual se les considera alcalinos.

Con el paso del tiempo se está informando cada día más, sobre la alimentación alcalina y como la acidez de los alimentos puede afectar nuestra salud. La alimentación propuesta por los naturistas cada vez da más importancia a los alimentos alcalinos, ya que la evolución que sufren los alimentos después de la digestión, influyen de manera importante en la correcta actividad del organismo.

Después de haber terminado el proceso de digestión, todo aquello que hubiéramos ingerido se descompone en los nutrientes básicos de los cuales estaba formado; es decir que pasa a ser materia asimilable por nuestro organismo, que dependiendo de la cantidad de proteínas, hidratos de carbono, grasas, minerales y vitaminas, etc. que posean determinarán la acidez o la alcalinidad residual. Es decir, dejan unas “cenizas” que pueden ser de naturaleza alcalina o ácida dependiendo de que si los alimentos ingeridos fueron alimentos alcalinos o alimentos ácidos.

“Los alimentos que contribuyen con los macro y los micronutrientes necesarios para nuestro sustento y mantenimiento diario, también son los responsables de las aportaciones de acidez y alcalinidad a nuestro organismo, las cuales son imprescindibles para tener un organismo sano.” (Eco Vida, 2016)

La tercera razón es el medio ambiente. Mueren muchas más células saludables por la acción de la polución del aire, el agua y la tierra que por la muerte natural producida por el proceso metabólico. Esta situación genera más desechos tóxicos en el cuerpo.

El aire que respiramos está compuesto por 78% de nitrógeno, 21% de oxígeno, 0.093% de argón y una porción de vapor de agua. Cuando hablamos de contaminación, nos referimos a la alteración de esta composición, producida por causas naturales o provocadas por el hombre. Las primeras no se pueden evitar, pero las segundas si.

“Las fuentes que provocan la contaminación se clasifican en fijas: son todas las instalaciones establecidas en un sólo lugar, que tengan como finalidad desarrollar operaciones y procesos industriales, comerciales; y fuentes móviles, que son todo equipo o



maquinaria no fijos, con motores de combustión y similares, que con motivo de su operación generan emisiones contaminantes a la atmósfera.” (UNAD, 2016)

“La industria y el transporte son las dos principales fuentes de contaminación atmosférica. Datos oficiales revelan que el transporte público de pasajeros, de carga y particulares, generan el 80 % del total de los contaminantes a la atmósfera, el 3% lo representa la industria y el 10% restante el comercio y los servicios.

Se consumen 43 millones de litros de combustible al día. El 10% del presupuesto oficial, se destina al sector salud (enfermedades cardiovasculares y respiratorias).

Esta contaminación produce en los seres humanos como en los animales, irritación de las mucosas que puede derivar en graves enfermedades respiratorias y hasta llevar a la muerte. Como también producir diversos cánceres ya sea a los pulmones como al corazón.” (Luis Fernando Atilano Martínez, 2016)

Los desechos que no pueden ser eliminados por el cuerpo, se solidifican convirtiéndose en colesterol, ácido graso, ácido úrico, piedras en los riñones y vejiga, uratos, fosfatos, sulfatos, etc. y, de forma aún desconocida, se acumulan en el organismo. Esta acumulación de desechos en el organismo es lo que se conoce como el envejecimiento.

Motivación para abordarlo resaltando la importancia de resolverlo

Como se mencionó anteriormente, el objetivo principal de este proyecto es tratar de ayudar a las personas y concientizar a las mismas en su forma de vida diaria para la prevención de enfermedades.

Uno de los principales objetivos de la ingeniería es aplicar conocimientos, técnicas y herramientas para simplificar el accionar de las personas, por lo que abordamos este proyecto de manera tal que el proceso de control de pH esté automatizado, reduciendo al mínimo la interacción del usuario.

Actualmente no existe ninguna aplicación informática abocada a este problema, por lo cual se realizó este proyecto. Teniendo en cuenta la importancia de este problema, a continuación se detallan algunos aspectos que motivaron a llevarlo a cabo:

1. La salud de las personas es uno de los puntos más importantes por lo cual se realiza este proyecto y nos resulta de gran interés trabajar sobre esta necesidad. Como todos sabemos, las enfermedades son cada vez más frecuentes y a menor edad por los motivos que se detallaron anteriormente. Nuestro objetivo es aplicar la informática para reducir este problema, y lograr mejor calidad de vida en las personas.
2. En el proyecto no se necesitan muchos recursos monetarios para poder ejecutar de una forma correcta el sistema domótico, y esto motiva a la realización del mismo. Por esto, si realizamos un análisis de costo/beneficio para la adquisición del sistema domótico, el usuario va a resultar favorecido ya que obtendrá los mejores y mayores resultados de su estado de salud con una tasa de inversión (dinero) menor.



3. Permitir a cualquier persona, con un nivel de conocimiento básico de manejo de aplicaciones informáticas, acceder al sistema en cuestión. Esto hace referencia al hardware o sistema domótico como así también al sistema web para una posterior consulta sobre sus mediciones.
4. El proyecto es muy complejo, y se basa en teorías que tienen una relevancia innegable en el ámbito de la medicina. En la actualidad no existe una aplicación informática para poder controlar el pH de las personas. La única forma de realizar este control sin acudir a un médico es mediante unas tiras reactivas de pH, un proceso totalmente manual, motivo por el cual a las personas no les resulta práctico. Poder realizar un sistema que permita automatizar el procedimiento, es uno de los principales fines del proyecto, ya que el mismo permitirá acceder a más hogares y por lo tanto a más usuarios.
5. La domótica es un campo que todavía no está muy investigado en la provincia y no se le da el reconocimiento que posee. Actualmente existen placas (Arduino) que permiten realizar domótica sin tener extenso conocimiento sobre temas de electrónica, y se pueden conseguir manuales de ayuda para que este trabajo sea más ameno.
6. Realizar un proyecto informático funcional que abarque desde la creación del hardware hasta el desarrollo de software.
7. Concretar un sistema que tenga una aplicación práctica y teórica al finalizar el proyecto. Con leves mejoras el sistema creado podrá tener una salida al mercado instantánea. El mismo permitirá una rápida adaptación y evolución al ser totalmente modular.

Que pasos se realizarán

Para el desarrollo del sistema web se utilizará la metodología SCRUM, la cual permite adoptar una estrategia de desarrollo incremental, en lugar de planificación y ejecución completa del producto y a su vez, permite hacer un solapamiento de las diferentes fases del desarrollo.

Los pasos que se realizarán para ejecutar el proyecto son los siguientes:

1. Análisis del requerimiento:
 - Descripción del problema
 - Especificación del caso a analizar
 - Descripción de los principales inconvenientes del paciente según el pH obtenido, junto con su grado de importancia e impacto en la vida diaria del individuo.
 - Modelo de negocio
 - Entrevistas y encuestas
 - Investigación sobre tecnologías existentes
 - Software
 - Hardware
 - Software



- Hardware
 - Conectividad
 - Evaluación:
 - Evaluación económica
 - Evaluación técnica
 - Evaluación operativa
 - Evaluación legal
 - Validación de requerimientos
 - Documentación de requerimientos
2. Diseño del Sistema:
- Interfaz
 - Master page
 - Administración de usuarios
 - Administración de dietas
 - Administración de órdenes de compra
 - Logueo al sistema
 - Administración de mediciones
 - Datos
 - Funcional
 - Módulos de Healty
 - Administración de usuarios
 - Administración de dietas
 - Administración de órdenes de compra
 - Logueo al sistema
 - Administración de mediciones
 - Arduino
 - Obtención de medición
 - Obtención del pH.
 - Procesamiento de datos.
 - Envío al servidor.
 - Interacción con otros sistemas externos
 - Envío de orden de compra a un supermercado
 - Obtención de respuesta de un supermercado
 - Sistema domótico
 - Soporte
 - Posicionamiento de componentes
 - Plan de contingencias
 - Adquisición de componentes
 - Entregable de diseño
 - Validación del diseño
 - Documentación del diseño
3. Desarrollo del Sistema:
- Interfaz
 - Master page
 - Administración de usuarios
 - Administración de dietas
 - Administración de órdenes de compra
 - Logueo al sistema
 - Administración de mediciones
 - Base de datos



- Sistema domótico
 - Soporte
 - Posicionamiento de componentes
 - Codificación
- Funcional
 - Módulos de Healty
 - Administración de usuarios
 - Administración de dietas
 - Administración de órdenes de compra
 - Logueo al sistema
 - Administración de mediciones
 - Arduino
 - Obtención de medición
 - Obtención del pH.
 - Procesamiento de datos.
 - Envío al servidor.
 - Interacción con otros sistemas externos
 - Envío de orden de compra a un supermercado
 - Obtención de respuesta de un supermercado
- Entregable de desarrollo
- Validación de desarrollo
- Validación del sistema domótico
- Documentación de desarrollo

4. Pruebas del Sistema:

- Componentes domóticos
 - Unitarias
- Funcionalidad del sistema domótico
 - Integración
 - Validación
 - Sistema
- Funcionalidad del sistema web (Healty)
 - Administración de usuarios
 - Administración de dietas
 - Administración de órdenes de compra
 - Logueo al sistema
 - Administración de mediciones
- Datos
- Interacción entre sistemas
 - Envío de orden de compra a un supermercado
 - Obtención de respuesta de un supermercado
 - Comunicación entre Arduino y Healty
- Validación de pruebas
- Documentación de pruebas

5. Despliegue:

- Monitorización y mantenimiento del sistema domótico y web
- Generación del informe final
- Revisión del proyecto
- Validación del despliegue
- Documentación de despliegue



Los pasos anteriormente mencionados, serán explicados con mayor detalle a la hora de desarrollar los capítulos posteriores.

A su vez, se hará uso de la guía de buenas prácticas del PMI (Project Management Institute) conocida como PMBOK, para gestionar diferentes áreas de conocimiento dentro del Proyecto. Dicha guía será explicada con más detalle en el capítulo 8 - ANEXO – Metodología para la Gestión del Proyecto.

Criterios de éxito

Además del objetivo académico por el cual se lleva a cabo este proyecto, se deberán tener en cuenta algunos criterios o pautas para lograr gestionar exitosamente el trabajo.

Salida del producto al mercado

1. Es fundamental conseguir una comprensión clara del problema que se tiene, que es el estado de salud de las personas. Una vez logrando esto, se puede establecer efectivamente la solución que se propone en este trabajo final, que es llegar a concientizar a las mismas en su forma de vida diaria para prevenir enfermedades como el cáncer.
2. Incorporar a una rutina diaria el control de la salud por medio de este sistema domótico, que es diferente a los sistemas manuales que se encuentran hoy en día en el mercado para la medición de pH.
3. Realizar un sistema domótico de bajo costo que permita llegar a la mayor parte de la población. Actualmente la domótica, en su gran mayoría, no posee esta característica ya que los costos de los productos del mercado son muy elevados.

Desarrollo de la solución

1. Poder realizar un sistema domótico funcional, en donde la tecnología, metodologías empleadas y recursos, sean de fácil alcance para ejecutar el mismo de la manera más eficaz y eficiente, y así obtener un producto final de alta calidad.
2. Obtener un resultado desde el sistema domótico, y que el mismo sea enviado al servidor web para una posterior consulta de esta información por parte del usuario final. Estos datos serán de vital importancia, ya que esto determinará el estado de salud que posee la persona. Los mismos deberán ser obtenidos de forma eficaz y comprensible para el usuario.
3. Por último, el usuario final, según los resultados obtenidos, podrá seleccionar desde una lista de ingredientes aquellos que necesita para realizar o preparar su dieta asociada al pH. Esto se transformará en una petición a un supermercado, que posea un sistema de ventas por internet.



Desafíos técnicos

1. Desarrollar un prototipo, el cual permita homologar mediante una serie de pruebas, un sistema domótico, el cual cumpla con las necesidades presentadas en el corriente proyecto. Estas pruebas deberán ser superadas antes de poder entregar una versión final del sistema. La homologación abarca tanto software como hardware.
2. Contar con los conocimientos necesarios en domótica, SCRUM, Arduino, Ruby on Rails y en los distintos componentes que se involucran en este proyecto para poder cumplimentar con un resultado final de calidad esperada.



2 – ESTADO DE LA CUESTIÓN

Antes de desarrollar los antecedentes relacionados con el problema en cuestión, resulta necesario definir determinados aspectos.

“El cáncer es un proceso de crecimiento y diseminación incontrolados de células. Puede aparecer prácticamente en cualquier lugar del cuerpo”. (OMS, 2016)

El tumor suele invadir el tejido circundante y puede provocar metástasis en puntos distantes del organismo. Muchos tipos de cáncer se podrían prevenir, evitando la exposición a factores de riesgo comunes como el humo de tabaco. Además, un porcentaje importante de cánceres pueden curarse mediante cirugía, radioterapia o quimioterapia, especialmente si se detectan en una fase temprana.

Datos y cifras sobre el cáncer (según OMS)

- *El cáncer es una de las primeras causas de muerte a nivel mundial; en 2012 se le atribuyeron 8,2 millones de muertes.*
- *Los cánceres que causan un mayor número anual de muertes son los de pulmón, hígado, estómago, colon y mama.*
- *Los tipos más frecuentes de cáncer son diferentes en el hombre y en la mujer.*
- *Aproximadamente un 30% de las muertes por cáncer, se deben a cinco factores de riesgo comportamentales y alimentarios (índice de masa corporal elevado, consumo insuficiente de frutas y verduras, falta de actividad física y consumo de tabaco y alcohol) y, por lo tanto, pueden prevenirse.*
- *Las infecciones que pueden provocar cáncer, como las causadas por los virus de las hepatitis B y C y el del papiloma humano, son responsables del 20% de las muertes por cáncer en los países de ingresos bajos y medianos, y del 7% en los países de ingresos altos.*
- *El tabaquismo es el factor de riesgo que por sí solo provoca un mayor número de casos y a nivel mundial, causa aproximadamente un 22% de las muertes por cáncer y un 71% de las muertes por cáncer de pulmón.*
- *El cáncer comienza con la transformación de una sola célula, que puede tener su origen en agentes externos y en factores genéticos heredados.*
- *El 70% de todas las muertes por cáncer registradas en 2012 se produjeron en África, Asia, América Central y Sudamérica. Se prevé que los casos anuales de cáncer aumentarán de 14 millones en 2012 a 22 millones en las próximas dos décadas.*

(OMS, 2016)



Analizando la información anterior, provista por la OMS, podemos entender que el cáncer es una de las principales causas de muerte en el mundo. Actualmente, los avances médicos sobre el tema no han podido encontrar una solución al problema, no existe una cura. Lo que sí existe, es una serie de pautas que permiten acomodarse a un estilo de vida saludable, el cual reduce las posibilidades de contraer esta enfermedad drásticamente.

Las pautas mencionadas hacen hincapié en dos grandes grupos: un estilo de vida con un estrés equilibrado, y una buena alimentación. Si bien estas recomendaciones son de público conocimiento, existe un gran grado de desinformación sobre el problema, lo que lleva a un constante crecimiento de casos de cáncer alrededor del mundo.

“Gracias a la revolución industrial, hemos degradado nuestra alimentación al punto de que lo bueno, natural y nutritivo de los alimentos se ha quitado en gran parte, dejando una cantidad considerable de residuos químicos utilizados para la preservación, crecimiento y estética de los alimentos

Estamos consumiendo, la mayoría de alimentos procesados, con efectos de acidificación corporal e intoxicación; esto nos está enfermando física y mentalmente, hasta el punto de tener una sociedad controlada por la depresión, enfermedades degenerativas, debilidad crónica, falta de memoria y concentración.

Se prevé que a nivel mundial, la mortalidad por cáncer aumentará un 45% entre 2007 y 2030 (pasará de 7,9 millones a 11,5 millones de defunciones), debido en parte al crecimiento demográfico y al envejecimiento de la población. En las estimaciones se han tenido en cuenta, las ligeras reducciones previstas de la mortalidad por algunos tipos de cáncer en países con grandes recursos. Se estima que durante el mismo periodo el número de casos nuevos de cáncer, aumentará de 11,3 millones en 2007 a 15,5 millones en el año 2030.

En la mayor parte de los países desarrollados, el cáncer es la segunda causa principal de mortalidad después de las enfermedades cardiovasculares, y los datos epidemiológicos muestran el comienzo de esta tendencia en el mundo menos desarrollado, en particular en los países en transición y países de ingresos medianos, por ejemplo en América del Sur y Asia. Más de la mitad de los casos de cáncer se registran ya en países en desarrollo.

La aparición de cáncer se ha asociado a varios factores de riesgo comunes. A saber: un modo de vida poco sano (consumo de tabaco y alcohol, dieta inadecuada, falta de actividad física) y exposición a carcinógenos (por ejemplo amianto) en el entorno laboral o en el medio ambiente (por ejemplo por contaminación del aire en locales cerrados), radiación (por ejemplo ultravioleta o ionizante) y algunas infecciones (por ejemplo hepatitis B o infección por virus del papiloma humano).” (OMS, 2016)



Fundamentos teóricos y prácticos que sustentan las soluciones escogidas

La solución propuesta se desarrolla mediante un sistema domótico, capaz de interactuar con el usuario y controlar su estado de salud.

La domótica en Argentina, todavía no tiene la relevancia que fue adquiriendo con el paso del tiempo en otros países desarrollados. Realizar un sistema innovador, en un área tan poco explorada a nivel país, fue uno de las motivaciones para iniciar el proyecto.

Hardware

Para la realización del hardware se utilizó Arduino. A continuación se describe una breve reseña del mismo:

Arduino es un prototipo de plataforma de código abierto basado en un sistema de fácil uso de hardware y software. Las placas Arduino son capaces de leer entradas de luz en un sensor, un botón presionado, o un mensaje en Twitter, transformarlo en una salida, activar un motor, encender un LED, y publicar algo online.

Uno es capaz de decirle a la placa Arduino que hacer, enviando un conjunto de instrucciones al microcontrolador situado en la misma. Para hacer esto, se usa el lenguaje de programación Arduino (basado en Wiring), y el software de Arduino (IDE), basado en Processing.

Todas las placas Arduino son completamente Open-Source, dando la capacidad a sus usuarios de construir proyectos independientes y adaptarlos a sus eventuales necesidades. El software es también de código abierto, y está creciendo conforme crecen las contribuciones de los usuarios a nivel mundial.

Se seleccionó Arduino por la simpleza a la hora de prototipar el hardware, ya que como se especifica anteriormente, no se necesita ser experto en electrónica para poder realizar un proyecto complejo.

De todas maneras, fue necesaria una breve capacitación en electrónica, ya que, por ejemplo, un error en el voltaje de entrada, en el amperaje de salida, o en alguna de las otras variables, podría provocar desde una falla en la salida/entrada de los pines de Arduino, hasta que la placa se averíe, ya que las mismas son muy sensibles.

Otro de los motivos para elegir Arduino, fue por la cantidad de información disponible en distintos foros, y por la disponibilidad de las placas en Argentina. Arduino se convirtió en el líder de los proyectos DIY (Do It Yourself). No resulta un problema encontrar información en la red sobre los distintos módulos disponibles para la interacción con Arduino; es más, el IDE de Arduino dispone de ejemplos con codificación funcional y explicada para los módulos más conocidos. Además de los foros, se dispone de las datasheets (hojas de datos) de los distintos microcomponentes que integran tanto Arduino como sus módulos.

La disponibilidad de las placas en Argentina, fue un tema de principal consideración para el proyecto, ya que se consideraron otras placas de microcontroladores, como Raspberry Pi, Open BCI, etc para el desarrollo, pero presentan inconvenientes debido al elevado precio,



o a que las leyes de importaciones de Argentina no permitían el ingreso de las mismas. Aun considerando esto, Arduino fue la opción que más se adaptó a nuestras necesidades.

Software

Otro aspecto a considerar en la realización del sistema domótico fue la aplicación web, donde el usuario podrá controlar su progreso, consultar dietas, realizar compras a supermercados, etc. Para desarrollar la página web, se tuvieron en cuenta una serie de lenguajes que se adecuaban a los requerimientos.

Luego de un análisis detallado se llegó a la conclusión, de que entre los lenguajes disponibles para realizar un entorno web, el que mejor se adapta a los requerimientos y funcionalidades que requiere el sistema, era Ruby on Rails.

Criterios de selección del lenguaje

- Debe ser un lenguaje que permita el desarrollo de aplicaciones web.
- Entre otros beneficios de utilizar MVC (Modelo - Vista - Controlador), esto nos facilitará la interacción con el hardware a través de las peticiones realizadas con Arduino.
- En cuanto a seguridad, deberá tener librerías incorporadas las cuales sean capaces de reducir las vulnerabilidades del sistema.
- Otro requisito fundamental es que permite la interacción con otros sistemas webs, ya sea por librerías, o que permita la creación de las mismas.
- El lenguaje debe tener guías disponibles para consultas.
- Preferiblemente debe tener una comunidad amplia, para consulta en caso de ser necesario.

Ruby es un lenguaje de programación dinámico y de código abierto, enfocado en la simplicidad y productividad.

Ruby es considerado un lenguaje flexible, ya que permite a sus usuarios alterarlo libremente. Las partes esenciales de Ruby pueden ser quitadas o redefinidas a gusto. Se puede agregar funcionalidad a partes ya existentes; Ruby intenta no restringir al desarrollador.

Rails es un framework de desarrollo de aplicaciones web, escrito en el lenguaje Ruby. Está diseñado para hacer la programación en Ruby más fácil, otorgando al programador todo lo necesario para comenzar un proyecto.

Permite escribir menos código para cumplir tus objetivos, comparados con otros frameworks. Desarrolladores experimentados, aseguran que también hace la programación más entretenida.

Que trabajos del tipo propuesto han sido o están siendo realizados (si existen)

Hoy en día existen herramientas o equipos tecnológicos que se utilizan para realizar mediciones de pH en orina, pero no se conocen trabajos o proyectos que están siendo realizados a los fines que se propone en este trabajo.

En la Facultad no se conoce algún trabajo final con la temática que se está planteando, el combinar la informática, domótica y la medicina con el objetivo de que las personas concienticen sobre su estado de salud.

Técnicas y herramientas que fueron requeridos en el tratamiento del problema

Entre las tecnologías que se utilizaran para el tratamiento del problema son las siguientes:

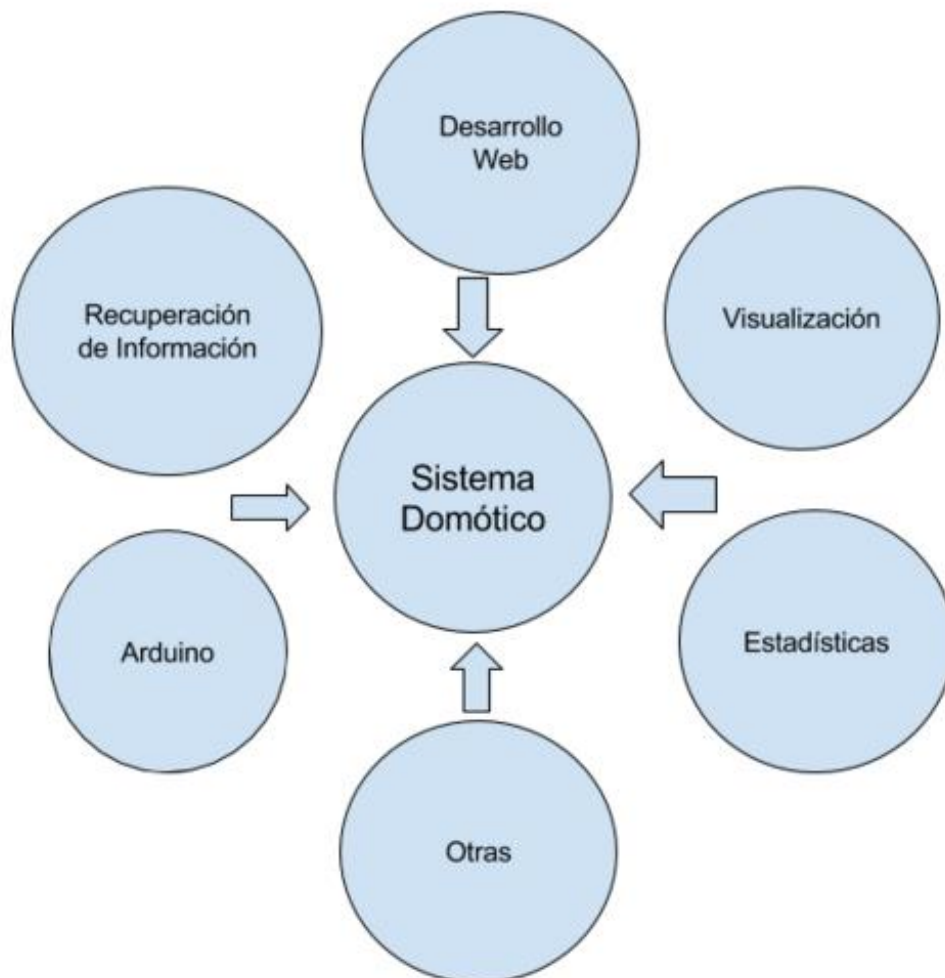


Imagen 1: Técnicas y Herramientas utilizadas

Arduino: se mencionó anteriormente lo que es la tecnología Arduino. Cabe aclarar que para la realización de este proyecto con Arduino, se utilizaron muchas partes o submódulos los cuales se detallan a continuación:

- Sensor de Ultrasonidos HC-SR04

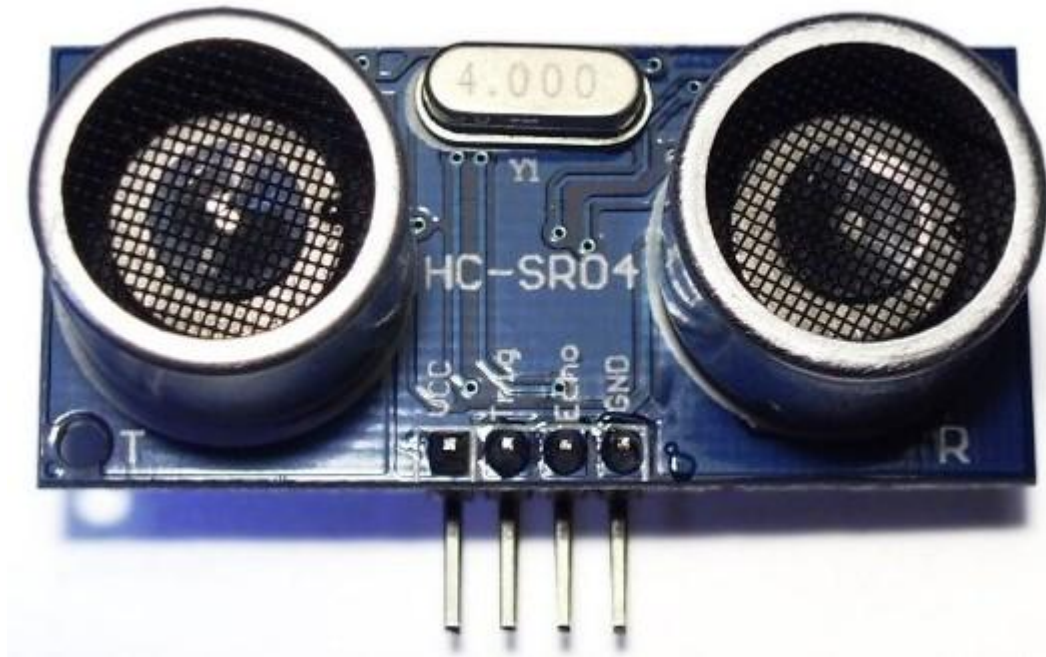


Imagen 2: Sensor de Ultrasonidos HC-SR04

El sensor de ultrasonidos se enmarca dentro de los sensores para medir distancias o superar obstáculos, entre otras posibles funciones. Es un módulo que incorpora un par de transductores de ultrasonido (inaudible para el oído humano por su alta frecuencia), que se utilizan de manera conjunta, para determinar la distancia del sensor con un objeto colocado enfrente de este. En nuestro caso se lo utilizará para la medición de distancias.

Este sensor tiene un rango de distancias sensible entre 3cm y 3m con una precisión de 3mm.

- Sensor de Color TCS3200



Imagen 3: Sensor de Color TCS3200

Es un completo sensor de color RGB con 4 diodos LED incorporados, y puede detectar casi un infinito rango de colores. Sus aplicaciones son varias como por ejemplo medir la luz ambiental, clasificación por color etc. En nuestro proyecto se lo utilizará para clasificar las tiras reactivas según el color.

Internamente el sensor tiene un array de pequeño fototransistores, cada uno de ellos con filtros rojo, verde y azul o sin filtro (clear), lo que le permite ser bastante preciso y sensible. Los filtros de cada color se distribuyen uniformemente a lo largo de la matriz, para eliminar el sesgo de ubicación entre los colores. El dispositivo tiene un oscilador, que produce una salida de onda cuadrada, cuya frecuencia es proporcional a la intensidad del color detectado.

- Módulo Wifi ESP-8266



Imagen 4: Módulo Wifi ESP-8266

El ESP-8266 es un microcontrolador del fabricante chino Espressif, que incluye la capacidad Wifi.

Es un chip altamente integrado, diseñado para las necesidades de un mundo cada vez más conectado. Ofrece una solución completa y autónoma de redes Wifi, lo que le permite acceder a cualquiera de las aplicaciones host o descargar todas las funciones de redes Wifi de otro procesador de aplicaciones.

Este módulo incluye todo lo necesario para conectarse a un punto de acceso Wifi mediante comandos de texto AT, vía puerto serie, que puede ser configurada a distintas velocidades. Una vez que lo instruimos para que se conecte a nuestra Wifi, el módulo es capaz de enviar información que le remitimos vía puerta serie a una dirección IP y puerto que deseamos.

El ESP-8266 tiene capacidades de gran alcance a bordo de procesamiento y almacenamiento, que permiten que sea utilizado con sensores y dispositivos específicos de la aplicación, a través de sus GPIOs (Entrada/Salida de Propósito General), con un desarrollo mínimo por adelantado y la carga mínima durante el tiempo de ejecución. Su alto grado de integración en el chip, permite la circuitería externa mínima.

- Fuente de Alimentación



Imagen 5: Fuente de Alimentación

Una fuente de alimentación, es un dispositivo que convierte la corriente alterna (CA) en una o varias corrientes continuas (CC), que alimentan los dispositivos de un aparato o componente electrónico al que se conectan.

Esta fuente de alimentación para protoboards es fácilmente de montar, y ofrece la posibilidad de alimentar un circuito electrónico con diferentes salidas.

Se puede conectar directamente sobre una protoboard y gracias a su selector, poder elegir entre una tensión de 3V o 5V para las distintas aplicaciones. Dispone de un conector DC "Jack" para conectar un transformador externo de corriente, o la posibilidad de conectarla a través de un conector USB.

- Servo



Imagen 6: Servo

Normalmente los motores habituales, lo que hacen es transformar la energía eléctrica en un giro continuo, que podemos usar para desarrollar trabajo mecánico.

“Los servos son también motores de corriente continua, pero en lugar de diseñarse para obtener un giro continuo que podamos aprovechar, se diseñan para que se muevan en un ángulo fijo en respuesta a una señal de control, y se mantengan fijos en esa posición.” (PROMETEC, 2016)

Estos servos o servomotores son muy frecuentes en robótica, por la capacidad que presentan para moverse a un ángulo concreto y mantenerse allí.

Habitualmente los servos tiene un margen de operación, es decir, pueden moverse entre 0° y ángulo dado, que suele ser de 180°, pero existen modelos comerciales de todas la características imaginables (incluyendo servos de 360°).

- Placa Arduino UNO



Imagen 7: Placa Arduino UNO

El UNO es una placa electrónica basada en el ATmega328P. Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 se pueden utilizar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reinicio.

Contiene todo lo necesario para apoyar el microcontrolador; simplemente conectarlo a un ordenador con un cable USB o el poder con un adaptador de CA o la batería a CC.

“Arduino UNO puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino, se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador.”
(Arduino, 2016)

Desarrollo WEB: se plantea el desarrollo de un sistema web, en donde el usuario final podrá interactuar y observar los resultados de sus mediciones para tener conocimiento de su estado de salud.



Recuperación de Información: se basa en la forma en que se van a tomar los datos desde el sistema domótico, para poder reflejarlos en una base de datos relacional. Estos datos deberán estar integrados, consistentes y no redundantes.

Visualización: estas herramientas son de gran utilidad porque a través de ellas los datos numéricos y textuales resultan mucho más amigables y comprensibles. Se integrarán librerías dentro del sistema web, para realizar esta visualización de algunos datos que se convertirán en información por el significado que llevan.

Estadísticas: es una ciencia formal y una herramienta que estudia el uso de los datos y el análisis proveniente de una muestra representativa de datos.

Otras: este sistema domótico precisa de la interacción con otras disciplinas y tecnologías: medicina, tecnologías de telecomunicación, tecnologías aplicadas a internet, etc.

Temas pendientes de resolución.

En lo que se refiere a una enfermedad como el Cáncer, que se puede presentar por muchos motivos, podemos encontrar muchos temas pendientes de resolución.

- Es una enfermedad que está muy analizada y avanzada en términos de investigación pero no se encuentran (o no se presentan) soluciones definitivas para la misma.

En este proyecto, nos enfocamos solamente en analizar una de las causas por las cuales se puede provocar un cáncer en la persona.

Cabe destacar que el cáncer no es una única enfermedad, existen distintos tipos, y para cada uno un tratamiento específico. Siempre se habla de cura para el cáncer de determinado órgano.

Esta enfermedad es un tema bastante extenso y está más relacionado con la medicina que con la informática. Este proyecto tiene como fin, aportar todo lo que creamos capaz de realizar por parte de la informática al estado de salud de las personas.

- *“La Quimioterapia es el uso de fármacos para destruir las células cancerosas. Actúa evitando que estas células crezcan y se dividan en más células. Debido a que las células cancerosas en general crecen y se dividen más rápido que las células sanas, la quimioterapia las destruye más rápido que a las células sanas.*

Como los fármacos de la quimioterapia son muy potentes, esto causa daño a muchas células en crecimiento, incluido las células sanas. Este daño causa un efecto secundario de la quimioterapia.

El objetivo depende del tipo de cáncer y de cuanto se haya diseminado. A veces el objetivo principal es destruir todo el cáncer y evitar que vuelva a aparecer. Si eso no es posible, la quimioterapia puede retrasar o ralentizar el crecimiento del cáncer. “(EFD, 2016)



Si bien la quimioterapia es recomendada por muchos médicos, es un tema muy discutido. Actualmente se presenta una tendencia medicinal a volcarse a otro tipo de soluciones menos invasivas, llamadas alternativas. No se puede resolver el grado de beneficios de la misma. Por este motivo, este proyecto escapa a los conceptos de esta práctica.

- El único ámbito en el cual hacemos énfasis es en el tema alimenticio, ya que lo consideramos uno de los más importantes y en el cual se puede aplicar a una mayor cantidad de usuarios. Es un hábito que incumbe a todas las personas en comparación de otros que son muy específicos de cada persona.



3 - DEFINICIÓN DEL PROBLEMA

En este capítulo se definirán claramente las necesidades y objetivos que se persiguen en el trabajo final. Además, para completar la exacta definición del problema, se detallarán los alcances de la solución, como así también las especificaciones de requisitos y restricciones que han de considerarse.

El cáncer en la sociedad

“El cáncer es una enfermedad que tiene su precedencia histórica, pero que fue avanzando y fue asentándose más en la sociedad conforme pasa el tiempo. Esto se debe a varios motivos que van creciendo conforme a la evolución de la sociedad por la mayor exigencia de las personas en su vida cotidiana, vida social, relaciones en el trabajo, el avance de la industrialización de los alimentos, el continuo estrés al que estamos expuestos y a la falta de una costumbre de cuidados de la salud.” (Juana Villalva, 2016)

Estos factores recién mencionados, producen cambios en el comportamiento fisiológico y psicológico de las personas, por lo cual el cuerpo (a nivel celular) debe adaptarse a un nuevo ambiente más hostil. Esto provoca que el organismo genere más “desechos”, los cuales siempre son eliminados (a veces es tanta la cantidad de toxinas que nuestro cuerpo no está listo para eliminarlas totalmente).

Esta acumulación de desechos, produce un ambiente carente de oxígeno para las células. Las células cancerígenas no son otra cosa que el resultado de la mutación por instinto de supervivencia en un ambiente hostil. La falta de oxígeno obliga a las células a mutar para poder mantenerse vivas.

Este tipo de células son las que provocan el cáncer. Cuando se comienzan a generar células cancerígenas, las mismas empiezan a comunicarse entre ellas e intercambian información por medio de esta comunicación intercelular, y esta forma de supervivencia se propaga. Esta es una de las teorías de las metástasis.

Si bien hoy en día no se consiguió una cura contra esta enfermedad o para revertir la mutación celular, se encuentran varios estudios disponibles sobre las causas del cáncer.

Con nuestros conocimientos no podemos encontrar una cura contra el cáncer, ya que esto sería más abocado a la parte medicinal, pero si podemos utilizar la informática para que, con base en estudios médicos previamente realizados, ayuden a la sociedad a adoptar estos hábitos que se mencionan en dichos estudios.

Estos hábitos que se pueden deducir de la Tesis del Dr. Otto Heinrich Warburg, si bien ayudan a la prevención del cáncer reduciendo al mínimo los ambientes carentes de oxígeno para evitar la mutación de las células, no aseguran que al seguir estos hábitos se evite la enfermedad.



Usuarios involucrados

Para ser más específico en la definición del usuario, detallaremos a continuación un usuario determinado, en el cual nos basaremos para la realización de la domótica de todo el proyecto. Los detalles se basarán solamente en características prácticas que puedan aportar información relevante al proyecto (se evitará información medicinal innecesaria para mantener el entendimiento del lector).

- Usuario Final / Paciente:
 - **Nombre:** Elio Adrián
 - **Apellido:** Díaz
 - **Edad:** 69 años
 - **Sexo:** Masculino
 - **Estado civil:** casado
 - **Enfermedades relevantes:** Diabetes, hipertensión.
 - **Nivel de pH actual:** 4.4
 - **Profesión:** Empresario.
 - **Horarios de trabajo:** 8:00 - 14:00, 17:00 -22:30. Lunes a Sábados
 - **Observaciones:**
 - El usuario presenta un estilo de vida sumamente perjudicial para la salud del mismo. Esta afirmación la basamos en que su lugar de trabajo presenta **altos grados de contaminación ambiental, sonora y visual**.
 - El paciente posee una **dieta no regulada**, lo cual presenta un gran perjuicio a su pH.
 - La cantidad de horas trabajadas superan las 8 hs recomendadas por la OMS y por las normas de higiene y seguridad implementadas en nuestra provincia, lo cual perjudica a su condición biopsicosocial, produciendo un alto grado de **estrés**.
- Usuario Final / Grupo familiar:
 - Compuesto por:
 - Esposa: Ana María Martínez.
 - Hijos: Claudio Federico Díaz Martínez, María Lorena Díaz Martínez, Maximiliano Díaz Martínez, Jimena Natalia Díaz Martínez, María Guadalupe Díaz Martínez y Facundo Adrián Díaz Martínez.
 - Edades: 23 a 67 años.
 - Nivel de conocimientos informáticos: Medio.

Objetivos

Como se planteó anteriormente a la hora de explicar la problemática que se tiene, el objetivo principal de este proyecto final es armar un sistema domótico que permita mejorar la calidad de vida del usuario, concientizando a las mismas en su forma de vida diaria para

prevenir enfermedades, dirigiendo principalmente nuestra atención en el cáncer. Para poder realizar esto, nos enfocaremos en objetivos secundarios que permitirán cumplir el objetivo primario.

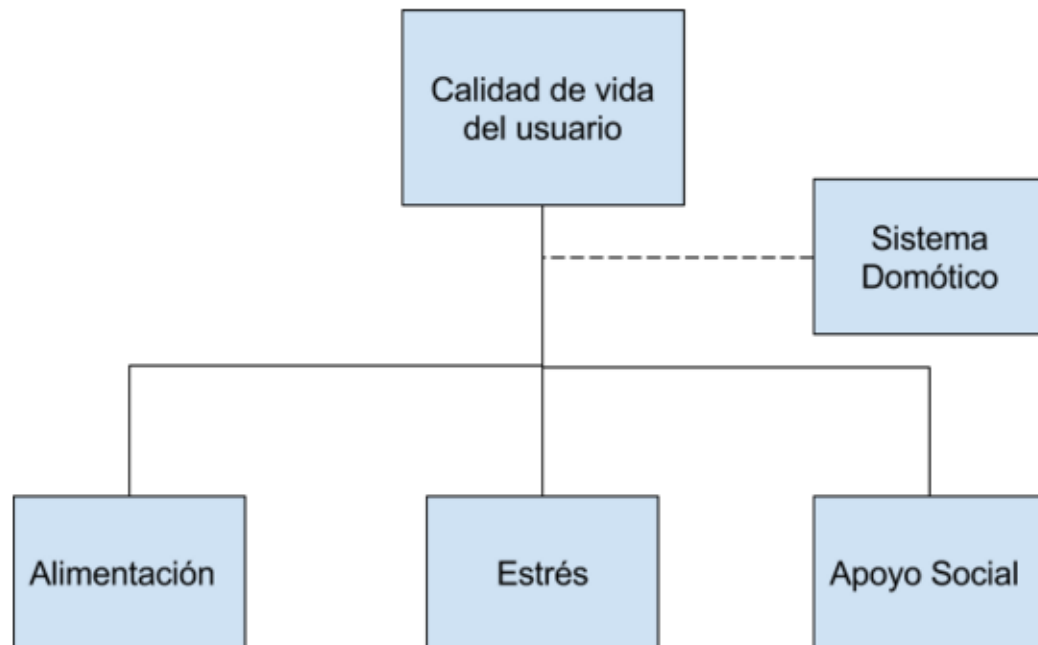


Imagen 8: Factores que influyen en la calidad de vida de la persona.

Teniendo en cuenta el alto grado de personas que no se realizan controles ni chequeos médicos para controlar su estado de salud, se propone la implementación de este sistema domótico a través del cual se podrá obtener información confiable.

Las metas que se consideran para cumplir con la implementación del objetivo planteado son las siguientes:

- Lograr un amplio entendimiento del problema que se tiene y de las partes que se ven involucradas en el sistema domótico. Esto permitirá obtener resultados precisos que son útiles para el usuario final.
- Lograr que los pacientes usen este sistema domótico en su rutina diaria para el control de la salud.
- Realizar un sistema domótico en el cual se obtengan valores confiables y precisos. Estos resultados obtenidos deberán tener un alto grado de confiabilidad para que nuestro usuario final pueda seleccionar su dieta.
- Interpretar y exponer estos resultados de una forma clara y amigable hacia el usuario final, es este caso el paciente.
- Aplicar la metodología seleccionada (SCRUM) y la guía de buenas prácticas para la gestión de proyectos (PMBOK), de una forma correcta para gestionar dicho proyecto de una forma óptima y aumentar las posibilidades de concluir el trabajo exitosamente.



Requisitos

- Se debe aplicar una correcta gestión para las áreas de: Alcance, Tiempo, Costos, Calidad, Recursos Humanos, Comunicaciones, Riesgos, Adquisiciones e Integración.
- Se deberá cuidar la confidencialidad de la información que se obtiene desde el sistema domótico, ya que estos resultados son datos sensibles del paciente.

Suposiciones

- Los datos utilizados para ejecutar este proyecto estarán relacionados con las mediciones de pH que se realicen los pacientes.

Restricciones

- El tiempo del proyecto tiene como máximo una duración de 2 años.
- El tiempo en el que se realizará el proyecto será en el horario de 15:30 a 21:30 de lunes a viernes.
- El sistema domótico está desarrollado solo para el género masculino. Quedará como una futura mejora el poder implementarlo para el género femenino.
- Algunos componentes de hardware son difíciles de conseguir dado al problema de las importaciones que tiene el país.
- Debido a la sensibilidad de algunos datos, se deberá guardar confidencialidad de dicha información.

Alcance

En esta etapa debemos garantizar que el proyecto incluya todo (y únicamente) el trabajo requerido para completarlo exitosamente. A continuación se detalla el alcance del proyecto.

- Crear un sistema informático capaz de mejorar la calidad de vida de las personas. Cabe aclarar que dicho sistema informático es un prototipo funcional, el cual debería estar suficientemente perfeccionado y cumplir con normas y leyes, para recién poder salir a producirse como un producto final.
- La solución a implementar proporcionará el hardware y software de base necesario, como así también el equipamiento necesario para la plataforma Web.
- Módulos del sistema Web:
 - Login del sistema
 - Portal de inicio
 - Resultado de mediciones
 - Dietas disponibles
 - E-commerce



- El sistema permitirá un correcto y eficiente registro y control de las dietas recomendadas de los distintos pH obtenidos.
- El software propuesto será accesible desde cualquier terminal, que cuente con un navegador web y una conexión al servidor donde estará hosteada la aplicación.
- Se propone el análisis, diseño, desarrollo y prueba del sistema. La implementación de hardware necesario para poder desplegar la aplicación no está en el alcance del presente proyecto.
- El sistema a desarrollar será totalmente modular, permitiendo la rápida y fácil adaptación de posibles futuras ampliaciones de funcionalidades.
- El sistema cumplirá con todos los estándares de seguridad propuestos por las prácticas de programación en Ruby on Rails.
- La conexión a internet es provista por un ISP externo, este servicio es ajeno a este proyecto.
- El software dispondrá de control de acceso (logs de registros), solicitando usuario y contraseña, lo cual permitirá acceder a información de acuerdo al perfil del usuario determinado por administrador de la aplicación.
- La implementación de este sistema no contempla tareas relacionadas a la carga de información del usuario en la base de datos.

Análisis FODA

La matriz FODA es una herramienta de análisis que puede ser aplicada a cualquier situación, producto, empresa, etc, que esté actuando como objeto de estudio en un determinado momento de tiempo.

Para comenzar un análisis FODA, se debe hacer una distinción crucial entre las cuatro variables por separado y determinar qué elementos corresponden a cada una. Estas cuatro variables son:

- Fortalezas: son las capacidades especiales con que cuenta la empresa, y que le permite tener una posición privilegiada frente a la competencia. Recursos que se controlan, capacidades y habilidades que se poseen, actividades que se desarrollan positivamente, etc.
- Oportunidades: son aquellos factores que resultan positivos, favorables, explotables, que se deben descubrir en el entorno en el que actúa la empresa, y que permiten obtener ventajas competitivas.
- Debilidades: son aquellos factores que provocan una posición desfavorable frente a la competencia, recursos de los que se carece, habilidades que no se poseen, actividades que no se desarrollan positivamente, etc.
- Amenazas: son aquellas situaciones que provienen del entorno y que pueden llegar a atentar incluso contra la permanencia de la organización.

Tanto las fortalezas como las debilidades son internas de la organización, por lo que es posible actuar directamente sobre ellas. En cambio las oportunidades y las amenazas son externas, y solo se puede tener injerencia sobre ellas modificando los aspectos internos.

Factores Externos Factores Internos	Fortalezas	Debilidades
	Oportunidades ESTRATEGIAS OFENSIVAS	ESTRATEGIAS ADAPTATIVAS
Amenazas	ESTRATEGIAS REACTIVAS	ESTRATEGIAS DEFENSIVAS

Imagen 9: Análisis FODA

En base a esto, se plantean para el sistema domótico lo siguiente:

Fortalezas:

- Construcción del primer producto nacional encargado de medir el pH en orina.
- Elaboración del producto con materiales de bajo costo.
- Posibilidad de realizar innovaciones en el producto por el equipo desarrollador.
- Conocimiento específico en electrónica y domótica del equipo desarrollador que permite realizar correcciones para su mejoramiento.
- Posibilidad de obtener resultados en tiempo real por parte del sistema.

Oportunidades:

- Aumento de enfermedades que requieran la realización de este estudio.
- Crear mayor interés de los clientes acerca de la importancia de la realización del estudio para el diagnóstico de una enfermedad.
- Construcción del sistema de diferentes maneras por sus factores tecnológicos variados.
- Aumento de canales para realizar la compra del producto.
- Disminución de precios de insumos para la creación del producto.

Debilidades:

- Surgimiento de nuevos requisitos de patentamiento que no estén incluidos en el sistema.
- Falta de publicidad del producto.
- El sistema domótico no permite su uso para personas de género femenino.
- Posibilidad de falla del sistema por falta de capacitación del usuario para ponerlo en funcionamiento.



Amenazas:

- Empresas de desarrollo de equipos domóticos ofreciendo un producto similar.
- Bloqueo en importaciones de productos para el armado del sistema.
- Mejores ofertas por parte de los competidores.
- Temas inflacionarios por la situación del país.
- La innovación en otros países de un sistema similar, que hagan que el producto quede fuera del mercado por ser antiguo.
- Surgimiento de nuevas leyes que hagan que el producto caduque.
- Existencia de sistemas para medir el pH en ambos géneros.

Una vez que se tienen registradas las principales fortalezas, oportunidades, debilidades y amenazas, se pueden formular estrategias alternativas que alineen los factores internos y externos:

- Estrategias Ofensivas: utilizan fortalezas para aprovechar oportunidades
- Estrategias Adaptativas: permiten superar debilidades aprovechando oportunidades
- Estrategias Reactivas: utilizan fortalezas para evitar o reducir el efecto de amenazas
- Estrategias Defensivas: permiten a la vez superar debilidades y evitar amenazas.

Estrategias Ofensivas:

- Utilizar la elaboración del producto con bajo costo para provocar un aumento de ventas por los diferentes canales.
- Poder implementar el primer sistema de producción nacional para medir el pH en orina, debido al aumento de enfermedades que requieran la realización de un estudio.
- A través del conocimiento en domótica y electrónica, crear un mayor interés de los clientes acerca de la importancia de la medición del pH en orina, para el diagnóstico de enfermedades.
- Elevar la producción del sistema, aprovechando la disminución del costo de la materia prima.

Estrategias Adaptativas:

- Debido al aumento de enfermedades que requieren el estudio de medición de pH en orina, el sistema podrá hacerse más conocido y lograr una promoción del mismo sin que la misma sea buscada por sus creadores.
- Lograr a través de la diversidad de factores tecnológicos y el bajo costo de insumos, la elaboración del sistema de medición de pH en orina de personas de género femenino.
- Debido al interés por prevenir enfermedades, los clientes utilizarán el sistema con amplia satisfacción por la posibilidad de obtener el resultado en tiempo real, aunque el mismo solo abarque al género masculino.



Estrategias Reactivas:

- Realizar la compra de materia prima para el armado del sistema, elaborando un amplio stock para paliar la posibilidad del bloqueo de importaciones que lleven a la falta de producción.
- Por la capacidad de innovación del equipo desarrollador, intentar mejoras en el sistema, para que el mismo sea digno de competir con sistemas con mismas cualidades.
- Colocar énfasis en la posibilidad de obtener resultados en tiempo real para poder aplacar la falta del sistema de medición de pH en orina de personas de género femenino.

Estrategias Defensivas:

- Mantener al equipo desarrollador informado de actividad de innovación del sistema, para poder efectuar modificaciones que superen la falencia de la no medición del pH en orina femenina, y que la competencia se apropie de nuestros clientes.
- Tener amplio stock y aumentar la publicidad del sistema para evitar que el mismo caduque, ya sea por no cumplir con requisitos puntuales de otros productos con sus mismas características, como así también que el sistema desarrollado no logre cumplir las expectativas de los clientes.
- Continuar la búsqueda de sistemas alternativos que permitan la medición del pH en orina para ambos géneros, manteniendo el bajo costo en sus insumos logrando así mejorar los resultados para posicionar nuestro sistema por sobre el de los competidores.



4 - SOLUCIÓN PROPUESTA

La propuesta central del presente proyecto de tesis fue mejorar la calidad de vida de los usuarios. Este fin, como respuesta a los ideales planteados en los procesos de ingeniería. Al ser este objetivo tan general, optamos por descifrar una solución que aporte a los fines del mismo, por lo cual desarrollamos el sistema que será a continuación explicado.

Para poder lograr este objetivo nos enfocamos en el equilibrio del pH del organismo para poder prevenir enfermedades. La solución propuesta de parte de la rama de la informática, consiste en aportar al usuario la facilidad de medir automáticamente el nivel de salud que posee su organismo a través del pH, reduciendo al mínimo la interacción que este deba realizar para obtener los mismos resultados.

Se debe tener en cuenta, que actualmente no existe un procedimiento automatizado para realizar reingeniería, solo un procedimiento totalmente manual. También se facilita la lectura de estos datos, proporcionando información clara y precisa y agilizando los procedimientos que el usuario deba realizar para seguir con una guía de pautas propuestas para equilibrar el estado de salud del organismo.

Esta solución informática está basada en reducir el accionar del usuario al mínimo evitando, en lo que sea posible, las fallas de control rutinario.

Se ideó un dispositivo de fácil uso, el cual deberá ser instalado y fijado definitivamente en el sanitario del hogar del usuario, y mantendrá comunicación constante con servidores alojados en la nube. Los pasos a realizar para darle un uso correcto al sistema serán los siguientes:

1. Cada vez que el mismo orine sobre el artefacto construido, este será capaz de obtener información relacionada con el pH, para ser más precisos, obtendrá el nivel de acidez o alcalinidad de su organismo.
2. Estos datos serán procesados y enviados a un servidor, por medio de una conexión a internet, y de un módulo capaz de realizar esta transacción.
3. El servidor destinado a este proyecto, será capaz de captar estos datos por medio de una aplicación web desarrollada en Ruby on Rails, procesarlos, y luego emitir un resultado basado en distintos procesos de control minucioso.
4. El usuario tendrá acceso ilimitado a estos datos, pudiendo consultar su historial de mediciones, recomendaciones (para poder equilibrar su pH si es que es necesario), y también tendrá acceso a listas de compras, de distintos productos, necesarios para seguir con las instrucciones propuestas por el sistema. Estas órdenes de compras serán enviadas a supermercados, para que estos puedan despachar los productos al hogar del usuario.
5. Una vez que los productos lleguen al hogar del usuario, éste tendrá todos los materiales necesarios para seguir las recomendaciones propuestas.

Si el usuario sigue estos pasos según indica el proyecto, podemos afirmar que reducirá notablemente sus posibilidades de contraer cáncer, basándose en la tesis expuesta por el Dr. Otto Heinrich Warburg.



Análisis de Porter

Amenaza de entrada de nuevos competidores

Con la existencia de ANMAT (Administración Nacional de Medicamentos, Alimentos y Tecnología Médica), la entrada de nuevos competidores al mercado requiere de una serie de cumplimientos en normas de calidad, eficacia y seguridad de los equipos desarrollados.

Por dicho motivo, el desarrollo de este producto encargado de medir el pH en orina, debe aprovechar este mercado tentador, no solo manteniendo y tratando de cumplir con las normativas establecidas por ANMAT, sino buscando la excelencia en el producto que se ofrece y la distinción del mismo.

Rivalidad de los Competidores

Existen varios competidores que brindan productos de buena calidad y que cumplen con las normativas de ANMAT, por lo que esto ocasiona una gran competencia entre los diferentes desarrolladores de productos. Los puntos más importantes a tener en cuenta a la hora de esta rivalidad son: costo del producto, calidad, servicio al cliente, publicidad, canales de venta, stock del producto, etc.

Por estos motivos, es de suma importancia, que el producto que se ofrece cuente con un costo accesible al público, que la calidad del mismo sea excelente, tenga una buena gestión en el marketing del producto, realizar un análisis de mercado para tener una estrategia de venta y así llegar a la mayor cantidad de usuarios, y por último, brindar un buen servicio de atención al cliente ante cualquier duda sobre el uso del mismo.

Poder de negociación de los proveedores

No existen muchos proveedores de insumos que son necesarios para la realización y armado del sistema domótico, por lo que este poder de negociación está del lado del proveedor.

Dicha situación es un problema, ya que se tienen que analizar los costos de los insumos de acuerdo a los volúmenes de pedidos que se realizan, los tiempos de demora, tiempos de reposición, precios entre proveedores, etc.

Poder de negociación de los compradores

Al haber muchos competidores que brindan productos de buena calidad, el poder de negociación de los compradores se aumenta, ya que se tienen muchas alternativas por las cuales se puede optar.

Estos compradores podrán elegir entre diferentes características por las cuales se decide entre un producto u otro (costo, calidad, tiempo de entrega, información que le brinda dicho producto, atención recibida, etc).

Por esto, debemos ser perspicaces y brindar al cliente seguridad en el producto que está adquiriendo, lo cual permitirá que nuevos clientes opten por el nuestro.



Amenaza de ingreso de productos sustitutos

Al haber competidores en el mercado, estos pueden ofrecer productos que puedan tomarse como sustitutos y que posean características innovadoras que hagan que los compradores los elijan. Además de estas características, hay que tener en cuenta los costos, calidad y que los equipos que se brindan sean los más eficaces. Entre estos productos sustitutos, podrían ser equipos desarrollados por un competidor, como también el servicio que puede brindar un bioquímico que ayude al control de la salud diaria de las personas.

Esto hace que el desarrollo de nuestro producto tenga que brindar características innovadoras, tener un bajo costo, alta calidad, etc.

Procedimiento lógico del sistema

Hardware

1. Cuando el usuario comienza el proceso para poder medir su pH, la primera interacción que debe realizar con el sistema, se da por medio del **sensor de ultrasonidos HC-SR04** (Imagen 10). El mismo detecta cuando una persona o un objeto se encuentra a 50 cm o menos de distancia. Cuando esta detección es positiva, la tarea del sensor de distancia es avisar a Arduino sobre esta medición, lo cual hará que automáticamente se ponga en funcionamiento el servomotor. Las mediciones se realizan periódicamente cada 500 milisegundos. El sensor está basado en un mecanismo que tiene de base un Transmisor (Tx) y un Receptor (Rx). Para poder detectar la distancia el procedimiento es el siguiente:
 - Activar el *Trig* del módulo. Esto significa que el sensor enviará un pulso de sonido ultrasónico por medio de Tx
 - Activar *Echo*. Esto es activar un reloj interno (*timer*), y medir el tiempo que transcurre desde que el *Trig* fue activado, hasta que *Echo* detecta el rebote de la onda de sonido.
 - Obteniendo este tiempo de respuesta, y considerando que la velocidad de propagación del sonido sobre el aire es de 343 m/s, se deben realizar cálculos matemáticos para obtener un resultado final.
La velocidad del sonido es de 340 metros por segundo. Esto se da si el sonido se propaga por el aire.

$$S = 340 \text{ m/s}$$

Esto es equivalente a:

$$S = 1/29 \text{ cm/us}$$

La distancia final, será el resultado de medir cuanto fue el tiempo que tardó el sensor en detectar el pulso de respuesta (*timer*) dividido 2. Dividimos por dos el tiempo, ya que la onda recorre el doble de la distancia entre el *Trig* y el objeto que está detectando, ya que la onda ultrasónica se dirige hacia el objeto, rebota, y se dirige hacia *Echo* recorriendo así una distancia * 2.

$$d = (\text{timer} / 2) / 29$$

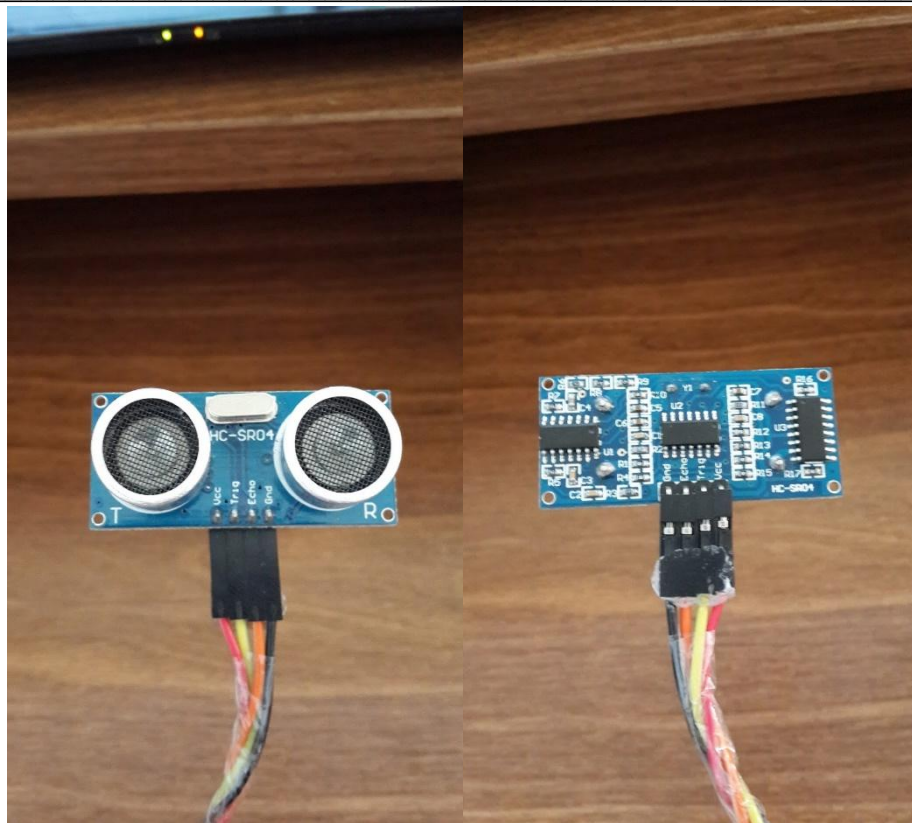


Imagen 10: Sensor Distancia HC-SR04 (Conexión)

2. El dispositivo en cuestión, va colocado en forma vertical sobre el inodoro del usuario. En las imágenes 11, 12 y 13 se pueden observar distintas partes:
 - **La base/soporte del dispositivo.** Es una estructura de metal diseñada para poder abarcar y proteger la gran mayoría de los dispositivos internos (entiéndase Arduino, fuente de alimentación, servo-motor y sensor de color). Además de protección sirve como sostén para la cinta transportadora, el servo y el sensor de color.
 - **La cinta transportadora.** Está pensada simplemente para poder conducir la tira reactiva desde su posición inicial hasta su destino, sin que esta pueda desviarse del camino deseado. La misma está construida con plástico.
 - **Servo-motor.** Su funcionamiento consiste en ser capaz de tomar la tira reactiva una vez que el usuario la introduzca en el dispositivo, y llevarla hasta que la misma pueda sobrepasar la cinta transportadora por aproximadamente 5 cm. Una vez que el usuario orina la tira reactiva - solamente en la parte que sobresale del dispositivo - el servomotor debe retraer la misma hasta que la parte donde se produce la medición (explicado posteriormente), se encuentra exactamente debajo del sensor de color. Luego de que el sensor realiza su tarea, el servo debe expulsar nuevamente la tira hacia abajo, para que la misma pueda caer al inodoro. En resumen, podemos dividir las tareas del servo en 3 partes (imágenes 11,12 y 13):

- i. Tomar la tira reactiva y empujarla hasta que sobresalga del dispositivo

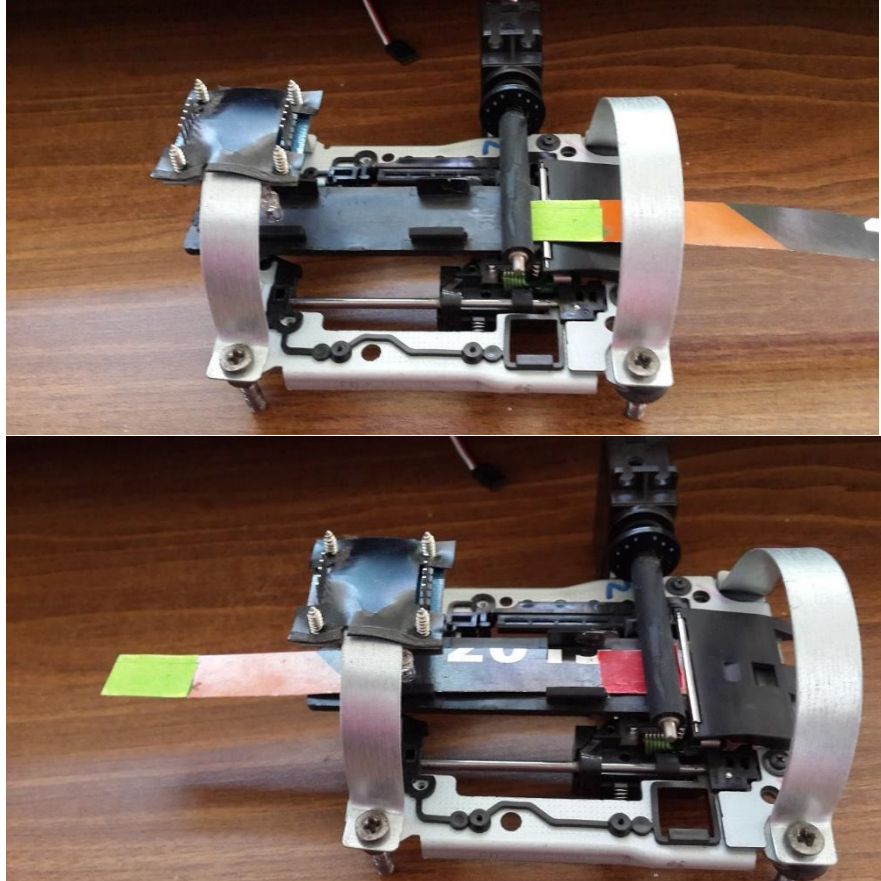


Imagen 11: Servo (Tomar tira reactiva)

- ii. Introducir nuevamente la tira para que pueda ser analizada

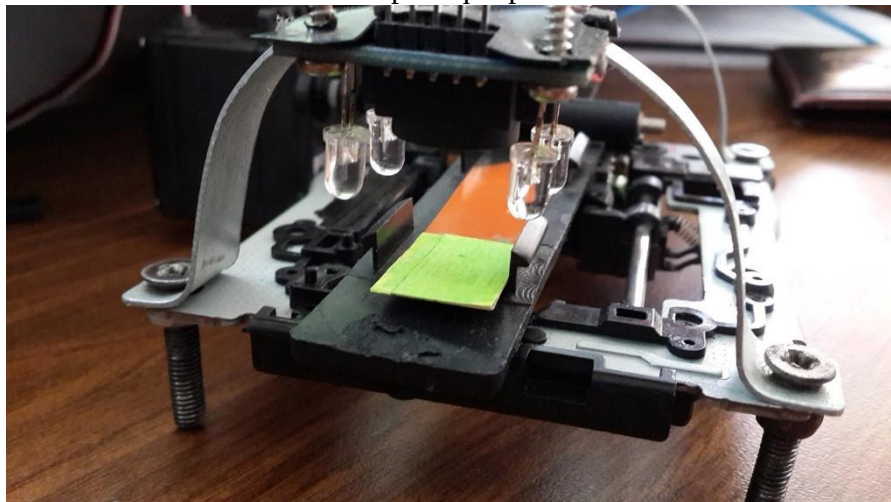


Imagen 12: Servo (Analizar tira reactiva)

iii. Expulsar completamente la tira.

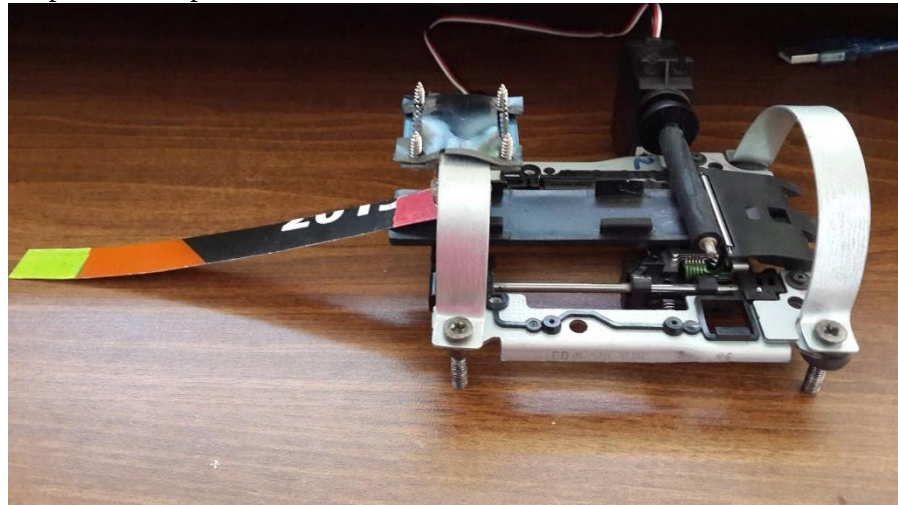


Imagen 13: Servo (Expulsar tira reactiva)

- **Sensor de color TCS3200.** Su función consiste en medir el color de la tira reactiva. Para realizar esto se basa en un sistema de filtros R-G-B (red, green, blue, en español rojo, verde y azul). La salida del sensor de color, que luego es interpretada por Arduino, especifica que cantidad de rojo, verde y azul detectó en la tira. Una vez obtenidos estos valores se puede deducir cual es el color presente. El sensor a su vez, posee cuatro LEDs que iluminan la tira reactiva para que la luz ambiente no interfiera con las mediciones. Una vez obtenidos los valores del RGB, Arduino es el encargado de transformar estos datos en un color final, para luego asignarle un pH.

El sensor cuenta con 2 pines dedicados a la medición del color. Estos pines son S2 y S3.

Para poder realizar las mediciones en RGB, los pasos a seguir son los siguientes:

- i. Cuando el pin S2 y S3 tienen un pulso bajo, el sensor representa en una cantidad numérica la cantidad de color rojo que se detectó en los fotosensores. Por lo cual el estado inicial de la medición es:

S2 = LOW

S3 = LOW

- ii. Para poder medir la cantidad de color azul, se debe establecer el pin S3 en **HIGH**.

S2 = LOW

S3 = HIGH

- iii. Para medir la cantidad de verde, se debe establecer el pin S2 en **HIGH**.

S2 = HIGH

S3 = HIGH

Todos estos procedimientos de cambio de pulso dan como resultado un vector de números, los cuales representan la cantidad (desde 0 a 255), reflejados en color: rojo, azul y verde que se detecta en la tira reactiva.



3. Una vez obtenido el color de la tira reactiva, por medio de los pasos especificados anteriormente, el siguiente objetivo es realizar la comunicación con el servidor hosteado en la nube, para así enviarle los parámetros que este necesita para poder modificar el perfil del usuario. Para poder cumplir con esto, fue indispensable utilizar el módulo Wifi ESP-8266.

Para establecer la conexión y posterior comunicación se utilizaron los siguientes comandos:

- AT: Configura en módulo ESP8266 en estado de trabajo, a la espera de comandos.
- AT+CWMODE = 1: Establece el modo en que el módulo Wifi se conecta a la red. En este caso es necesario conectarlo como STA (station). Todos los componentes que pueden conectarse a una red por medio de Wifi hacen referencia a una STA. Todos las STA están equipadas con un controlador de interfaz de red Wifi. (WNICs). Las estaciones Wifi se pueden clasificar en dos categorías:
 - Wireless access points: normalmente wireless routers, son estaciones base para las redes Wifi. Estos transmiten y reciben radiofrecuencias de los dispositivos disponibles en la red, para que estos se puedan comunicar entre sí.
 - Clients: pueden ser dispositivos móviles, como laptops, celulares, etc.

En el caso del módulo Wifi ESP-8266 se debe realizar la configuración para que actúe como un STA client. De la documentación del módulo podemos extraer lo siguiente:

WiFi Mode	AT+CWMODE? AT+CWMODE=1 AT+CWMODE=2 AT+CWMODE=3	Query STA AP BOTH
-----------	---	----------------------------

Imagen 14: Modos de conexión ESP-8266

El módulo es capaz de funcionar correctamente con los comandos AT+CWMODE = 1 (STA), AT+CWMODE = 2 (AP) y AT+CWMODE = 3(AP+STA).

Debido a los requerimientos del sistema y del presente proyecto, lo correcto sería utilizar el modo Wifi como un STA, por lo especificado anteriormente.

- AT+RST: Se resetea el módulo con la intención de eliminar posibles transacciones pendientes, o errores acarreados.
- AT+CWJAP="SSID","Password": Se establece una conexión con el router Wifi, especificando id y contraseña del punto de acceso.
- AT+CIPMODE = 1: Establece el modo de transmisión. Nos permite establecer dos tipos distintos, transmisión transparente (1) y transmisión no transparente (2).
 - El modo de transmisión 2 establece un protocolo de corrección de errores llamado RLP (radio link protocol). Se utiliza para realizar un control de errores en la comunicación entre STA y el AP.
 - El modo de transmisión 1 establece un protocolo sin corrección de errores sobre la interfaz.

En base a las pruebas realizadas sobre el módulo Wifi, se llegó a la conclusión de que el modo de transmisión más eficiente es el de transmisión transparente, ya que no se observaron errores de comunicación en las pruebas, y el tiempo de respuesta fue considerablemente mayor.

- AT+CIPSTART = <type>, <address>, <port>: establece la conexión con el servidor a donde deseamos mandar los datos, especificando el tipo de conexión (TCP o UDP), la dirección del servidor, y el puerto por donde se realizará la comunicación. En nuestro caso, el tipo de comunicación es TCP, la dirección es: <https://healty-tesis.herokuapp.com>, y el puerto es el 80.

- AT+CIPSEND: setea el módulo Wifi para comenzar a mandar los datos al servidor especificado en el comando anterior.
- GET /user/1/measurement/" + varcolor + " HTTP/1.1: donde se especifica varcolor se hace referencia al color obtenido durante la medición del pH en pasos anteriores.
Host: healty-tesis.herokuapp.com: especifica el host a donde se realiza el envío de datos. Estos dos últimos comandos conforman la consulta al servidor.

El módulo Wifi ESP-8266 se puede observar en la siguiente figura.

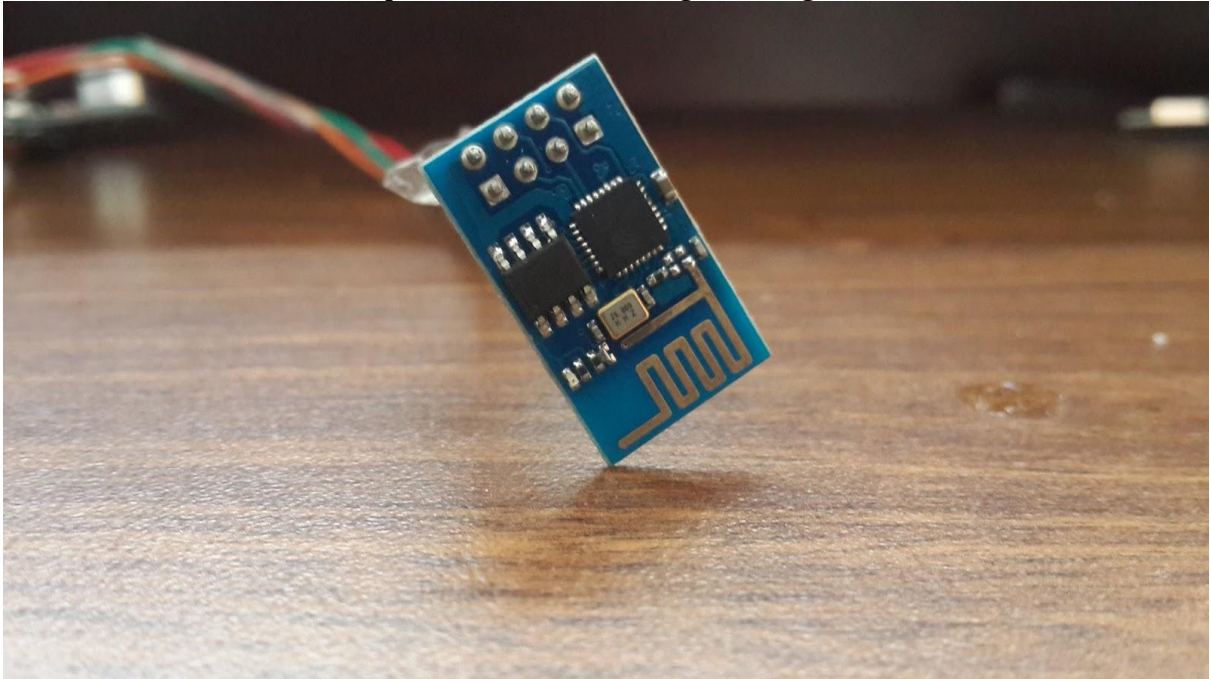


Imagen 15: Modulo ESP-8266

Conexiones físicas realizadas

- Sensor de color TCS3200

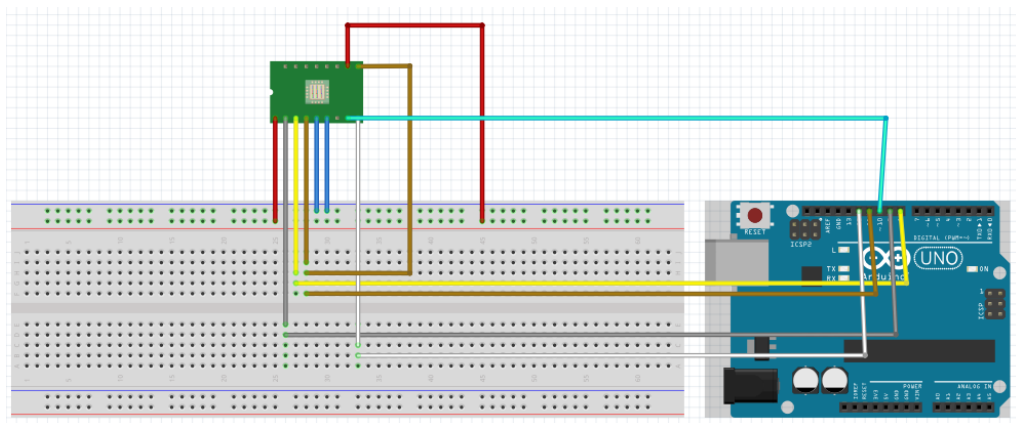


Imagen 16: Conexión Física del Sensor de Color

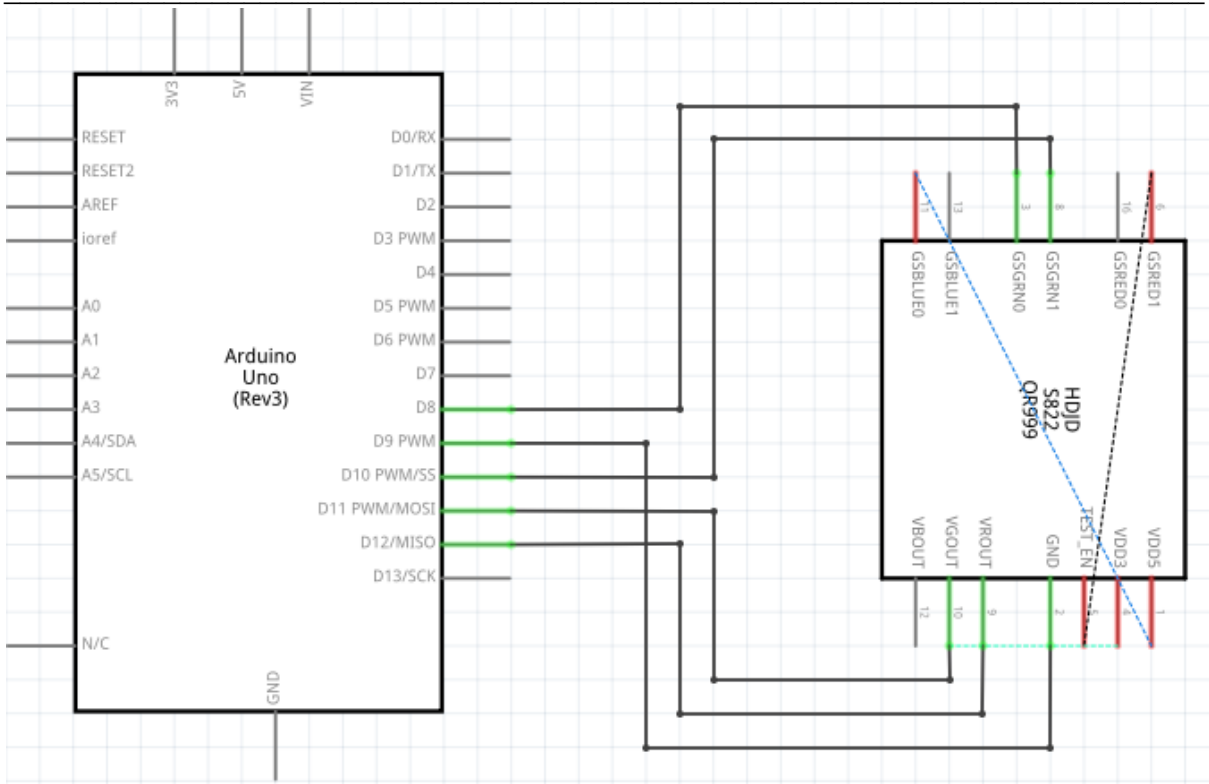


Imagen 17: Conexión Física del Sensor de Color (Circuito)

- Sensor de ultrasonidos HC-SR04

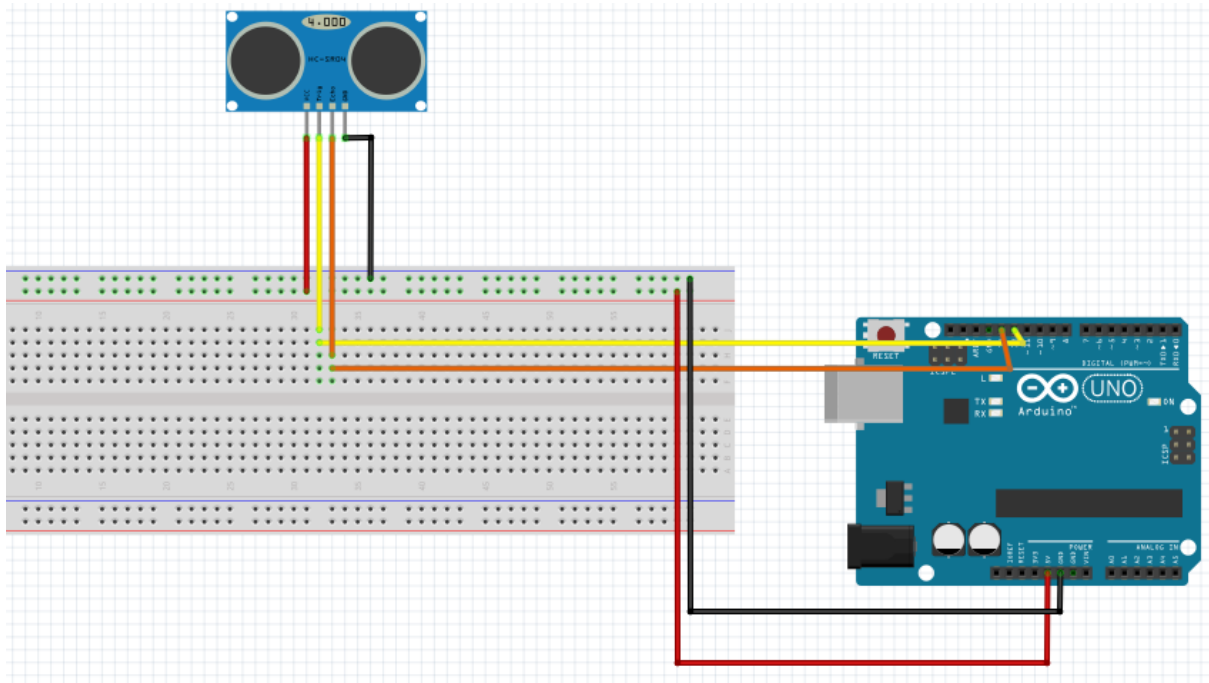


Imagen 18: Conexión Física del Sensor de Ultrasonidos HC-SR04

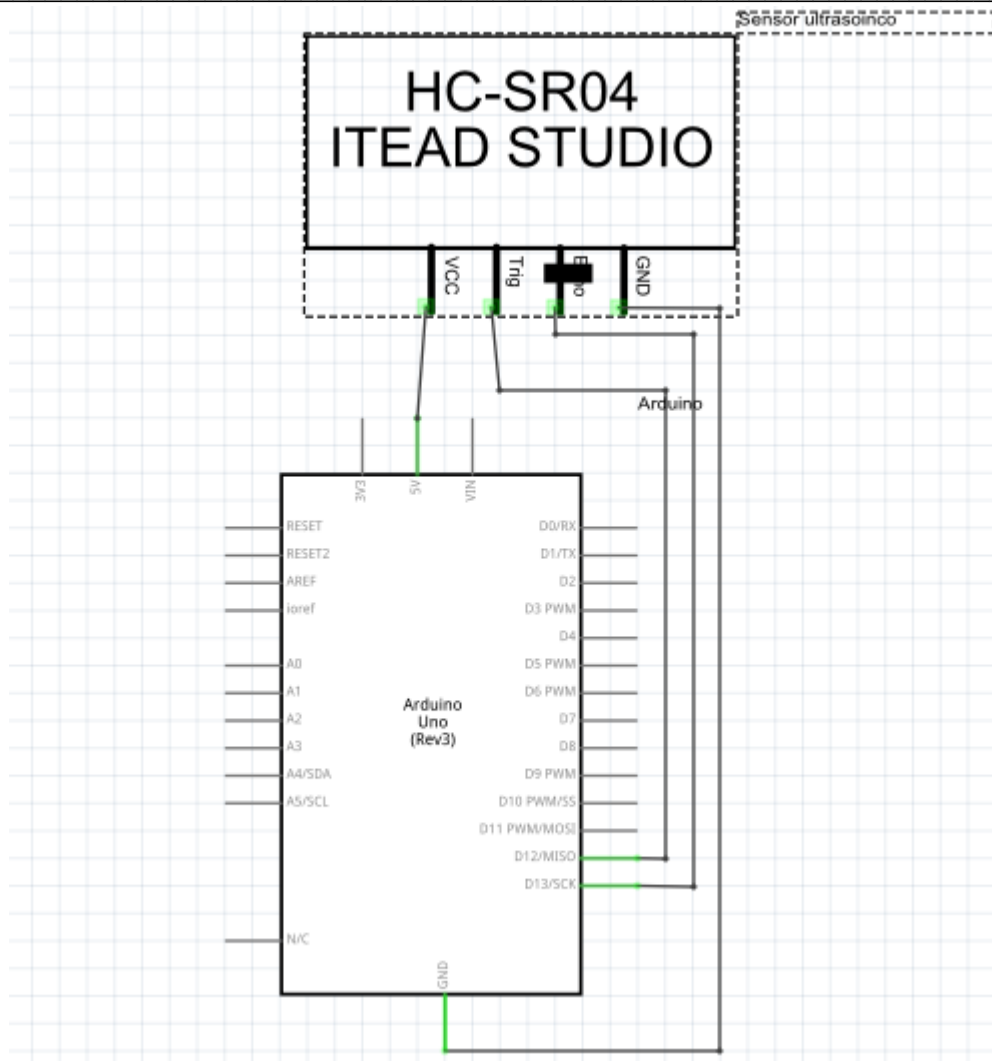


Imagen 19: Conexión Física del Sensor de Ultrasonidos HC-SR04 (Circuito)

- Módulo Wifi ESP-8266

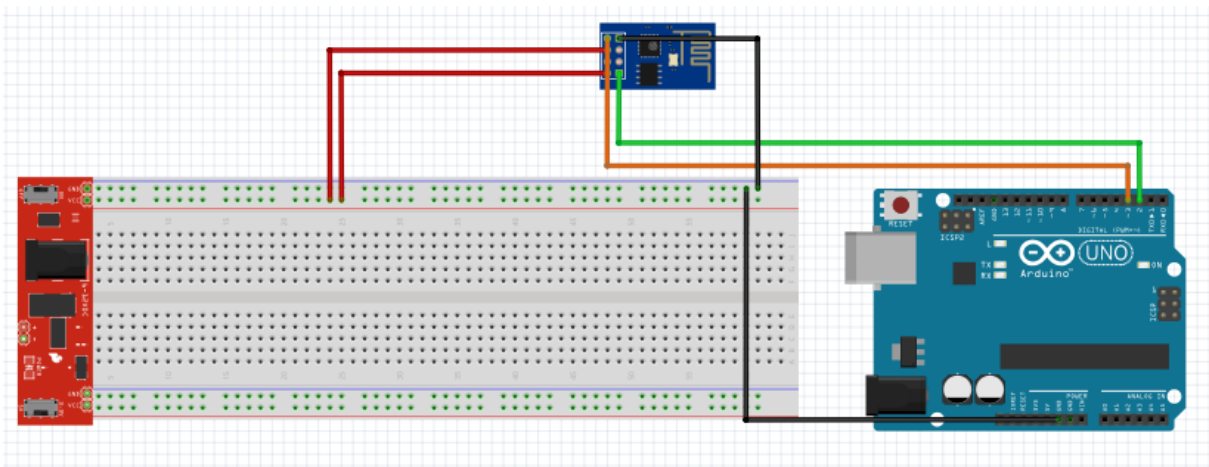


Imagen 20: Conexión Física del Módulo Wifi ESP-8266

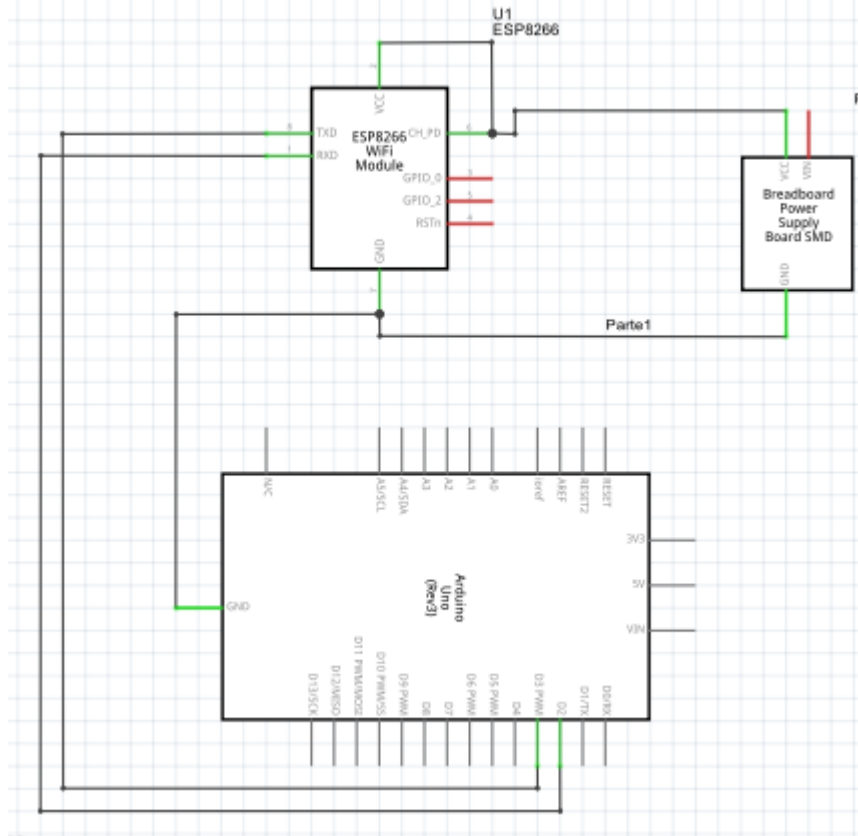


Imagen 21: Conexión Física del Módulo Wifi ESP-8266 (Circuito)

- Servo

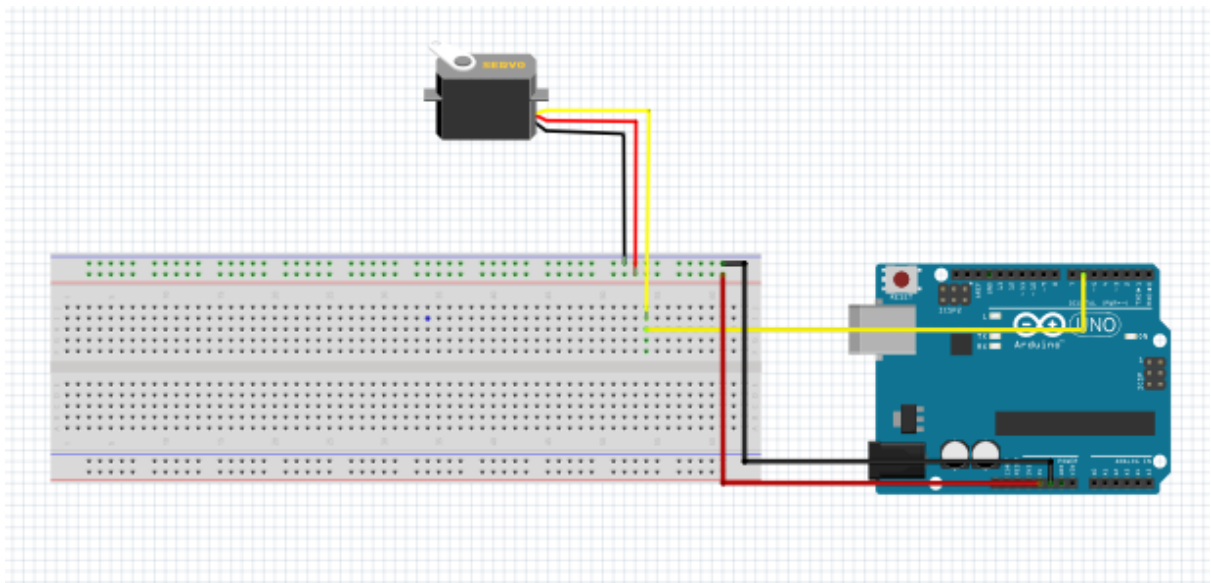


Imagen 22: Conexión Física del Servo

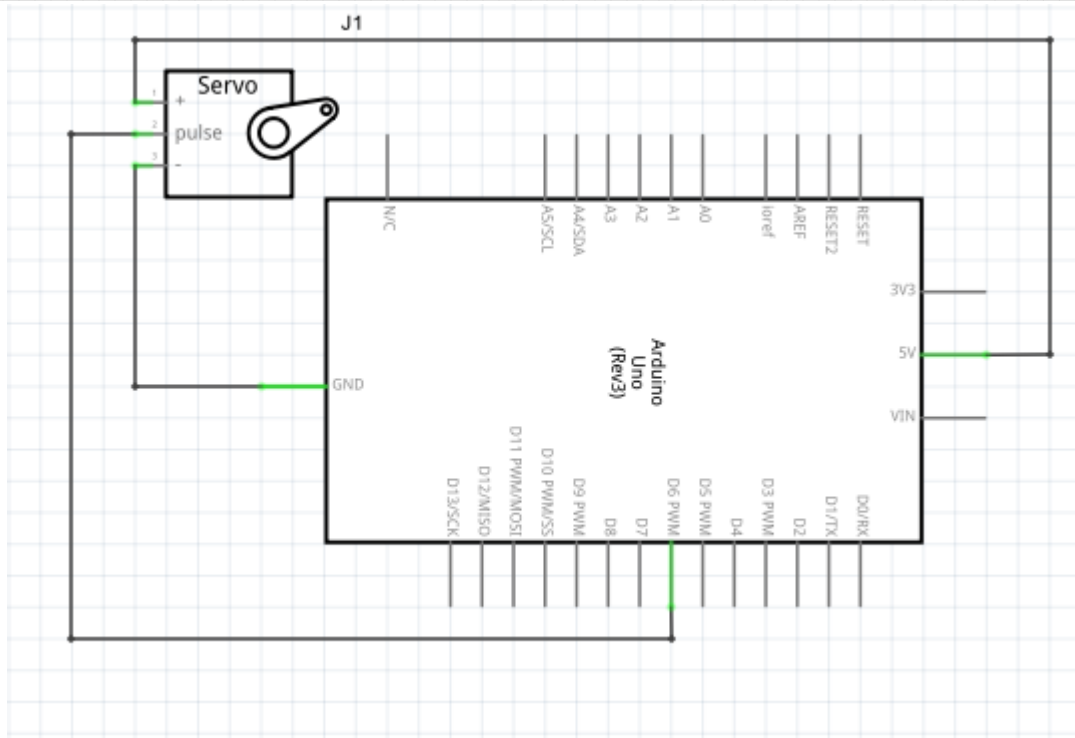


Imagen 23: Conexión Física del Servo (Circuito)

- Sistema completo

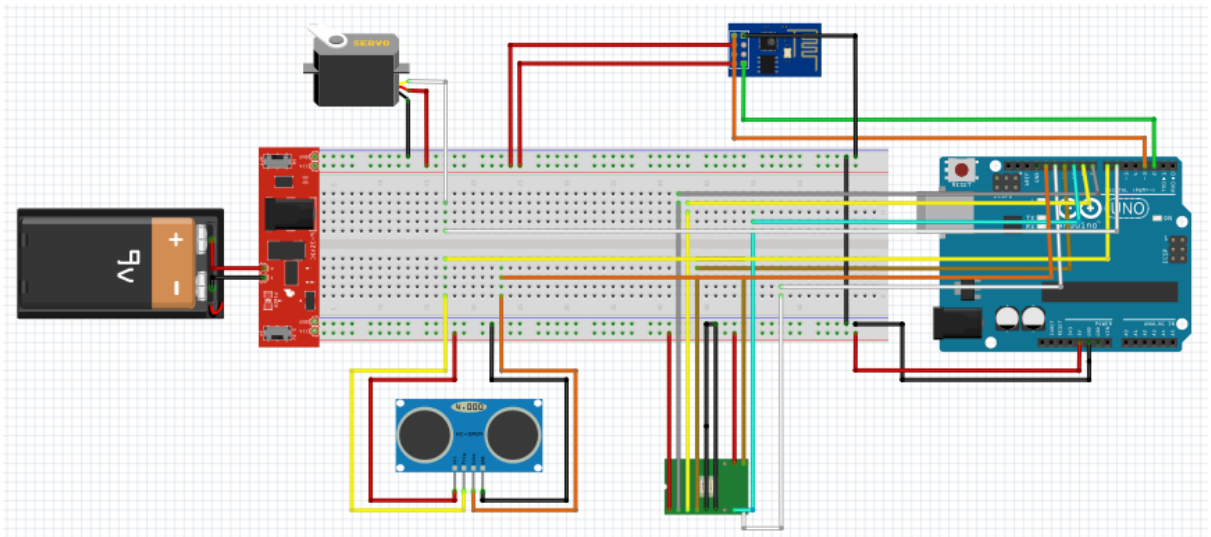


Imagen 24: Conexión Física del Sistema Completo

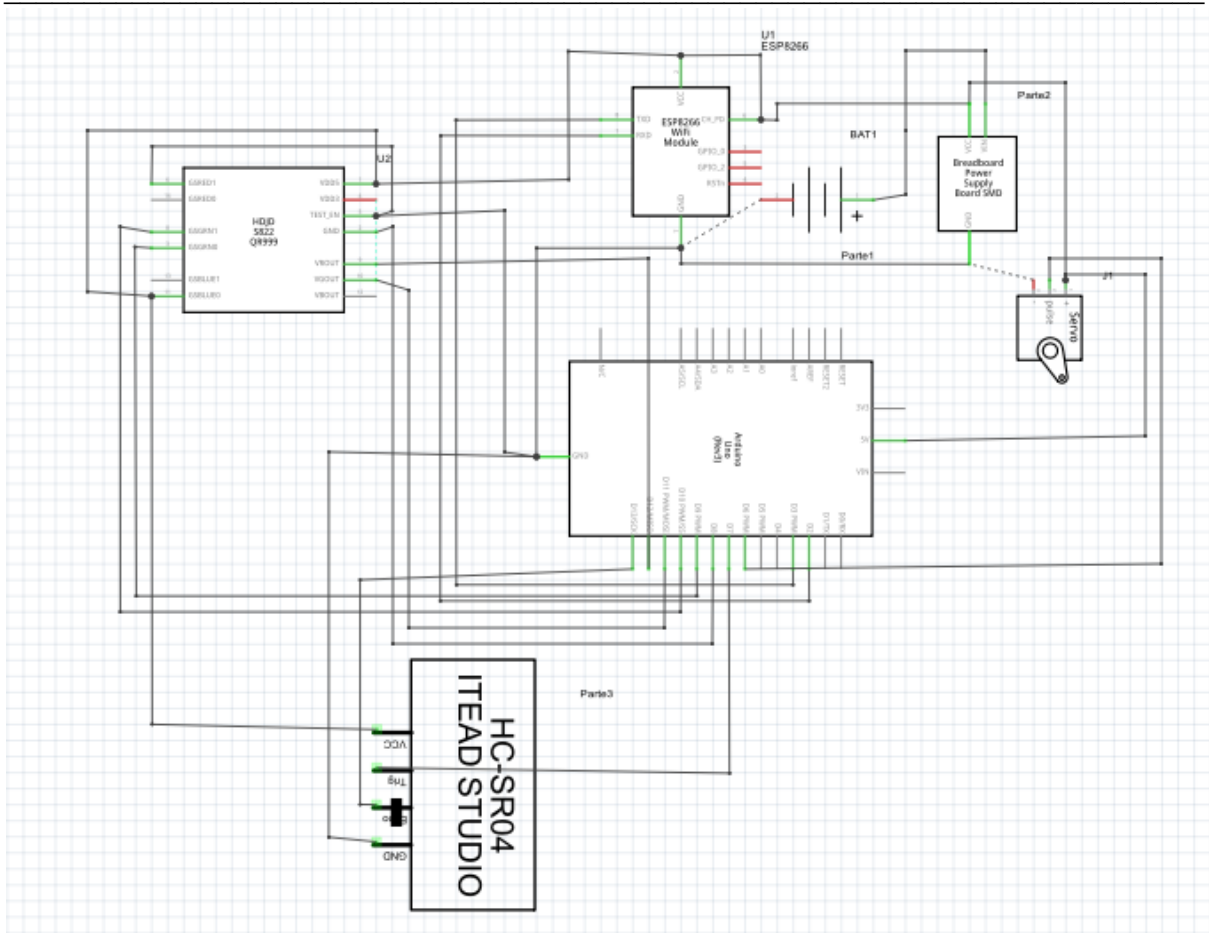


Imagen 25: Conexión Física del Sistema Completo (Circuito)

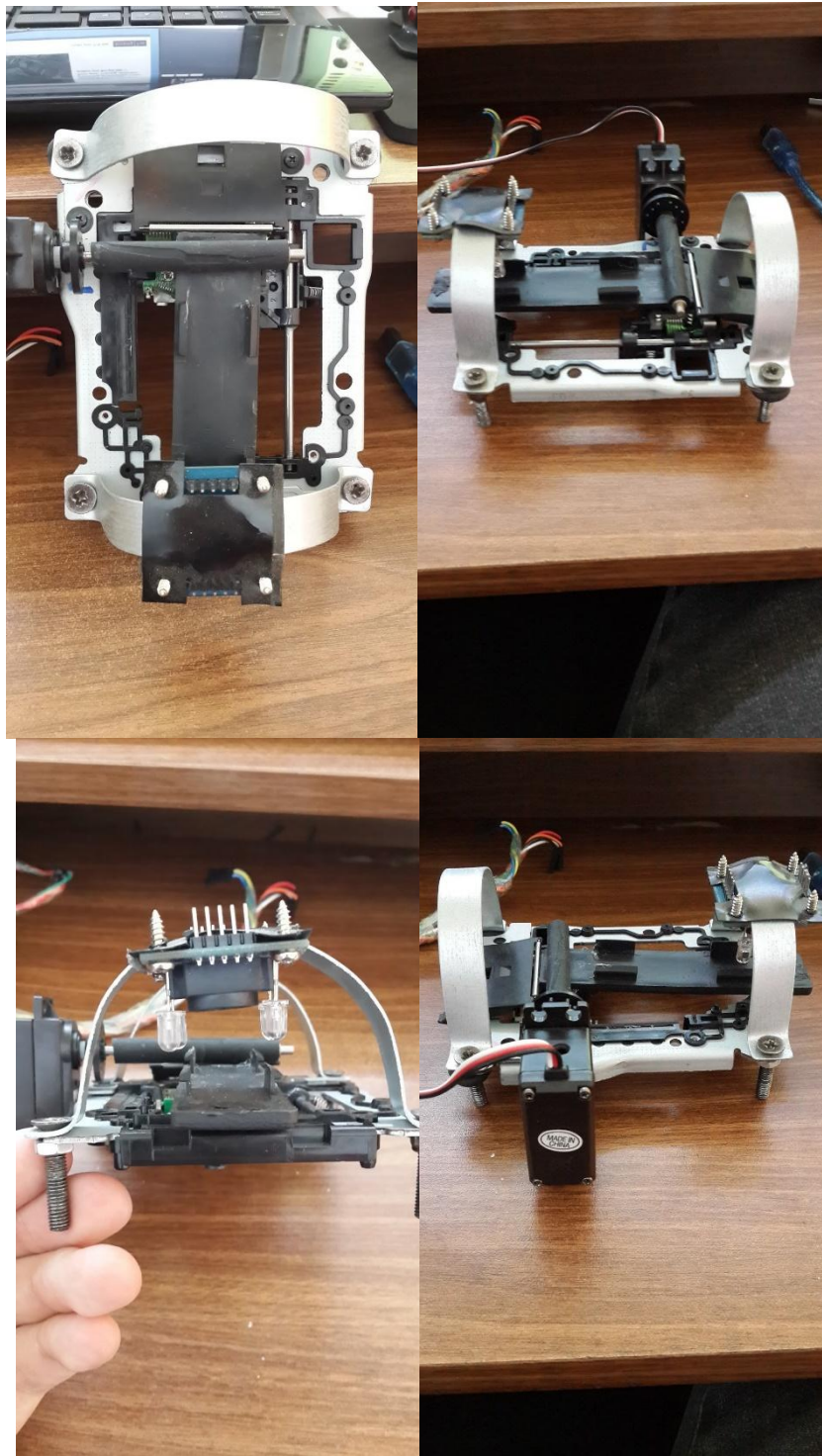


Imagen 26: Sistema Completo



Software

El software está desarrollado sobre una plataforma web basada en Ruby on Rails. Se utilizó el IDE Sublime Text para su codificación.

Este sistema posee las siguientes partes:

- Login: para el ingreso al sistema, se tendrá un control de acceso para el usuario final. Dicho acceso será mediante un nombre de usuario o email y una contraseña que será provista por el sistema. Dicha clave podrá ser modificada por el usuario.
- Inicio: contará con una página de inicio, la cual será la primera vista que observe el usuario en el sistema una vez autenticado al mismo. En dicha página se podrá ver una presentación de la aplicación.
- Resultados de mediciones: en esta sección, el usuario final podrá ver un historial de sus mediciones con el sistema domótico. Cada registro le mostrará si el paciente cuenta con un pH ácido o alcalino.
- Dietas disponibles: en dicha sección, se podrán observar las dietas disponibles según el pH obtenido por el paciente. Cada una de las dietas especificará los ingredientes que son necesarios para realizarla.
- E-Commerce: cuando se selecciona la dieta disponible y al paciente le faltan ingredientes para realizarla, podrá seleccionar aquellos ingredientes y ejecutar una orden de compra a un supermercado asociado. Las comunicaciones deben realizarse por medio de un protocolo seguro, para esto se debe utilizar https. Este protocolo de seguridad no se implementará en este proyecto, pero debe considerarse para futuras investigaciones.

A continuación explicaremos los pasos a realizar por el sistema, basándose en el paradigma Modelo-Vista-Controlador. Para poder lograr una mejor comprensión del lector, la explicación de este módulo se dividirá en dos partes:

1. Punto de vista del usuario:
 - a. El primer paso para acceder al sistema será ingresar un nombre de usuario y una contraseña. Los mismos serán otorgados al usuario al adquirir el dispositivo. La contraseña se encuentra encriptada por razones de seguridad. Se utilizó OpenSSL::Cipher para realizar la encriptación. Para realizar el logueo del usuario se utilizó la gema Devise.
 - b. Una vez creada la sesión del usuario, el siguiente paso será verificar las dietas disponibles para el pH especificado. Para esto se configuró la vista “home”. La misma está compuesta de dos partes:
 - i. La primera consiste en comunicarle información sobre el pH del usuario obtenido desde su última medición, y al mismo tiempo se le recomienda una dieta aleatoria. Esta dieta aleatoria depende del pH especificado. Un pH menor a 7 tendrá como resultado mostrar aleatoriamente dietas alcalinas. Un pH mayor a 7 mostrara dietas ácidas. Un pH igual a 7 mostrara dietas que contengan un pH equilibrado. La vista se observa en la imagen 27.

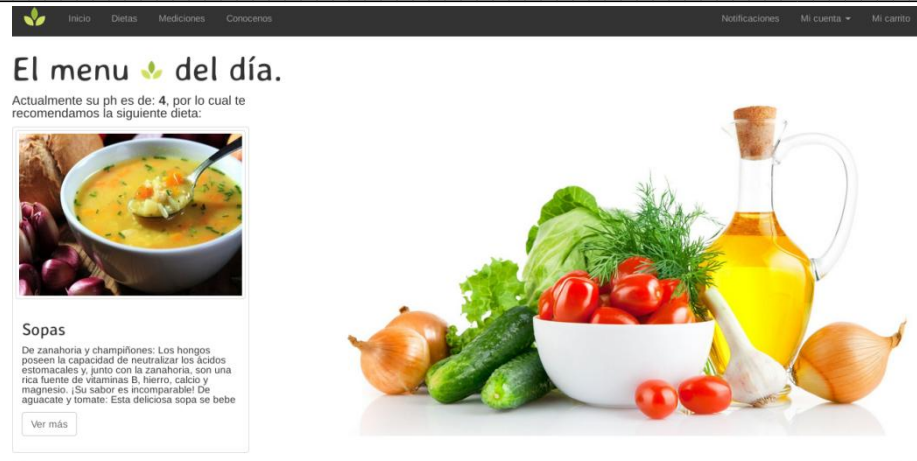


Imagen 27: Vista “Home” del Sistema

- ii. La segunda etapa muestra un total de 8 dietas específicas para el pH del usuario. Cada una con una imagen de vista previa. Al hacer click sobre la imagen, se despliega la receta con más detalle. Esto se observa en la imagen 28.

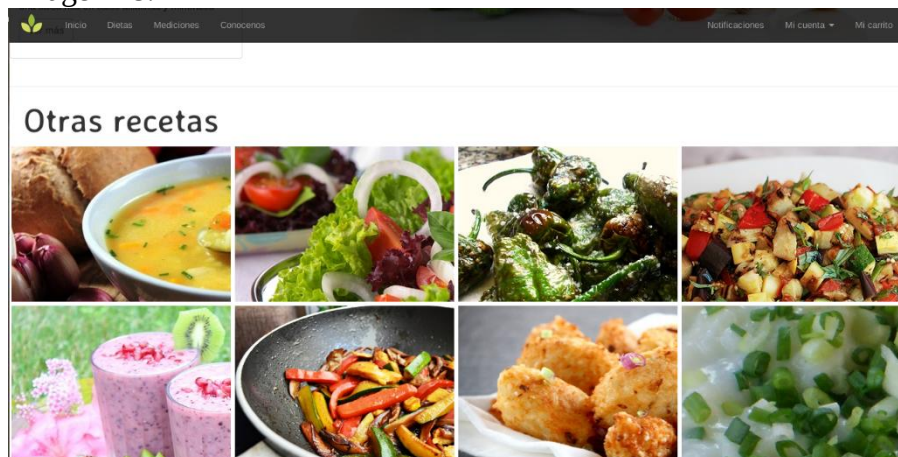


Imagen 28: Dietas Específicas según pH

- c. Seleccionada la dieta que el usuario desee, se desplegará la misma con más detalle. La misma incluirá tanto la receta, una foto de la misma y un listado de ingredientes con su respectiva cantidad y precio. Por cada ingrediente se incluirá un símbolo representativo de una compra, para que el usuario al hacer click en el mismo tenga acceso a la sección de compras. Esto se observa en la imagen 29.



Imagen 29: Detalle de una dieta seleccionada.

- d. Cada vez que se seleccione el símbolo del carro de compras, el usuario será redirigido a un detalle del ingrediente, el cual le permitirá seleccionar la cantidad deseada del mismo. Una vez establecida, el siguiente paso es clicar el botón “Agregar al carrito”. Esto se observa en la imagen 30.



Imagen 30: Agregar producto al carrito de compras

- e. El usuario ingresa ahora al carro de compras que se observa en la imagen 31. Se pueden presentar tres situaciones distintas:
- Es la primera vez que se genera una orden de compra: se crea la primera orden de compra y se añade el detalle (producto más cantidad)
 - Ya existía una orden de compra no pagada: se agrega el detalle a la orden de compra.
 - Existen órdenes de compra, pero todas están pagadas: se creará una nueva orden de compras con el detalle enviado.



Imagen 31: Carrito de Compras



- f. Una vez que el usuario ingresa al carro de compras, por cada orden de compras se presentan tres opciones: pagar, editar y eliminar.
- Pagar:** se muestra el detalle de la orden, el cual incluye una imagen del producto, el nombre, la cantidad y el precio respectivamente, junto con la opción de eliminar. Una vez terminada de detallar la orden de compra, se puede observar un formulario con los datos de contacto (1), datos de envío (2), y datos del medio de pago (3). Esto se ve en la imagen 32.
 - Email - Teléfono - Nombre
 - Dirección - Ciudad - Código Postal - Provincia
 - Número de tarjeta de crédito - Código de seguridad - Fecha de vencimiento - Nombre del titular - Tipo de documento - Número de documento - Proveedor de tarjeta - Cantidad de cuotas
 - Editar:** permite el acceso a la edición de la orden de compras. Tanto para modificar los datos que se asignaron previamente con respecto al contacto, medio de pago, envío. O para modificar los productos que están incluidos en la misma.
 - Eliminar:** remueve de la base de datos la orden, junto con todos los objetos que dependen de esta

The image shows two screenshots of a web application's checkout process. The top screenshot displays the 'Datos de contacto' (Contact Data) form, which is the first step in a three-step process (1, 2, 3). The form includes input fields for Email, Telefono (with a sub-field for 'Codigo de area incluido'), and Nombre (with a sub-field for 'Nombre completo'). A 'Siguiete' button is at the bottom right. The bottom screenshot shows the 'Datos de envío' (Shipping Data) form, which is the second step. It includes input fields for Direccion, Ciudad,Codigo postal, and Provincia. A 'Next' button is at the bottom right. Both screenshots show a shopping cart at the top with items: Zanahoria (7 units, \$35), Champiñon (19 units, \$0), and Cebolla Morada (1 unit, \$4). The navigation bar at the top of both screenshots includes links for Inicio, Dietas, Mediciones, Conocenos, and user account options.



The screenshot shows a web interface for payment methods. At the top, there is a navigation bar with links: Inicio, Dietas, Mediciones, Conocenos, Datos de contacto, Datos de envío, Medios de pago, Notificaciones, Mi cuenta, and Mi carrito. The main section is titled 'Medios de pago' and contains the following fields:

- Numero de tarjeta de credito:** A text input field containing '5031 7557 3453 0604'.
- Codigo de seguridad:** A text input field containing '123'.
- Fecha de expiracion:** Two dropdown menus showing '1' and '2016'.
- Nombre de titular:** A text input field containing 'APRO'.
- Tipo de documento:** A dropdown menu showing 'DNI'.
- Numero de documento:** A text input field containing '12345678'.
- Proveedor de tarjeta:** A dropdown menu.
- Cuotas:** A dropdown menu.

Below the input fields, there is a row of logos for various payment methods: Mercado Pago, VISA, Mastercard, American Express, PayPal, Neteller, Skrill, Nativia, Previsión, and Rapipago.

Imagen 32: Pago de una Orden de Compra

Según el patrón que siguen la numeración de la tarjeta de crédito que ingresa el usuario, se puede determinar a que ente proveedor pertenece. Distinguiendo esto podemos discriminar las promociones que ofrecen y la cantidad máxima de cuotas, junto con sus intereses.

- g. Una vez enviado el formulario a completar, en caso de que la transacción sea exitosa, se enviará toda la información de la orden de compras a la página del supermercado seleccionado. A modo de ejemplo de este proyecto de tesis, se diseñó una página básica de compraventa para poder representar lo que sería un sistema de comunicación con los mismos. La página web del supermercado captará esta información, la procesa y emite una respuesta, ya sea un mensaje de éxito o un mensaje de error, especificando detalladamente la causa. El pago realizado con la tarjeta de crédito será procesado en la página web Healty, la transacción que se realizará con el ente proveedor de servicios de venta online tendrá como resultado una respuesta de procesamiento de pago, ya sea exitosa o no. La información que se enviará al servidor del supermercado solamente, será la de confirmación de la compra, junto con los productos que incluye la misma, y su respectiva cantidad.

Documentación Técnica

El sistema web está documentado mediante la metodología de desarrollo ágil “SCRUM”. Se seleccionó esta metodología, porque es un proceso iterativo e incremental y favorece la adaptación al cambio, al estar desarrollando domótica integrada a un sistema de cloud computing, fue muy importante que la metodología seleccionada permita una fácil adaptación a los diferentes requerimientos que surjan. Hay que tener en cuenta que si bien este es un proyecto que tiene una salida práctica, incluye una investigación preliminar de una importancia muy relevante.

El producto avanzó en una serie de Sprints, que son períodos que variaron entre 2 y 4 semanas de duración. Estos requisitos fueron capturados como elementos de una lista llamada “Product Backlog” y dichos Sprints tienen como objetivo concretar un subconjunto conocido

como “*Sprint Backlog*”. Como el equipo de proyecto está conformado por pocas personas, esta metodología es ideal para la realización del mismo.



Imagen 33: Metodología SCRUM

Hay diferentes roles que son necesarios para la ejecución efectiva de la metodología SCRUM:

- **Product Owner:** representa la voz del cliente. Se asegura de que el equipo esté trabajando correctamente (desde una perspectiva de negocios). Escribe las historias de usuario y las prioriza. Las mismas son ubicadas en el *Product Backlog*.
- **Scrum Master:** su función es remover impedimentos que compliquen al equipo para entregar los objetivos del sprint.
- **Equipo:** tiene la obligación y responsabilidad de crear y entregar el producto. Estas personas que componen el equipo posiblemente tengan habilidades diferentes.

El equipo de desarrollo del presente proyecto cuenta con solamente dos personas, por lo que se hará una división tanto de tareas como de roles entre las mismas, a fin de poder cubrir las necesidades que implica este paradigma de proyecto.

Otro punto a tener en cuenta, es que esta metodología es iterativa e incremental y en el diagrama de GANTT del proyecto no se observan estas iteraciones, sino que se ven las diferentes etapas, aplicando la metodología de desarrollo en cascada. Esto se debe a poder observar con mayor simpleza y entendimiento este diagrama.

Entre lo que son reuniones del equipo del proyecto, existen diferentes alternativas, las cuales se detallan a continuación:

- **Daily Scrum:** son reuniones diarias que no duran más de 15 minutos y no están hechas para solucionar problemas. Esta ayuda a evitar otras reuniones innecesarias. Se realizan 3 preguntas durante este tipo de reunión:
 - i. ¿Qué se realizó ayer?
 - ii. ¿Cuáles son las tareas para hoy?
 - iii. ¿Hay obstáculos en el trabajo?



Dentro de nuestro equipo de proyecto, este tipo de reuniones se realizan todos los días en los cuales se realiza el trabajo final. Se respetan las 3 preguntas y a su vez, se ve si no va a haber un desvío en los tiempos planificados, por imprevistos dentro del proyecto.

- **Sprint Planning:** se realizó al inicio de cada sprint. Se seleccionó qué trabajo se debía realizar en base a la priorización que se hizo en el *Product Backlog*. El equipo sólo tomó los ítems más importantes, comprometiéndose a completarlos sin tomar más de lo que es capaz de realizar. Nos resultó conveniente dejar un sobrante de tiempo, entre la finalización de un *sprint* y el siguiente.

Esta reunión fue de mucha utilidad para el equipo del proyecto, ya que se pudo determinar por cada integrante qué actividades debería realizar, su duración y enfoque solo en las mismas.

- **Sprint Retrospective:** Tuvo como fin ajustar los puntos conflictivos y mejorar la adopción de la metodología. Esta reunión se realizó periódicamente para poder observar su funcionalidad. Se realizó luego de la terminación del *sprint* y participó todo el equipo. Su duración fue de 15 a 30 minutos.

La misma se realizó para poder observar el funcionamiento del sistema web y cómo interactuaba con el sistema domótico. Se pudieron detectar algunas falencias que fueron solucionadas antes de iniciar el siguiente *sprint*.

Como objetivo general, se propuso no extender el tiempo de las reuniones, con el fin de ser concisos con los errores que fueron surgiendo. A modo de facilitar la concreción de este objetivo, las reuniones fueron realizadas al finalizar la jornada de desarrollo y siempre de pie.

Para acceder a cada reunión los informes debían estar previamente definidos, y en la gran mayoría de los casos documentados. El resultado de estas reuniones fue muy fructífero, ya que fue donde se producía la comunicación más concisa y eficaz por parte de los integrantes del proyecto.

Product backlog

Se trata como un documento de alto nivel para todo el proyecto. Es el conjunto de todos los requisitos de proyecto, el cual contiene descripciones genéricas de funcionalidades deseables. Representa el que va a ser construido en su totalidad.

Cada ítem de la lista del *Product Backlog* se denomina Historia de Usuario, con características visibles y escritas en un lenguaje ameno para el usuario.



#	Backlog Item	Estimación (Complejidad)
1	Login Usuario	3
2	Logout Usuario	3
3	Obtención de dietas	3
4	Creación de orden de compra	13
5	Modificación de orden de compra	13
6	Eliminación de orden de compra	8
7	Visualización de órdenes de compra	5
8	Pago de orden de compra	21
9	Alta de medición	13
10	Alta de Usuario	8
11	Edición de Usuario	8
12	Baja de Usuario	5
13	Alta de Dietas	13
14	Edición de Dietas	13
15	Baja de Dietas	8
16	Visualización de Mediciones	8
17	Envío de orden de compra a supermercado	21
18	Obtención de respuesta del supermercado	8

Tabla 1: Product Backlog del Sistema



Historias de Usuario

1	Login de Usuario
Como usuario del sistema, necesito poder ingresar al sistema para hacer uso de las funcionalidades del mismo. ESTIMACIÓN: 3 COMPLEJIDAD: BAJA	
<u>Criterios de Aceptación:</u> • El usuario ya debe encontrarse registrado en el sistema. • Si la contraseña no coincide con el nombre de usuario, el sistema mostrará el mensaje “Error en credenciales (nombre de usuario y/o contraseña incorrectos)”. • Si el usuario no recuerda su contraseña, tendrá la opción de recuperar su clave.	

2	Logout de Usuario
Como usuario del sistema, necesito poder salir del mismo para que mi sesión no quede abierta. ESTIMACIÓN: 3 COMPLEJIDAD: BAJA	
<u>Criterios de Aceptación:</u> • El usuario se debe encontrar logueado en el sistema.	

3	Obtención de Dietas
Como paciente o familiar, necesito poder obtener el listado de dietas según el pH para elegir una y luego realizarla. ESTIMACIÓN: 3 COMPLEJIDAD: BAJA	
<u>Criterios de Aceptación:</u> • El usuario se debe encontrar logueado. • Las dietas se deben encontrar cargadas en el sistema. • Las dietas que se muestran al paciente y a sus familiares, deben estar filtradas según el pH obtenido desde el sistema domótico.	



4	Creación de Orden de Compra
Como paciente, necesito poder crear una orden de compra, para obtener los ingredientes necesarios para realizar la dieta. ESTIMACIÓN: 13 COMPLEJIDAD: MEDIA	
<u>Criterios de Aceptación:</u> • El usuario se debe encontrar logueado. • Para la creación de una nueva orden de compra, si o si, debe haber un detalle de un ingrediente junto a su cantidad. • No se permiten campos vacíos para la creación de una orden de compra junto a sus detalles. • Los ingredientes asociados a una dieta, deben encontrarse precargados desde el servicio que provee el supermercado. • Se deberán mostrar mensajes desde el sistema cuando se realice una creación, o si ocurriese un error.	

5	Modificación de Orden de Compra
Como paciente, necesito poder modificar una orden de compra, para editar o agregar ingredientes necesarios para realizar la dieta. ESTIMACIÓN: 13 COMPLEJIDAD: MEDIA	
<u>Criterios de Aceptación:</u> • El usuario se debe encontrar logueado. • Para la modificación de una orden de compra, si o si, debe haber una orden de compra creada junto a su detalle. • No se permiten campos vacíos para la edición de una orden de compra junto a sus detalles. • Los ingredientes asociados a una dieta deben encontrarse precargados, desde el servicio que provee el supermercado. • Si existiese una orden de compra no pagada, la misma podrá ser modificada. • Se deberán mostrar mensajes desde el sistema, cuando se realice una modificación o si ocurriese un error.	



6	Eliminación de Orden de Compra
Como paciente, necesito poder eliminar una orden de compra que no se encuentre pagada, para que no quede un registro sobre este documento. ESTIMACIÓN: 8 COMPLEJIDAD: BAJA	
<u>Criterios de Aceptación:</u> • El usuario se debe encontrar logueado. • Para la eliminación de una orden de compra, si o si, debe haber una orden de compra creada junto a su detalle. • Todas las órdenes de compra podrán ser eliminadas junto a sus detalles (excepto aquellas que ya se encuentran en un estado Pagado). • Se deberán mostrar mensajes desde el sistema cuando se realice una eliminación o si ocurriese un error.	

7	Visualización de Órdenes de Compra
Como paciente, necesito poder visualizar las órdenes de compra realizadas, para tener un seguimiento de las mismas. ESTIMACIÓN: 5 COMPLEJIDAD: BAJA	
<u>Criterios de Aceptación:</u> • El usuario se debe encontrar logueado. • En este listado se debe tener las opciones para editar, eliminar o pagar una orden de compra. • Si no existen órdenes de compra asociadas al paciente, el sistema deberá mostrar un mensaje “No existen Órdenes de Compras Asociadas”.	



8	Pago de Orden de Compra
Como paciente, necesito poder realizar el pago de una orden de compra, para que los ingredientes comprados en el supermercado lleguen a mi domicilio. ESTIMACIÓN: 21 COMPLEJIDAD: ALTA	
<u>Criterios de Aceptación:</u> • El usuario se debe encontrar logueado. • El modo de pago de la orden de compra debe ser realizada mediante tarjeta de crédito. • La validación de los datos de la tarjeta de crédito se deben hacer de forma online mediante Mercado Pago. • Solo se deben poder pagar aquellas órdenes de compra que se encuentren en el estado de Generadas y tengan como mínimo un detalle. • Para poder realizar el pago de forma correcta, primero se deben completar los datos de usuario, luego de envío y al final el medio de pago. • No se permiten campos vacíos a la hora de hacer el pago de la orden de compra. • La fecha de vencimiento de la tarjeta de crédito, debe ser mayor al día que se quiera realizar el pago. • El código de seguridad debe tener solo 3 caracteres numéricos. • La tarjeta de crédito debe tener solo 16 caracteres numéricos. • Cuando se presente un error en el pago de la orden de compra, el sistema deberá mostrar un mensaje indicando el tipo de error.	

9	Alta de Medición
Como paciente o familiar, necesito poder ver reflejada una nueva medición en el sistema web, para observar el valor de pH y seleccionar una dieta acorde al mismo. ESTIMACIÓN: 13 COMPLEJIDAD: MEDIA	
<u>Criterios de Aceptación:</u> • El usuario se debe encontrar habilitado. • La nueva medición deberá tener una fecha de realización, el usuario asociado al equipo y el valor de dicho análisis.	



10	Alta de usuario
Como usuario administrador del sistema, necesito poder registrar usuarios, para hacer uso de las funcionalidades del sistema. ESTIMACIÓN: 8 COMPLEJIDAD: MEDIA	
<u>Criterios de Aceptación:</u> • El nombre de usuario debe tener el formato de la dirección de correo electrónico del usuario. • El correo electrónico debe ser único. • El usuario debe tener asociado un ID representativo del sistema domótico. • La contraseña debe contener letras y números, y ser al menos de 8 caracteres y con un máximo de 32. • Cuando se realiza la confirmación de la contraseña, si no son iguales, el sistema deberá mostrar un mensaje de alerta “Las contraseñas no son iguales”. • No pueden dejarse campos sin completar. • Si ocurre un error en el registro, el sistema deberá brindar un mensaje de alerta al usuario, informando donde se encuentra su error y una descripción del mismo.	

11	Edición de usuario
Como usuario administrador del sistema, necesito poder editar los atributos de los distintos usuarios del sistema, para poder corregirlos según distintas necesidades. Como usuario paciente/familiar necesito poder editar mis datos de usuario. ESTIMACIÓN: 8 COMPLEJIDAD: MEDIA	
<u>Criterios de Aceptación:</u> • El usuario se debe encontrar logueado. • El usuario debe tener el rol de administrador para editar cualquier usuario. • Si se desea editar el nombre de usuario se debe ingresar una dirección de correo electrónico válida en el mismo. • La dirección de correo no debe ser repetida. • No puede ingresar campos vacíos. • Si se modifica el ID del sistema domótico, el nuevo ID debe estar asociado a un dispositivo perteneciente al usuario	



12	Baja de usuario
Como usuario administrador del sistema, necesito poder dar de baja a los usuarios, para mantener la base de datos actualizada y para evitar el acceso de usuarios sin autorización. ESTIMACIÓN: 5 COMPLEJIDAD: BAJA	
<u>Criterios de Aceptación:</u> • El usuario administrador se debe encontrar logueado. • Si se elimina un usuario con el rol de paciente, se deben eliminar los familiares asociados al mismo • Si los familiares están asociados a dos o más pacientes no se eliminarán sus registros, a menos que se eliminen todos los pacientes a los cuales se encuentran vinculados. • Al eliminar un usuario, se eliminarán también todas las órdenes de compra asociadas al mismo, junto con sus respectivos detalles.	

13	Alta de dietas
Como usuario administrador del sistema, necesito poder crear nuevas dietas y asociarla a ingredientes, para que los usuarios pacientes puedan acceder a las mismas. ESTIMACIÓN: 13 COMPLEJIDAD: MEDIA	
<u>Criterios de Aceptación:</u> • El usuario debe estar logueado • Los ingredientes asociados a la dieta, deben estar disponibles en el supermercado vinculado con Healty. • Ningún campo puede estar vacío. • El usuario que da de alta a la dieta, debe ser administrador del sistema	



14	Edición de dietas
Como usuario administrador del sistema, necesito poder editar los distintos atributos pertenecientes a una dieta, para poder corregir errores en las mismas o actualizarlas. ESTIMACIÓN: 13 COMPLEJIDAD: MEDIA	
<u>Criterios de Aceptación:</u> • El usuario debe estar logueado • La dieta debe estar previamente creada. • No se permiten campos vacíos. • Debe estar asociada a por lo menos un ingrediente. • El usuario que modifica la dieta debe ser administrador del sistema	

15	Baja de dietas
Como usuario administrador del sistema, necesito poder eliminar una dieta para que los pacientes y sus familiares puedan observar el listado disponible de las mismas. ESTIMACIÓN: 8 COMPLEJIDAD: MEDIA	
<u>Criterios de Aceptación:</u> • Solo un usuario administrador puede dar de baja a una dieta. • La dieta debe estar previamente cargada.	

16	Visualización de mediciones
Como usuario del sistema, necesito poder visualizar las mediciones asociadas a un usuario. ESTIMACIÓN: 8 COMPLEJIDAD: MEDIA	
<u>Criterios de Aceptación:</u> • El usuario debe estar logueado • Los usuarios con el rol de familiar, solo podrán visualizar las mediciones de los pacientes asociados al mismo. • Los usuarios con el rol de pacientes, solo podrán visualizar sus mediciones. • Para poder visualizar una medición de pH, la misma debe estar previamente cargada. • Si no existe ninguna medición, se debe mostrar un mensaje indicando que se debe	



realizar la primera medición para hacer un uso correcto del sistema.

17	Envío de orden de compra a supermercado
Como usuario, necesito enviar órdenes de compras a los supermercados de los ingredientes de las dietas que elegí.	
ESTIMACIÓN: 21 COMPLEJIDAD: ALTA	
<u>Criterios de Aceptación:</u> • El usuario debe estar logueado • Debe existir una conexión a internet. • Los campos de la orden de compras deben estar previamente validados y sin errores. • La respuesta previa de MercadoPago debe ser de éxito en la transacción. • No se permiten campos vacíos en la orden de compra. • La orden de compra debe ser enviada en formato JSON.	
18	Obtención de respuesta de supermercado
Como usuario, necesito obtener una respuesta de éxito o de error de parte del supermercado en donde fue enviada la orden, para estar seguro, de que la orden fue recibida y de que está en camino.	
ESTIMACIÓN: 8 COMPLEJIDAD: MEDIA	
<u>Criterios de Aceptación:</u> • El usuario debe estar logueado • La orden de compra debe haber impactado en el sistema del supermercado • La orden de compra debe ser recibida en formato JSON • El mensaje de respuesta del supermercado debe estar en formato JSON	

[Sprint backlog](#)

El *sprint backlog* es el subconjunto de requisitos que serán desarrollados durante el siguiente sprint. Al definir el *sprint backlog*, se describe cómo el equipo va a implementar los requisitos durante el sprint.

Según lo planificado, el desarrollo del Sistema Healty tiene una duración de aproximadamente 77 días hábiles. Se realizará todo el desarrollo del mismo en 4 (cuatro) *Sprints*. Se puede observar en la tabla 2, un resumen de las historias de usuario por *Sprint* y una duración estimada tanto en días hábiles como en semanas.



Número de Sprint	Duración Aprox (días)	Duración Aprox (semanas)	Historias de Usuario Asociadas
1	11	2	1,2,9,10,11,12
2	13	3	3,13,14,15,16
3	19	4	4,5,6,7,8
4	10	2	17,18

Tabla 2: Resumen de Historias de Usuario

Para mayor grado de detalle, en la tabla 3 se tienen las tareas que se realizarán por cada historia de usuario, asignándole un recurso humano y el esfuerzo necesario para poder cumplir con la tarea. La prioridad de las historias de usuario están relacionadas al *Sprint* asignado.

ID Historia	Prioridad	Sprint	Historia/Tareas	Esfuerzo (hs)	Asignado
1	1	1	Login Usuario	12	Miguel
			• T1. Implementar Devise	3	
			• T2. Verificar seguridad	3	
			• T3. Pruebas con diferentes tipos de usuarios	2	
			• T4. Diseño y codificación de vistas	4	
2	2	1	Logout Usuario	9	Miguel
			• T1. Verificar seguridad	3	
			• T2. Pruebas con diferentes tipos de usuarios	2	
			• T3. Diseño y codificación de vistas	4	
3	2	2	Obtención de dietas	18	Facundo
			• T1. Implementar actualización de ingredientes y stock según supermercado.	4	
			• T2. Implementar protocolo de comunicación con supermercado.	4	
			• T3. Verificación de filtros de dietas según pH.	2	
			• T4. Verificación de acceso según rol	2	
			• T5. Pruebas	2	
			• T6. Diseño y codificación de vistas	4	
4	3	3	Creación de Orden de Compra	30	Facundo
			• T1. Establecer proceso lógico de creación de órdenes de compra.	5	
			• T2. Diseño y codificación de las vistas	15	
			• T3. Pruebas	10	
5	3	3	Modificación de Orden de Compra	30	Miguel
			• T1. Establecer proceso lógico de modificación de órdenes de compra.	5	
			• T2. Diseño y codificación de las vistas	15	
			• T3. Pruebas	10	
6	3	3	Eliminación de Orden de Compra	18	Miguel
			• T1. Establecer proceso lógico de	5	



			eliminación de órdenes de compra.		
			• T2. Diseño y codificación de las vistas	5	
			• T3. Pruebas	5	
			• T4. Restringir acceso a la eliminación de orden de compra según rol.	3	
7	3	3	Visualización de Órdenes de Compra	18	Miguel
			• T1. Restringir acceso a la vista según rol.	5	
			• T2. Diseño y codificación de las vistas	5	
			• T3. Pruebas	8	
8	3	3	Pago de Orden de Compra	60	Facundo
			• T1. Creación de la cuenta en Mercadopago.	1	
			• T2. Implementación de API Mercadopago (server-side)	30	
			• T3. Implementación de API Mercadopago (client-side)	5	
			• T4. Implementación de Firma Digital	1	
			• T5. Diseño y codificación de vistas	9	
			• T6. Verificación de seguridad	5	
			• T7. Encriptación de datos	1	
			• T8. Pruebas	8	
9	1	1	Alta de Medición	30	Miguel
			• T1. Implementar protocolo de comunicación entre Arduino y Healty.	10	
			• T2. Establecer lógica del procesamiento de datos.	6	
			• T3. Implementación de seguridad	6	
			• T4. Pruebas	8	
10	1	1	Alta de Usuario	18	Facundo
			• T1. Diseño y Codificación de vistas	10	
			• T2. Pruebas	4	
			• T3. Restricción de acceso	4	
11	1	1	Edición de Usuario	18	Facundo
			• T1. Diseño y Codificación de vistas	10	
			• T2. Pruebas	4	
			• T3. Restricción de acceso	4	
12	1	1	Baja de Usuario	12	Facundo
			• T1. Diseño y Codificación de vistas	5	
			• T2. Pruebas	5	
			• T3. Restricción de acceso	2	
13	2	2	Alta de Dietas	30	Miguel
			• T1. Diseñar y codificar el sistema de archivos	5	
			• T2. Diseño y codificación de vistas	5	
			• T3. Implementar actualización de ingredientes según supermercado.	3	
			• T4. Implementar protocolo de comunicación con supermercado.	10	
			• T5. Pruebas	5	
			• T6. Restricción de acceso	2	
14	2	2	Edición de Dietas	30	Miguel
			• T1. Implementación del sistema de	3	



			archivos		
			• T2. Diseño y codificación de vistas	5	
			• T3. Implementar actualización de ingredientes según supermercado.	5	
			• T4. Implementar protocolo de comunicación con supermercado.	10	
			• T5. Pruebas	5	
			• T6. Restricción de acceso	2	
15	2	2	Baja de Dietas	18	Facundo
			• T1. Diseño y codificación de vistas	10	
			• T2. Restricción de acceso	5	
			• T3. Pruebas	3	
16	2	2	Visualización de Mediciones	24	Facundo
			• T1. Diseño y codificación de vistas	16	
			• T2. Restricción de acceso	5	
			• T3. Pruebas	3	
17	4	4	Envío de Orden de compra a un supermercado	60	Miguel
			• T1. Implementación del sistema de verificación de respuesta exitosa en Mercadopago.	15	
			• T2. Implementación del protocolo de comunicación entre Healty y Supermercado.	15	
			• T3. Verificación de seguridad	10	
			• T4. Diseño y codificación de vistas	12	
			• T5. Restricción de acceso	5	
			• T6. Pruebas	3	
18	4	4	Obtención de respuesta de supermercado	30	Facundo
			• T1. Implementación del protocolo de comunicación entre Supermercado y Healty.	15	
			• T2. Verificación de seguridad	3	
			• T3. Diseño y codificación de vistas	10	
			• T4. Pruebas	2	

Tabla 3: Resumen de Tareas por Historia de Usuario

Diagrama Entidad – Relación

El Diagrama Entidad-Relación (DER) asociado al sistema web Healty, fue generado por la librería Rails-erd y se puede observar en la imagen 34

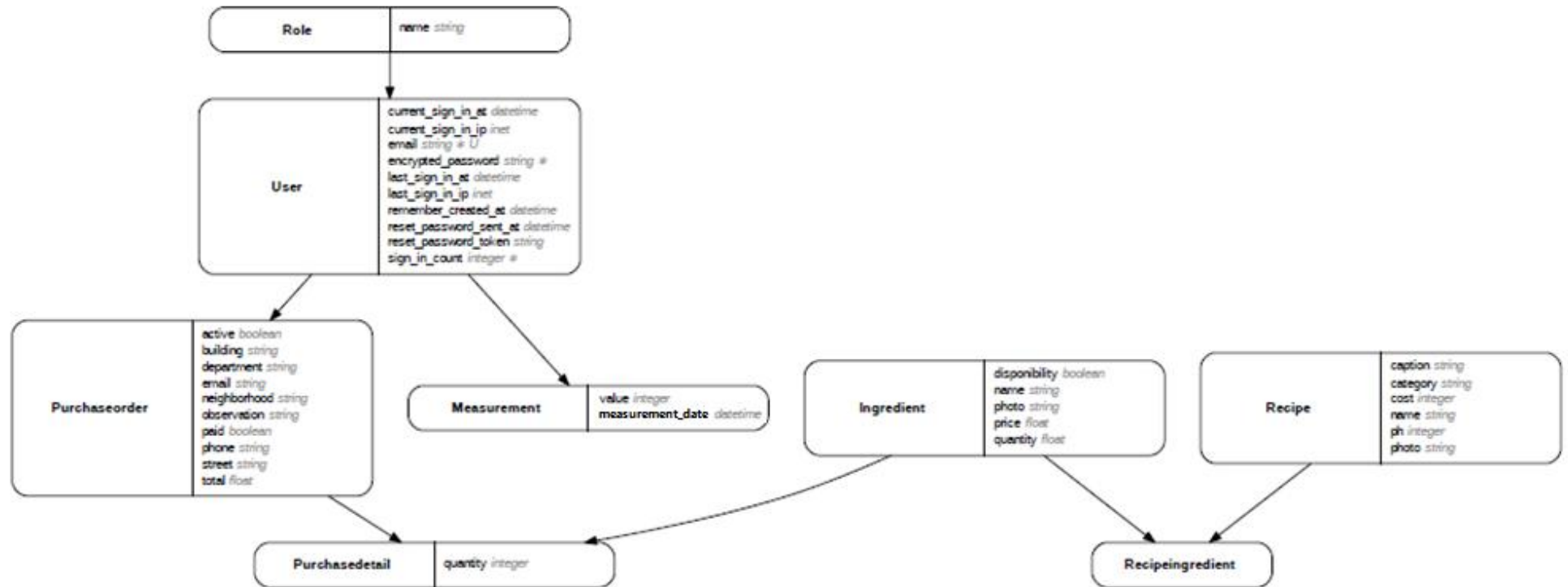


Imagen 34: Diagrama Entidad Relación (DER)



La solución a implementar en este proyecto final, está asociada a un esquema de base de datos que permite la interacción con un único supermercado. La tabla “Ingredientes” se basará en los códigos pertenecientes a dicha entidad, por lo que un cambio en los mismos se verá reflejado en la base de datos local. La obtención de datos del supermercado, se realizará a través de una consulta en formato JSON, la cual devolverá los registros de la tabla “Productos” del mismo.

Formulación del Proyecto

Como se mencionó en capítulos anteriores, para la formulación del proyecto se utilizará la guía de buenas prácticas (PMBOK). En esta sección se explicarán las diferentes áreas de conocimientos que serán utilizadas para la gestión del proyecto.

Alcance del Proyecto

En esta etapa debemos garantizar que el proyecto incluya todo (y únicamente) el trabajo requerido para completarlo exitosamente.

Se recopilarán los requisitos necesarios, con el fin de poder cumplir con los objetivos del proyecto. Estos requisitos podrán ser recopilados mediante técnicas y herramientas como ser: entrevistas, cuestionarios, observaciones, etc.

Como se mencionó anteriormente, la domótica no posee una metodología definida, pero se proponen una serie de pasos para llegar a finalizar el proyecto de la mejor forma y cumplir con los objetivos planteados.

Estructura de Desglose del Trabajo (EDT)

Este proceso consiste en dividir las tareas del proyecto en partes o componentes más pequeñas y fáciles de manejar.

La EDT nos sirve para organizar y definir el alcance total del proyecto. En este caso, en la imagen 35 se puede observar la estructura de desglose del trabajo del proyecto.

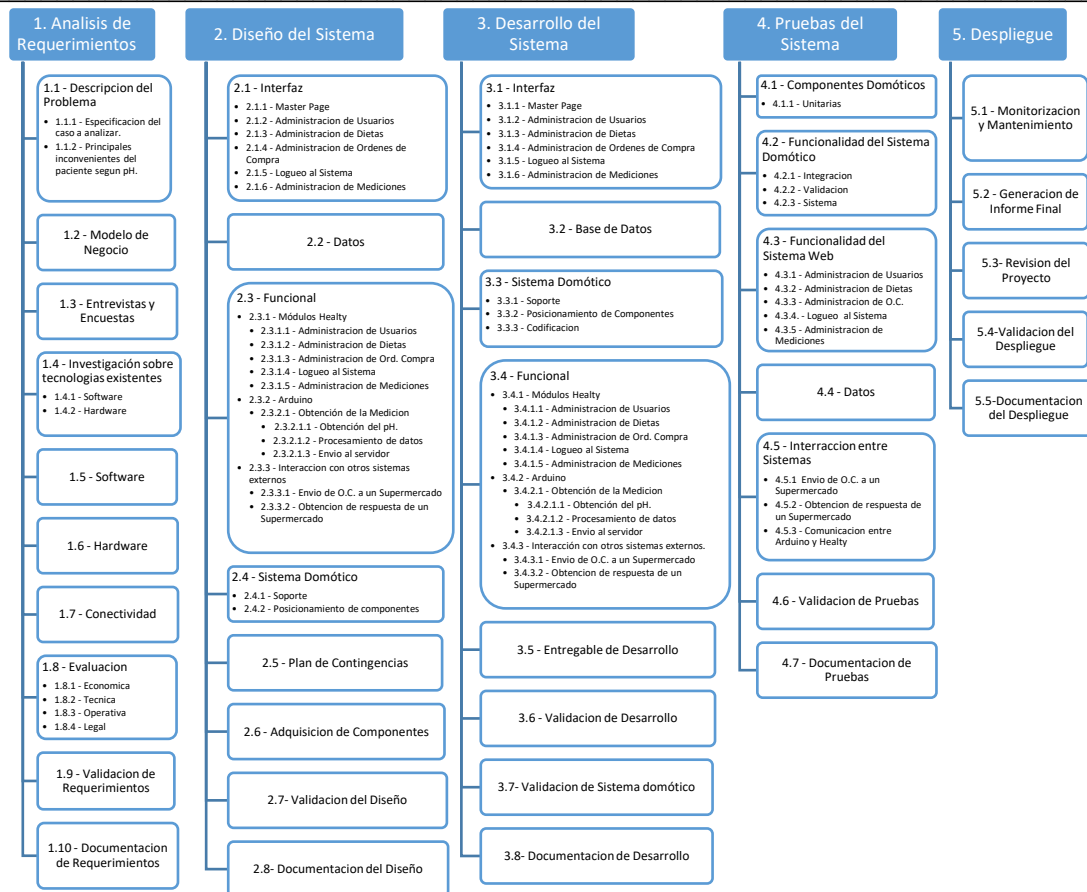


Imagen 35: Estructura de Desglose del Trabajo (EDT)

Diccionario de la EDT

La función que tiene el diccionario, es la de respaldar a la EDT y proporcionar una descripción más detallada de los componentes o tareas que serán realizadas.

Realizaremos una breve explicación de cada una de las tareas que se ejecutarán:

1. Análisis del requerimiento:

- Descripción del Problema:
 - Especificación del caso a analizar: se tendrán en cuenta los diferentes casos que podrían resultar al realizar un análisis de pH en la persona para determinar su estado de salud.
 - Descripción de los principales inconvenientes del paciente según el pH obtenido, junto con su grado de importancia e impacto en la vida diaria del individuo.
- Modelo de negocio: se debe tener un entendimiento del sistema que se está realizando y el fin por el cual se desarrolla.
- Entrevistas y encuestas: en base a diferentes entrevistas y encuestas que se realizarán para adquirir y obtener información, se podrán determinar distintos valores que serán determinantes para la realización del sistema.
- Investigación sobre tecnologías existentes: se hará una búsqueda sobre tecnologías que se podrán utilizar para la realización del proyecto final.



- Software: investigación de las tecnologías disponibles en software para la resolución del problema.
 - Hardware: investigación de las tecnologías disponibles en hardware para la resolución del problema.
 - Software: Se realiza un análisis más profundo sobre la tecnología de software seleccionada para la realización tanto del sistema domótico como del sistema web.
 - Hardware: Se realiza un análisis más profundo sobre la tecnología de hardware seleccionada, que será necesaria para la construcción del sistema domótico.
 - Conectividad: Se analiza la forma de comunicación y conectividad que se tendrán entre los diferentes sistemas, y cuál es la mejor opción para implementarla.
 - Evaluación:
 - Evaluación económica: una evaluación del costo de desarrollo frente al beneficio final producido por el sistema desarrollado.
 - Evaluación técnica: un estudio de la funcionalidad, el rendimiento, las restricciones que pueden afectar a la posibilidad de realización de un sistema aceptable.
 - Evaluación operativa: un estudio para determinar si el sistema podrá emplearse cuando esté desarrollado.
 - Evaluación legal: determinación de cualquier infracción, violación o ilegalidad que pudiese resultar del desarrollo del sistema.
 - Validación de requerimientos: este proceso consiste en comprobar que los requisitos fueron especificados de acuerdo a las necesidades requeridas, y que cumplan con el fin del proyecto. Esto evita el riesgo de realizar una mala implementación.
 - Documentación de requerimientos: esta documentación es parte del proyecto final.
2. Diseño del Sistema:
- Interfaz: Se realiza el diseño de la interfaz del sistema web Healty, la cual debe cumplir ciertos estándares de diseño y ser lo más amigable para el usuario final.
 - Master page: En la misma se definen la cabecera y el pie del sitio web, para que todas las otras páginas del sistema Healty hereden de la misma.
 - Administración de usuarios: se define el estándar de diseño para la administración de usuarios del sistema. Esto hace referencia al Alta, Modificación y Eliminación de usuarios.
 - Administración de dietas: se define el estándar de diseño para la administración de Dietas del sistema. Esto hace referencia al Alta, Modificación y Eliminación de dietas.
 - Administración de órdenes de compra: se define el estándar de diseño para la administración de Órdenes de Compra. Esto hace referencia al Alta, Modificación, Eliminación, Visualización y Pago de Órdenes de Compra.
 - Logueo al sistema: se define el estándar de diseño para el Login/Logout del sistema.



- Administración de mediciones: se define el estándar de diseño para la administración de Mediciones. Esto hace referencia a la visualización y obtención de dietas según el pH de la medición.
- Datos: Se realiza el diseño de la estructura de datos utilizada en el sistema web Healty.
- Funcional
 - Módulos de Healty: De acuerdo a los diferentes requerimientos que se tiene, se plantean diferentes diseños para el desarrollo de los módulos del sistema web Healty.
 - Administración de usuarios: comportamiento del módulo asociado al Alta, modificación y eliminación de usuarios.
 - Administración de dietas: comportamiento del módulo asociado al Alta, modificación y eliminación de dietas.
 - Administración de órdenes de compra: comportamiento del módulo asociado al Alta, modificación, eliminación, visualización y pago de órdenes de compra.
 - Logueo al sistema: comportamiento del módulo asociado al Login/Logout del sistema.
 - Administración de dietas: comportamiento del módulo asociado a la visualización y obtención de dietas según pH de la medición.
 - Arduino: Se empieza a diseñar el proceso de funcionamiento del sistema domótico.
 - Obtención de medición: comportamiento del sistema domótico a la hora de enviar el alta de la medición al sistema Healty.
 - Obtención del pH: el sistema obtendrá a través de la muestra de orina un valor asociado a la tira reactiva.
 - Procesamiento de los datos: cada valor obtenido en la tira reactiva está vinculado a un pH, en este paso se realizará la asociación entre los mismos.
 - Envío al servidor: una vez obtenido el pH, se realizará el envío de datos al servidor.
 - Interacción con otros sistemas externos: se empieza a diseñar el proceso de comunicación con sistemas externos.
 - Envío de orden de compra a un supermercado: comportamiento del módulo asociado a como se realizará el envío de una orden de compra generada al supermercado.
 - Obtención de respuesta de un supermercado: comportamiento del módulo asociado a como se realizará la obtención de una respuesta por parte del supermercado.
- Sistema Domótico: Se diseña toda la parte del hardware del sistema domótico.
 - Soporte: se realiza el diseño del armazón en donde estarán ubicados todos los componentes adicionales a Arduino.
 - Posicionamiento de componentes: se establece la forma y ubicación de los diferentes componentes asociados.
- Plan de contingencias: Se diseña un plan de contingencias que ayudará a disminuir el riesgo por algún problema que ocurriese en el proyecto.
- Adquisición de componentes: En base al análisis de requerimientos, ya se tiene una decisión de los componentes o materiales que se necesitan para la



realización del sistema domótico, por lo cual se empieza a gestionar la búsqueda y compra de los mismos a diferentes proveedores.

- Validación del diseño: Comprobar que se cumple con las especificaciones requeridas para el objetivo del proyecto.
- Documentación del diseño: esta documentación es parte del proyecto final.

3. Desarrollo del Sistema:

- Interfaz: Se desarrolla toda la interfaz del sistema web Healty en base al diseño que se realizó.
 - Master page: se realiza la codificación del Master Page.
 - Administración de usuarios: se realiza la codificación de las vistas para la administración de usuarios (Alta, Modificación y Eliminación).
 - Administración de dietas: se realiza la codificación de las vistas para la administración de dietas (Alta, Modificación y Eliminación).
 - Administración de órdenes de compra: se realiza la codificación de las vistas para la administración de Órdenes de Compra (Alta, Modificación, Eliminación, Visualización y Pago).
 - Logueo al sistema: se realiza la codificación de las vistas para el Login/Logout del sistema.
 - Administración de mediciones: se realiza la codificación de las vistas para la administración de mediciones (Visualización y Obtención de dietas según el pH asociado a la medición).
- Base de datos: En base al DBMS seleccionado, se desarrolla la estructura de bases de datos que va a utilizar el sistema web Healty.
- Sistema domótico: En base al diseño que se realizó, se empieza el armado del sistema domótico.
 - Soporte: se realiza el armado del gabinete o armazón para mantener ubicados firmemente los diferentes componentes del sistema.
 - Posicionamiento de componentes: se ubican los diferentes componentes en el armazón.
 - Codificación: se realiza el desarrollo del sistema domótico para que su comportamiento sea acorde a lo definido en el diseño.
- Funcional
 - Módulos de Healty: Se realiza el desarrollo de los diferentes módulos en base al diseño establecido.
 - Administración de usuarios: se realiza la codificación de la lógica para la administración de usuarios (Alta, modificación y eliminación).
 - Administración de dietas: se realiza la codificación de la lógica para la administración de dietas (Alta, modificación y eliminación).
 - Administración de órdenes de compra: se realiza la codificación de la lógica para la administración de órdenes de compra (Alta, modificación, eliminación, visualización y pago).
 - Logueo al sistema: se realiza la codificación de la lógica para el Login/Logout del sistema.
 - Administración de mediciones: se realiza la codificación de la lógica para la administración de mediciones (visualización y obtención de dietas según pH).
 - Arduino: Se realiza la programación del código que hará que el sistema domótico funcione de acuerdo a los requerimientos necesarios.



- Obtención de medición: se realiza la codificación de la lógica para el envío de la medición al sistema Healty.
 - Obtención del pH
 - Procesamiento de los datos
 - Envío al servidor
 - Interacción con otros sistemas externos: Se desarrolla la forma en que se implementan servicios para la comunicación con sistemas externos.
 - Envío de orden de compra a un supermercado: se realiza la codificación de la lógica asociada a como se realizará el envío de una orden de compra generada al supermercado.
 - Obtención de respuesta de un supermercado: se realiza la codificación de la lógica asociada a como se realizará la obtención de una respuesta por parte del supermercado.
 - Entregable de desarrollo: Se obtiene un producto funcional que puede ser presentado al cliente. El mismo es un incremento al producto potencialmente entregable.
 - Validación de desarrollo: Comprobar que se cumplen todos los requerimientos necesarios para cumplir con los objetivos del proyecto.
 - Validación del sistema domótico: Comprobar que se cumplen todos los requerimientos necesarios para cumplir con los objetivos del proyecto.
 - Documentación de desarrollo: esta documentación es parte del proyecto.
4. Pruebas del Sistema:
- Componentes domóticos: se realizarán pruebas de los diferentes componentes que se utilizarán para el armado del sistema domótico.
 - Unitarias: se ejecutan los componentes domóticos de manera individual, sin ninguna conexión a otro componente para poder observar su comportamiento.
 - Funcionalidad del sistema domótico: se realizarán diferentes tipos de pruebas para comprobar la fidelidad y el buen funcionamiento del sistema domótico.
 - Integración: consiste en la prueba de cada uno de los componentes en interacción con otros.
 - Validación: se evalúa si el sistema domótico funciona de acuerdo a las expectativas razonables en base a los requerimientos.
 - Sistema: este sistema domótico es incorporado a los otros elementos que forman parte de todo el sistema para poder observar su comportamiento.
 - Funcionalidad del sistema Web: se realizarán pruebas para comprobar el correcto funcionamiento del sistema web Healty.
 - Administración de usuarios: se realizan diferentes pruebas para observar el comportamiento del módulo de administración de usuarios.
 - Administración de dietas: se realizan diferentes pruebas para observar el comportamiento del módulo de administración de dietas.
 - Administración de órdenes de compra: se realizan diferentes pruebas para observar el comportamiento del módulo de administración de órdenes de compra.
 - Logueo al sistema: se realizan diferentes pruebas para observar el comportamiento del Login/Logout al sistema.
 - Administración de mediciones: se realizan diferentes pruebas para observar el comportamiento del módulo de administración de mediciones.



- Datos: Se realizarán pruebas para comprobar la integridad de la información que se tiene en el sistema web Healty.
 - Interacción entre sistemas: Se realizarán pruebas de conexión y envío de información, entre los diferentes sistemas involucrados en el proyecto final.
 - Envío de orden de compra a un supermercado: se realizan diferentes pruebas, para observar si el envío de una orden de compra al supermercado es correcto.
 - Obtención de respuesta de un supermercado: se realizan diferentes pruebas, para observar si se obtiene de una forma correcta la respuesta desde el supermercado.
 - Comunicación entre Arduino y Healty: se realizan diferentes pruebas para observar, si el valor de pH de una medición, se envía de manera correcta al sistema web Healty.
 - Validación de pruebas: Comprobar que se cumple con las especificaciones requeridas para el objetivo del proyecto
 - Documentación de pruebas: esta documentación es parte del proyecto.
5. Despliegue:
- Monitorización y mantenimiento del sistema domótico y web: se tendrá planificado cómo se hará el mantenimiento del sistema en general.
 - Generación del informe final: se realizará un informe final, en donde se podrán observar los resultados obtenidos.
 - Revisión del proyecto: se hace una revisión final del proyecto, para ver si cumple con las especificaciones descriptas anteriormente, en base a los resultados obtenidos a la hora de la ejecución del mismo.
 - Validación del despliegue: Comprobar que se cumple con las especificaciones requeridas para el objetivo del proyecto
 - Documentación de despliegue: esta documentación es parte del proyecto.

Recursos Necesarios

Inventario de Recursos:

Entre los recursos que se encuentran disponibles para la realización del proyecto final se tiene:

- Hardware:
 - Arduino UNO
 - Sensor de Ultrasonidos HC-SR04
 - Sensor de Colores TCS3200
 - Servo
 - ESP-8266
 - Fuente de Alimentación
 - Notebooks:
 - ACER:
 - Microprocesador: Intel Core I5.
 - Memoria RAM: 4Gb.
 - SSD: 240Gb.
 - BANGHO:
 - Microprocesador: Intel Core I7.
 - Memoria RAM: 8Gb.
 - HDD: 500Gb.



- Software:
 - Base para notebooks:
 - ACER:
 - Sistema Operativo: Windows 8.
 - Entorno de Virtualización: VMWare 10
 - Sistema Operativo: Ubuntu 14
 - IDE: Sublime Text 3
 - BD: PostgreSQL, PgAdmin
 - Lenguaje de Programación: Ruby 1.9.2
 - Framework de desarrollo: Rails 4.2
 - BANGHO:
 - Sistema Operativo: Windows 10 Pro
 - Entorno de Virtualización: VMWare 10
 - Sistema Operativo: Ubuntu 14
 - IDE: Sublime Text 3
 - BD: PostgreSQL, PgAdmin
 - Lenguaje de Programación: Ruby 1.9.2
 - Framework de desarrollo: Rails 4.2
 - Base para servidor: Servidor Heroku
 - PaaS - Platform as a Service.
 - Arquitectura Dyno: Free, 512MB, CPU 1x, No dedicado.
 - Linux Ubuntu. Soporte para Ruby.
 - Base para Arduino: Lenguaje propio basado en Processing.
 - Recursos Humanos:
 - Facundo Adrián Díaz Martínez
 - Miguel Carlos Natino
 - Otros:
 - Protoboard: utilizada para realizar todas las conexiones entre los diferentes submódulos de Arduino.
 - Transformador: utilizado para conectar la fuente de alimentación.
 - Soporte para Arduino (Gabinete): realizado a medida para poder proteger y soportar los diferentes submódulos de Arduino.
 - Pines Macho y Hembra: realizar conexiones en el circuito propuesto.
 - Cables de Alimentación: utilizados como alimentación para el Arduino UNO.
 - Tiras reactivas: utilizadas para medir el pH del usuario.

Recursos Humanos del Proyecto

La gestión de recursos humanos, incluye los procesos para organizar y gestionar al equipo del proyecto. El equipo del proyecto, está formado por aquellas personas a las cuales se les ha asignado roles y responsabilidades para cumplir con los objetivos.

Cabe aclarar que en este proyecto, solo se cuenta con dos personas que realizan todo el trabajo, así que deberán cumplir diferentes roles y tareas que se detallan a continuación:

- Analista: es aquella persona que tiene como fin analizar el problema y describirlo, con el propósito de ser solucionado mediante un sistema de información. Debe tener un conocimiento del paradigma que se está utilizando, el modelado funcional, modelo de datos y un conocimiento sobre la tecnología a emplear.



- **Diseñador:** tiene como fin, el definir una arquitectura de hardware y software, componentes, módulos y datos de un sistema, a los efectos de satisfacer los requerimientos ya definidos en el análisis.
- **Desarrollador:** es aquella persona que escribe, depura y mantiene el código fuente de un programa informático, es decir, el conjunto de instrucciones que ejecuta el hardware de una computadora para realizar una tarea determinada. En el caso de dicho proyecto, también le llamamos desarrollador a la persona encargada de armar el prototipo funcional del sistema domótico.

En la tabla 4 podemos observar las cantidades de recursos que se poseen y en la tabla 5, la matriz de responsabilidades asignada a cada uno de los roles que participan en el proyecto.

Rol	Cantidad
Analista	1
Diseñador	1
Desarrollador	2

Tabla 4: Cantidades de RRHH

Rol / Etapa	Análisis	Diseño	Desarrollo	Pruebas	Despliegue
Analista	R	P	-	P	R
Diseñador	-	R	V	R	P
Desarrollador	-	O	R	R	P

Tabla 5: Matriz de Responsabilidades

En la matriz de responsabilidades se pueden observar dos valores diferentes:

P: Participa

O: Opina

R: Responsable

V: Revisa

Las diferentes etapas están relacionadas al alcance del proyecto, basadas en los diferentes puntos que se establecieron.



Tiempo del Proyecto

La gestión del tiempo del proyecto, incluye todos los procesos requeridos para administrar la finalización del proyecto en tiempo y forma.

En la tabla 7, podemos observar las tareas que se van a realizar y la duración de cada una de ellas. En la imagen 36, se observa el diagrama de Gantt asociado al proyecto.

Los tiempos y fechas que se colocaron, ya son vinculados a cuando finalice el cursado de la materia “Proyecto de Grado”.

Tareas

En la tabla 6 se puede observar la duración total del proyecto final, que es de 121 días hábiles. A su vez, se observan las tareas generales junto a sus duraciones, los inicios y final de cada una de ellas.

Dentro de las tareas generales HEALTY y SISTEMA DOMÓTICO, se tienen las etapas de diseño, desarrollo y pruebas.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Proyecto Final	121 días	mié 21/10/15	mié 06/04/16	
2	1 - Analisis de Requerimientos	28 días	mié 21/10/15	vie 27/11/15	
21	HEALTY	77 días	lun 30/11/15	mar 15/03/16	20
132	SISTEMA DOMOTICO	40 días	lun 30/11/15	vie 22/01/16	20
169	5 - Despliegue	16 días	mié 16/03/16	mié 06/04/16	21,132

Tabla 6: Duración Total del Proyecto

La duración de las diferentes tareas generales son las siguientes:

- Análisis del requerimiento: 28 días hábiles
- Diseño, desarrollo y pruebas de Healty: 77 días hábiles
- Diseño, desarrollo y pruebas del sistema domótico: 40 días hábiles
- Despliegue: 16 días hábiles



A continuación en la tabla 7 se desagregan las tareas generales en tareas más pequeñas, para un mayor detalle y entendimiento de las mismas. Como información adicional de dicha tabla, además de la duración junto a sus comienzos y finales, se tiene la columna predecesoras que se utilizan para secuenciar las actividades o tareas que se poseen.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Proyecto Final	121 días	mié 21/10/15	mié 06/04/16	
2	1 - Analisis de Requerimientos	28 días	mié 21/10/15	vie 27/11/15	
3	1.1 - Descripcion del Problema	5 días	mié 21/10/15	mar 27/10/15	
4	1.1.1 - Especificacion del caso a analizar	2 días	mié 21/10/15	jue 22/10/15	
5	1.1.2 - Principales inconvenientes del paciente según pH	3 días	vie 23/10/15	mar 27/10/15	4
6	1.2 - Modelo de Negocio	5 días	mié 28/10/15	mar 03/11/15	3
7	1.3 - Entrevistas y Encuestas	3 días	mié 04/11/15	vie 06/11/15	6
8	1.4 - Investigacion sobre tecnologías existentes	2 días	lun 09/11/15	mar 10/11/15	
9	1.4.1 - Software	2 días	lun 09/11/15	mar 10/11/15	7
10	1.4.2 - Hardware	2 días	lun 09/11/15	mar 10/11/15	7
11	1.5 - Software	3 días	mié 11/11/15	vie 13/11/15	8
12	1.6 - Hardware	3 días	mié 11/11/15	vie 13/11/15	8
13	1.7 - Conectividad	3 días	lun 16/11/15	mié 18/11/15	11,12
14	1.8 - Evaluacion	2 días	jue 19/11/15	vie 20/11/15	13
15	1.8.1 - Economica	1 día	jue 19/11/15	jue 19/11/15	
16	1.8.2 - Tecnica	1 día	jue 19/11/15	jue 19/11/15	
17	1.8.3 - Operativa	1 día	vie 20/11/15	vie 20/11/15	15,16
18	1.8.4 - Legal	1 día	vie 20/11/15	vie 20/11/15	15,16
19	1.9 - Validacion de requerimientos	5 días	lun 23/11/15	vie 27/11/15	17,18
20	1.10 - Documentacion de requerimientos	0 días	vie 27/11/15	vie 27/11/15	19
21	HEALTY	77 días	lun 30/11/15	mar 15/03/16	20
22	SPRINT 1	16 días	lun 30/11/15	lun 21/12/15	
23	2 - Diseño del Sistema	6 días	lun 30/11/15	lun 07/12/15	
24	2.1 - Interfaz	2 días	lun 30/11/15	mar 01/12/15	
25	2.1.1 - Master Page	1 día	lun 30/11/15	lun 30/11/15	20
26	2.1.2 - Administracion de Usuarios	1 día	mar 01/12/15	mar 01/12/15	25



UCS – Fac. De Ingeniería
Domótica aplicada a la prevención de enfermedades
Capítulo 4: Solución Propuesta

Diagrama de Gantt	27	2.1.5 - Logueo al Sistema	1 día	mar 01/12/15	mar 01/12/15	25
	28	2.2 - Datos	1 día	mié 02/12/15	mié 02/12/15	26,27
	29	2.3 - Funcional	1 día	jue 03/12/15	jue 03/12/15	
	30	2.3.1 - Modulos Healty	1 día	jue 03/12/15	jue 03/12/15	
	31	2.3.1.1 - Administracion de Usuarios	1 día	jue 03/12/15	jue 03/12/15	28
	32	2.3.1.4 - Logueo al Sistema	1 día	jue 03/12/15	jue 03/12/15	28
	33	2.7 - Validacion del Diseño	2 días	vie 04/12/15	lun 07/12/15	24,28,29
	34	2.8 - Documentacion del Diseño	0 días	lun 07/12/15	lun 07/12/15	33
	35	3 - Desarrollo del Sistema	8 días	mar 08/12/15	jue 17/12/15	34
	36	3.1 - Interfaz	4 días	mar 08/12/15	vie 11/12/15	
	37	3.1.1 - Master Page	2 días	mar 08/12/15	mié 09/12/15	34
	38	3.1.2 - Administracion de Usuarios	2 días	jue 10/12/15	vie 11/12/15	37
	39	3.1.5 - Logueo al Sistema	1 día	jue 10/12/15	jue 10/12/15	37
	40	3.2 - Base de Datos	1 día	mar 08/12/15	mar 08/12/15	34
	41	3.4 - Funcional	3 días	lun 14/12/15	mié 16/12/15	
	42	3.4.1 - Modulos Healty	3 días	lun 14/12/15	mié 16/12/15	
	43	3.4.1.1 - Administracion de Usuarios	3 días	lun 14/12/15	mié 16/12/15	36,40
	44	3.4.1.4 - Logueo al Sistema	2 días	lun 14/12/15	mar 15/12/15	36,40
	45	3.5 - Entregable de Desarrollo	0 días	mié 16/12/15	mié 16/12/15	43,44
	46	3.6 - Validacion de Desarrollo	1 día	jue 17/12/15	jue 17/12/15	45
	47	3.8 - Documentacion de Desarrollo	0 días	jue 17/12/15	jue 17/12/15	46
	48	4 - Pruebas del Sistema	2 días	vie 18/12/15	lun 21/12/15	47
	49	4.3 - Funcionalidad del Sistema Web	1 día	vie 18/12/15	vie 18/12/15	
	50	4.3.1 - Administracion de Usuarios	1 día	vie 18/12/15	vie 18/12/15	47
	51	4.3.4 - Logueo al Sistema	1 día	vie 18/12/15	vie 18/12/15	47
Diagrama de Gantt	52	4.4 - Datos	1 día	vie 18/12/15	vie 18/12/15	47
	53	4.6 - Validacion de Pruebas	1 día	lun 21/12/15	lun 21/12/15	50,51,52
	54	4.7 - Documentacion de Pruebas	0 días	lun 21/12/15	lun 21/12/15	53
	55	SPRINT 2	16 días	mar 22/12/15	mar 12/01/16	54
	56	2 - Diseño del Sistema	4 días	mar 22/12/15	vie 25/12/15	
	57	2.1 - Interfaz	1 día	mar 22/12/15	mar 22/12/15	
	58	2.1.3 - Administracion de Dietas	1 día	mar 22/12/15	mar 22/12/15	54
	59	2.1.6 - Administracion de Mediciones	1 día	mar 22/12/15	mar 22/12/15	54
	60	2.3 - Funcional	2 días	mié 23/12/15	jue 24/12/15	
	61	2.3.1 - Modulos Healty	2 días	mié 23/12/15	jue 24/12/15	
	62	2.3.1.2 - Administracion de Dietas	2 días	mié 23/12/15	jue 24/12/15	58,59
	63	2.3.1.5 - Administracion de Mediciones	2 días	mié 23/12/15	jue 24/12/15	58,59
	64	2.7 - Validacion del Diseño	1 día	vie 25/12/15	vie 25/12/15	57,60
	65	2.8 - Documentacion del Diseño	0 días	vie 25/12/15	vie 25/12/15	64
	66	3 - Desarrollo del Sistema	9 días	lun 28/12/15	jue 07/01/16	65
	67	3.1 - Interfaz	2 días	lun 28/12/15	mar 29/12/15	
	68	3.1.3 - Administracion de Dietas	2 días	lun 28/12/15	mar 29/12/15	65
	69	3.1.3 - Administracion de Mediciones	2 días	lun 28/12/15	mar 29/12/15	65
	70	3.4 - Funcional	5 días	mié 30/12/15	mar 05/01/16	
	71	3.4.1 - Modulos Healty	5 días	mié 30/12/15	mar 05/01/16	
	72	3.4.1.2 - Administracion de Dietas	5 días	mié 30/12/15	mar 05/01/16	68,69
	73	3.4.1.5 - Administracion de Mediciones	3 días	mié 30/12/15	vie 01/01/16	68,69
	74	3.5 - Entregable de Desarrollo	0 días	mar 05/01/16	mar 05/01/16	72,73
	75	3.6 - Validacion de Desarrollo	2 días	mié 06/01/16	jue 07/01/16	74
	76	3.8 - Documentacion de Desarrollo	0 días	jue 07/01/16	jue 07/01/16	75
	77	4 - Pruebas del Sistema	3 días	vie 08/01/16	mar 12/01/16	76
	78	4.3 - Funcionalidad del Sistema Web	2 días	vie 08/01/16	lun 11/01/16	



Diagrama de Gantt	79	4.3.2 - Administracion de Dietas	2 días	vie 08/01/16	lun 11/01/16	76
	80	4.3.5 - Administracion de Mediciones	1 día	vie 08/01/16	vie 08/01/16	76
	81	4.4 - Datos	1 día	vie 08/01/16	vie 08/01/16	76
	82	4.6 - Validacion de Pruebas	1 día	mar 12/01/16	mar 12/01/16	79,80,81
	83	4.7 - Documentacion de Pruebas	0 días	mar 12/01/16	mar 12/01/16	82
	84	SPRINT 3	28 días	mié 13/01/16	vie 19/02/16	83
	85	2 - Diseño del Sistema	7 días	mié 13/01/16	jue 21/01/16	
	86	2.1 - Interfaz	4 días	mié 13/01/16	lun 18/01/16	
	87	2.1.4 - Administracion de Ordenes de Compra	4 días	mié 13/01/16	lun 18/01/16	83
	88	2.3 - Funcional	5 días	mié 13/01/16	mar 19/01/16	
	89	2.3.1 - Modulos Healty	5 días	mié 13/01/16	mar 19/01/16	
	90	2.3.1.3 - Administracion de Ord. Compra	5 días	mié 13/01/16	mar 19/01/16	83
	91	2.7 - Validacion del Diseño	2 días	mié 20/01/16	jue 21/01/16	86,88
	92	2.8 - Documentacion del Diseño	0 días	jue 21/01/16	jue 21/01/16	91
	93	3 - Desarrollo del Sistema	16 días	vie 22/01/16	vie 12/02/16	92
	94	3.1 - Interfaz	5 días	vie 22/01/16	jue 28/01/16	
	95	3.1.4 - Administracion de Ordenes de Compra	5 días	vie 22/01/16	jue 28/01/16	92
	96	3.4 - Funcional	8 días	vie 29/01/16	mar 09/02/16	
	97	3.4.1 - Modulos Healty	8 días	vie 29/01/16	mar 09/02/16	
	98	3.4.1.3 - Administracion de Ord. Compra	8 días	vie 29/01/16	mar 09/02/16	95
	99	3.5 - Entregable de Desarrollo	0 días	mar 09/02/16	mar 09/02/16	98
	100	3.6 - Validacion de Desarrollo	3 días	mié 10/02/16	vie 12/02/16	99
Diagrama de Gantt	101	3.8 - Documentacion de Desarrollo	0 días	vie 12/02/16	vie 12/02/16	100
	102	4 - Pruebas del Sistema	5 días	lun 15/02/16	vie 19/02/16	101
	103	4.3 - Funcionalidad del Sistema Web	4 días	lun 15/02/16	jue 18/02/16	
	104	4.3.3 - Administracion de O.C.	4 días	lun 15/02/16	jue 18/02/16	101
	105	4.4 - Datos	2 días	lun 15/02/16	mar 16/02/16	101
	106	4.6 - Validacion de Pruebas	1 día	vie 19/02/16	vie 19/02/16	104,105
	107	4.7 - Documentacion de Pruebas	0 días	vie 19/02/16	vie 19/02/16	106
	108	SPRINT 4	17 días	lun 22/02/16	mar 15/03/16	107
	109	2 - Diseño del Sistema	4 días	lun 22/02/16	jue 25/02/16	
	110	2.3 - Funcional	2 días	lun 22/02/16	mar 23/02/16	
	111	2.3.3 - Interaccion con otros sistemas externos	2 días	lun 22/02/16	mar 23/02/16	
	112	2.3.3.1 - Envio de O.C a un Supermercado	2 días	lun 22/02/16	mar 23/02/16	107
	113	2.3.3.2 - Obtencion de respuesta de un Supermercado	2 días	lun 22/02/16	mar 23/02/16	107
	114	2.7 - Validacion del Diseño	2 días	mié 24/02/16	jue 25/02/16	110
	115	2.8 - Documentacion del Diseño	0 días	jue 25/02/16	jue 25/02/16	114
	116	3 - Desarrollo del Sistema	8 días	vie 26/02/16	mar 08/03/16	115
	117	3.4 - Funcional	7 días	vie 26/02/16	lun 07/03/16	
	118	3.4.3 - Interaccion con otros sistemas externos	7 días	vie 26/02/16	lun 07/03/16	
	119	3.4.3.1 - Envio de O.C. a un Supermercado	7 días	vie 26/02/16	lun 07/03/16	115
	120	3.4.3.2 - Obtencion de respuesta de un Supermercado	4 días	vie 26/02/16	mié 02/03/16	115
	121	3.5 - Entregable de Desarrollo	0 días	lun 07/03/16	lun 07/03/16	119,120



Diagrama de Gantt	122	3.6 - Validacion de Desarrollo	1 día	mar 08/03/16	mar 08/03/16	121
	123	3.8 - Documentacion de Desarrollo	0 días	mar 08/03/16	mar 08/03/16	122
	124	▢ 4 - Pruebas del Sistema	5 días	mié 09/03/16	mar 15/03/16	123
	125	4.4 - Datos	2 días	mié 09/03/16	jue 10/03/16	123
	126	▢ 4.5 - Interaccion entre Sistemas	3 días	mié 09/03/16	vie 11/03/16	
	127	4.5.1 - Envio de O.C a un Supermercado	1 día	mié 09/03/16	mié 09/03/16	123
	128	4.5.2 - Obtencion de respuesta de un Supermercado	1 día	mié 09/03/16	mié 09/03/16	123
	129	4.5.3 - Comunicación entre Arduino y Healty	3 días	mié 09/03/16	vie 11/03/16	123,168
	130	4.6 - Validacion de Pruebas	2 días	lun 14/03/16	mar 15/03/16	125,126
	131	4.7 - Documentacion de Pruebas	0 días	mar 15/03/16	mar 15/03/16	130
	132	▢ SISTEMA DOMOTICO	40 días	lun 30/11/15	vie 22/01/16	20
	133	▢ 2 - Diseño del Sistema	10 días	lun 30/11/15	vie 11/12/15	
	134	▢ 2.3 - Funcional	3 días	lun 30/11/15	mié 02/12/15	
	135	▢ 2.3.2 - Arduino	3 días	lun 30/11/15	mié 02/12/15	
	136	▢ 2.3.2.1 - Obtencion de Medicion	3 días	lun 30/11/15	mié 02/12/15	20
	137	2.3.2.1.1 - Obtención del pH	1 día	lun 30/11/15	lun 30/11/15	
	138	2.3.2.1.2 - Procesamiento de Datos	1 día	mar 01/12/15	mar 01/12/15	137
	139	2.3.2.1.3 - Envio al Servidor	1 día	mié 02/12/15	mié 02/12/15	138
	140	▢ 2.4 - Sistema Domotico	4 días	jue 03/12/15	mar 08/12/15	
	141	2.4.1 - Soporte	2 días	jue 03/12/15	vie 04/12/15	136
	142	2.4.2 - Posicionamiento de Componentes	2 días	lun 07/12/15	mar 08/12/15	141
	143	2.5 - Plan de Contingencias	3 días	lun 30/11/15	mié 02/12/15	20
	144	2.6 - Adquisicion de Componentes	0 días	mié 02/12/15	mié 02/12/15	143
	145	2.7 - Validacion de Diseño	3 días	mié 09/12/15	vie 11/12/15	134,140
Diagrama de Gantt	146	2.8 - Documentacion del Diseño	0 días	vie 11/12/15	vie 11/12/15	145
	147	▢ 3 - Desarrollo del Sistema	13 días	lun 14/12/15	mié 30/12/15	146
	148	▢ 3.3 - Sistema Domotico	5 días	lun 14/12/15	vie 18/12/15	
	149	3.3.1 - Soporte	3 días	lun 14/12/15	mié 16/12/15	146
	150	3.3.2 - Posicionamiento de Componentes	2 días	jue 17/12/15	vie 18/12/15	149
	151	3.3.3 - Codificacion	5 días	lun 14/12/15	vie 18/12/15	146
	152	▢ 3.4 - Funcional	5 días	lun 21/12/15	vie 25/12/15	
	153	▢ 3.4.2 - Arduino	5 días	lun 21/12/15	vie 25/12/15	
	154	▢ 3.4.2.1 - Obtencion de Medicion	5 días	lun 21/12/15	vie 25/12/15	148
	155	3.4.2.1.1 - Obtencion del pH	2 días	lun 21/12/15	mar 22/12/15	
	156	3.4.2.1.2 - Procesamiento de Datos	2 días	mié 23/12/15	jue 24/12/15	155
	157	3.4.2.1.3 - Envio al Servidor	1 día	vie 25/12/15	vie 25/12/15	156
	158	3.7 - Validacion de Sistema Domotico	3 días	lun 28/12/15	mié 30/12/15	148,152
	159	3.8 - Documentacion de Desarrollo	0 días	mié 30/12/15	mié 30/12/15	158



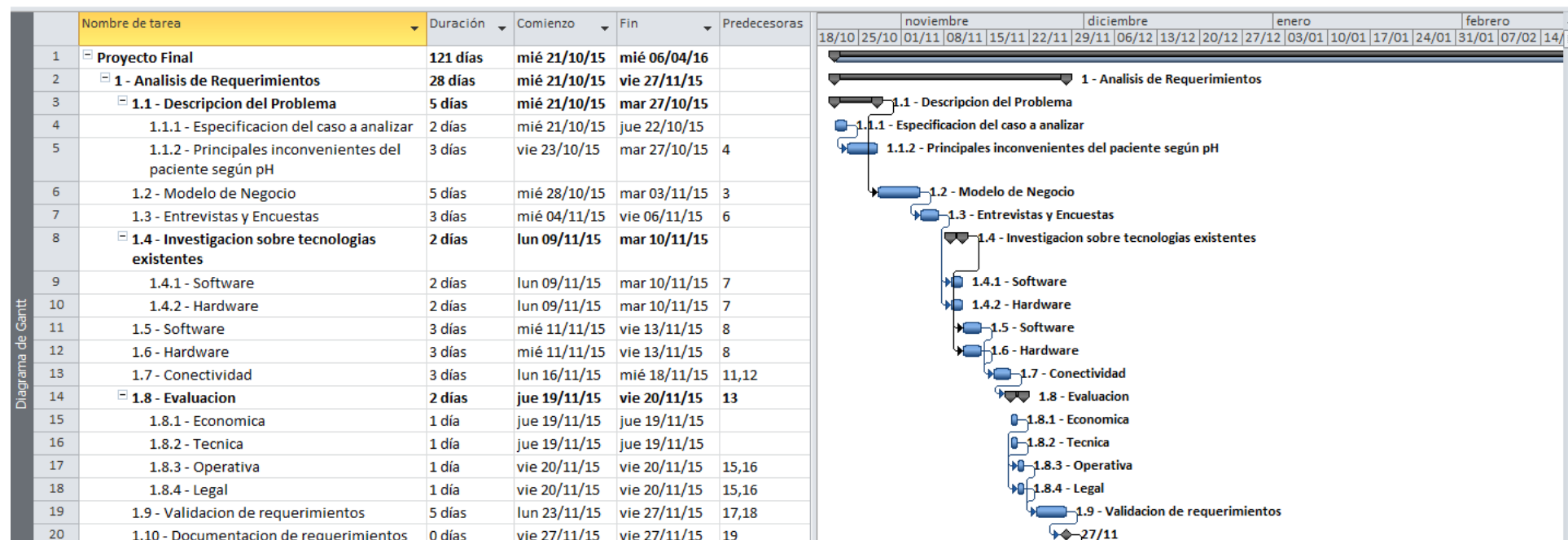
Diagrama de Gantt	160	▢ 4 - Pruebas del Sistema	17 días	jue 31/12/15	vie 22/01/16	159
	161	▢ 4.1 - Componentes Domoticos	3 días	jue 31/12/15	lun 04/01/16	
	162	4.1.1 - Unitarias	3 días	jue 31/12/15	lun 04/01/16	159
	163	▢ 4.2 - Funcionalidad del Sistema Domotico	11 días	mar 05/01/16	mar 19/01/16	
	164	4.2.1 - Integracion	3 días	mar 05/01/16	jue 07/01/16	161
	165	4.2.2 - Validacion	3 días	vie 08/01/16	mar 12/01/16	164
	166	4.2.3 - Sistema	5 días	mié 13/01/16	mar 19/01/16	165
	167	4.6 - Validacion de Pruebas	3 días	mié 20/01/16	vie 22/01/16	161,163
	168	4.7 - Documentacion de Pruebas	0 días	vie 22/01/16	vie 22/01/16	167
	169	▢ 5 - Despliegue	16 días	mié 16/03/16	mié 06/04/16	21,132
	170	5.1 - Monitorizacion y Mantenimiento	5 días	mié 16/03/16	mar 22/03/16	
	171	5.2 - Generacion de Informe Final	3 días	mié 23/03/16	vie 25/03/16	170
	172	5.3 - Revision del Proyecto	3 días	lun 28/03/16	mié 30/03/16	171
	173	5.4 - Validacion del Despliegue	5 días	jue 31/03/16	mié 06/04/16	171,172
	174	5.5 - Documentacion del Despliegue	0 días	mié 06/04/16	mié 06/04/16	173

Tabla 7: Desagregación de Tareas



Diagrama de Gantt

Es una herramienta gráfica útil cuyo objetivo es exponer el tiempo de dedicación previsto para las diferentes tareas o actividades que se poseen a lo largo de un tiempo total determinado.





	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	
						diciembre 29/1106/1213/1220/1227/12 enero 03/0110/0117/0124/01 febrero 31/0107/0214/0221/0228/02 marzo 06/0313/0320/0327/03
1	Proyecto Final	121 días	mié 21/10/15	mié 06/04/16		
2	1 - Analisis de Requerimientos	28 días	mié 21/10/15	vie 27/11/15		1 - Analisis de Requerimientos
21	HEALTHY	77 días	lun 30/11/15	mar 15/03/16	20	
22	SPRINT 1	16 días	lun 30/11/15	lun 21/12/15		SPRINT 1
23	2 - Diseño del Sistema	6 días	lun 30/11/15	lun 07/12/15		2 - Diseño del Sistema
24	2.1 - Interfaz	2 días	lun 30/11/15	mar 01/12/15		2.1 - Interfaz
25	2.1.1 - Master Page	1 día	lun 30/11/15	lun 30/11/15	20	2.1.1 - Master Page
26	2.1.2 - Administracion de Usuarios	1 día	mar 01/12/15	mar 01/12/15	25	2.1.2 - Administracion de Usuarios
27	2.1.5 - Logueo al Sistema	1 día	mar 01/12/15	mar 01/12/15	25	2.1.5 - Logueo al Sistema
28	2.2 - Datos	1 día	mié 02/12/15	mié 02/12/15	26,27	2.2 - Datos
29	2.3 - Funcional	1 día	jue 03/12/15	jue 03/12/15		2.3 - Funcional
30	2.3.1 - Modulos Healty	1 día	jue 03/12/15	jue 03/12/15		2.3.1 - Modulos Healty
31	2.3.1.1 - Administracion de Usuarios	1 día	jue 03/12/15	jue 03/12/15	28	2.3.1.1 - Administracion de Usuarios
32	2.3.1.4 - Logueo al Sistema	1 día	jue 03/12/15	jue 03/12/15	28	2.3.1.4 - Logueo al Sistema
33	2.7 - Validacion del Diseño	2 días	vie 04/12/15	lun 07/12/15	24,28,29	2.7 - Validacion del Diseño
34	2.8 - Documentacion del Diseño	0 días	lun 07/12/15	lun 07/12/15	33	07/12
35	3 - Desarrollo del Sistema	8 días	mar 08/12/15	jue 17/12/15	34	3 - Desarrollo del Sistema
36	3.1 - Interfaz	4 días	mar 08/12/15	vie 11/12/15		3.1 - Interfaz
37	3.1.1 - Master Page	2 días	mar 08/12/15	mié 09/12/15	34	3.1.1 - Master Page
38	3.1.2 - Administracion de Usuarios	2 días	jue 10/12/15	vie 11/12/15	37	3.1.2 - Administracion de Usuarios
39	3.1.5 - Logueo al Sistema	1 día	jue 10/12/15	jue 10/12/15	37	3.1.5 - Logueo al Sistema
40	3.2 - Base de Datos	1 día	mar 08/12/15	mar 08/12/15	34	3.2 - Base de Datos
41	3.4 - Funcional	3 días	lun 14/12/15	mié 16/12/15		3.4 - Funcional
42	3.4.1 - Modulos Healty	3 días	lun 14/12/15	mié 16/12/15		3.4.1 - Modulos Healty
43	3.4.1.1 - Administracion de Usuarios	3 días	lun 14/12/15	mié 16/12/15	36,40	3.4.1.1 - Administracion de Usuarios
44	3.4.1.4 - Logueo al Sistema	2 días	lun 14/12/15	mar 15/12/15	36,40	3.4.1.4 - Logueo al Sistema
45	3.5 - Entregable de Desarrollo	0 días	mié 16/12/15	mié 16/12/15	43,44	16/12
46	3.6 - Validacion de Desarrollo	1 día	jue 17/12/15	jue 17/12/15	45	3.6 - Validacion de Desarrollo
47	3.8 - Documentacion de Desarrollo	0 días	jue 17/12/15	jue 17/12/15	46	17/12
48	4 - Pruebas del Sistema	2 días	vie 18/12/15	lun 21/12/15	47	4 - Pruebas del Sistema
49	4.3 - Funcionalidad del Sistema Web	1 día	vie 18/12/15	vie 18/12/15		4.3 - Funcionalidad del Sistema Web
50	4.3.1 - Administracion de Usuarios	1 día	vie 18/12/15	vie 18/12/15	47	4.3.1 - Administracion de Usuarios
51	4.3.4 - Logueo al Sistema	1 día	vie 18/12/15	vie 18/12/15	47	4.3.4 - Logueo al Sistema
52	4.4 - Datos	1 día	vie 18/12/15	vie 18/12/15	47	4.4 - Datos
53	4.6 - Validacion de Pruebas	1 día	lun 21/12/15	lun 21/12/15	50,51,52	4.6 - Validacion de Pruebas
54	4.7 - Documentacion de Pruebas	0 días	lun 21/12/15	lun 21/12/15	53	21/12



Nombre de tarea					Duración	Comienzo	Fin	Predecesoras
1	Proyecto Final				121 días	mié 21/10/15	mié 06/04/16	
2	1 - Analisis de Requerimientos				28 días	mié 21/10/15	vie 27/11/15	
21	HEALTHY				77 días	lun 30/11/15	mar 15/03/16	20
22	SPRINT 1				16 días	lun 30/11/15	lun 21/12/15	
55	SPRINT 2				16 días	mar 22/12/15	mar 12/01/16	54
56	2 - Diseño del Sistema				4 días	mar 22/12/15	vie 25/12/15	
57	2.1 - Interfaz				1 día	mar 22/12/15	mar 22/12/15	
58	2.1.3 - Administracion de Dietas				1 día	mar 22/12/15	mar 22/12/15	54
59	2.1.6 - Administracion de Mediciones				1 día	mar 22/12/15	mar 22/12/15	54
60	2.3 - Funcional				2 días	mié 23/12/15	jue 24/12/15	
61	2.3.1 - Modulos Healty				2 días	mié 23/12/15	jue 24/12/15	
62	2.3.1.2 - Administracion de Dietas				2 días	mié 23/12/15	jue 24/12/15	58,59
63	2.3.1.5 - Administracion de Mediciones				2 días	mié 23/12/15	jue 24/12/15	58,59
64	2.7 - Validacion del Diseño				1 día	vie 25/12/15	vie 25/12/15	57,60
65	2.8 - Documentacion del Diseño				0 días	vie 25/12/15	vie 25/12/15	64
66	3 - Desarrollo del Sistema				9 días	lun 28/12/15	jue 07/01/16	65
67	3.1 - Interfaz				2 días	lun 28/12/15	mar 29/12/15	
68	3.1.3 - Administracion de Dietas				2 días	lun 28/12/15	mar 29/12/15	65
69	3.1.3 - Administracion de Mediciones				2 días	lun 28/12/15	mar 29/12/15	65
70	3.4 - Funcional				5 días	mié 30/12/15	mar 05/01/16	
71	3.4.1 - Modulos Healty				5 días	mié 30/12/15	mar 05/01/16	
72	3.4.1.2 - Administracion de Dietas				5 días	mié 30/12/15	mar 05/01/16	68,69
73	3.4.1.5 - Administracion de Mediciones				3 días	mié 30/12/15	vie 01/01/16	68,69
74	3.5 - Entregable de Desarrollo				0 días	mar 05/01/16	mar 05/01/16	72,73
75	3.6 - Validacion de Desarrollo				2 días	mié 06/01/16	jue 07/01/16	74
76	3.8 - Documentacion de Desarrollo				0 días	jue 07/01/16	jue 07/01/16	75
77	4 - Pruebas del Sistema				3 días	vie 08/01/16	mar 12/01/16	76
78	4.3 - Funcionalidad del Sistema Web				2 días	vie 08/01/16	lun 11/01/16	
79	4.3.2 - Administracion de Dietas				2 días	vie 08/01/16	lun 11/01/16	76
80	4.3.5 - Administracion de Mediciones				1 día	vie 08/01/16	vie 08/01/16	76
81	4.4 - Datos				1 día	vie 08/01/16	vie 08/01/16	76
82	4.6 - Validacion de Pruebas				1 día	mar 12/01/16	mar 12/01/16	79,80,81
83	4.7 - Documentacion de Pruebas				0 días	mar 12/01/16	mar 12/01/16	82

1 - Analisis de Requerimientos

HEALTHY

SPRINT 1

SPRINT 2

2 - Diseño del Sistema

2.1 - Interfaz

2.1.3 - Administracion de Dietas

2.1.6 - Administracion de Mediciones

2.3 - Funcional

2.3.1 - Modulos Healty

2.3.1.2 - Administracion de Dietas

2.3.1.5 - Administracion de Mediciones

2.7 - Validacion del Diseño

25/12

3 - Desarrollo del Sistema

3.1 - Interfaz

3.1.3 - Administracion de Dietas

3.1.3 - Administracion de Mediciones

3.4 - Funcional

3.4.1 - Modulos Healty

3.4.1.2 - Administracion de Dietas

3.4.1.5 - Administracion de Mediciones

05/01

3.6 - Validacion de Desarrollo

07/01

4 - Pruebas del Sistema

4.3 - Funcionalidad del Sistema Web

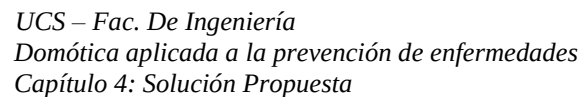
4.3.2 - Administracion de Dietas

4.3.5 - Administracion de Mediciones

4.4 - Datos

4.6 - Validacion de Pruebas

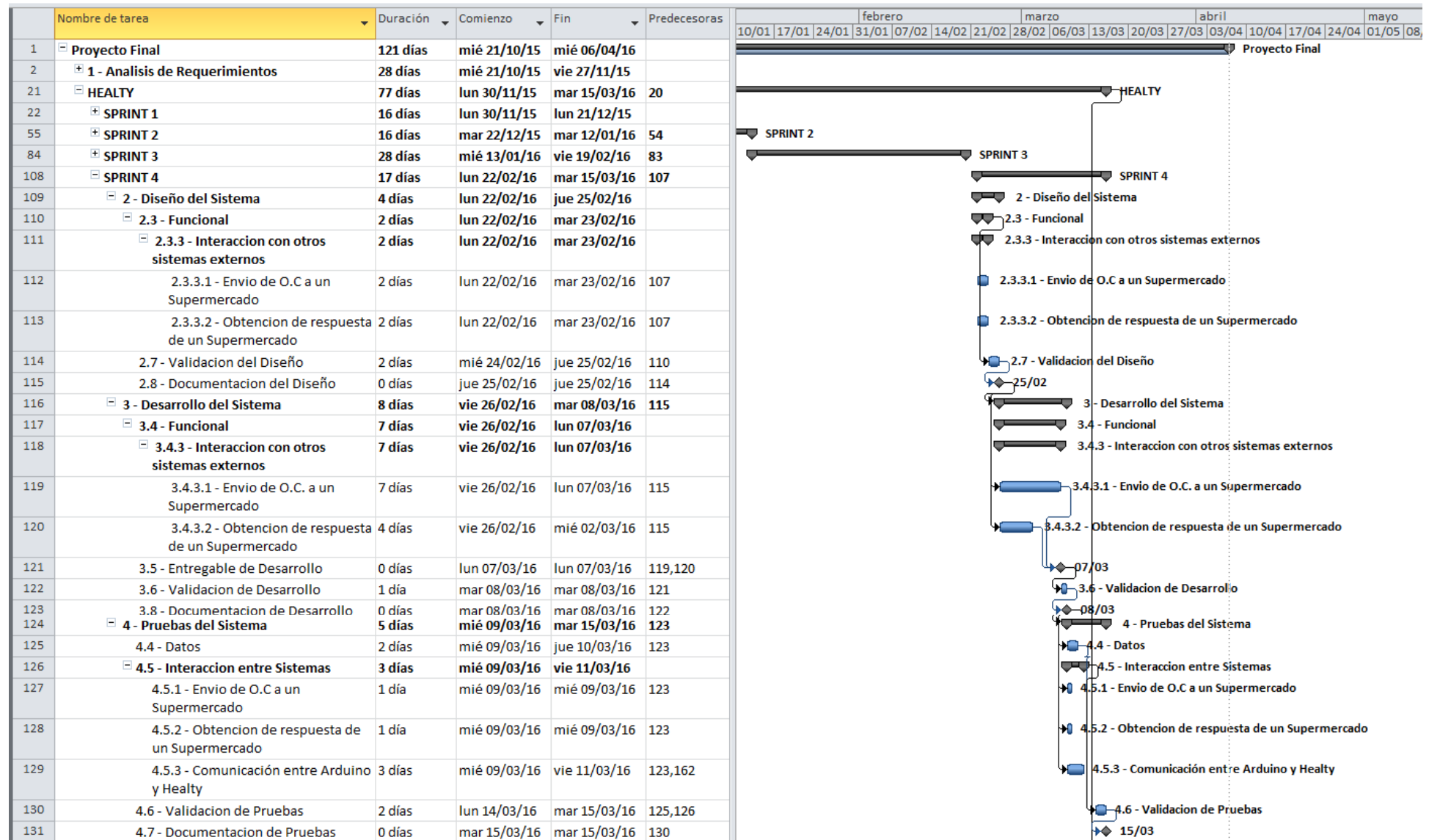
12/01



El diagrama de Gantt muestra el cronograma de un proyecto de desarrollo de software. El eje temporal abarca desde el 29/11 hasta el 27/03. El proyecto comienza con el SPRINT 1 (Análisis de Requerimientos) el 29/11. El SPRINT 2 (Diseño del Sistema) comienza el 06/12 y el SPRINT 3 (Desarrollo del Sistema) el 13/12. Las tareas principales y sus sub-tareas se detallan a continuación:

- SPRINT 1: 1 - Analisis de Requerimientos** (29/11 - 06/12)
 - 2 - Diseño del Sistema (06/12 - 13/12)
 - 2.1 - Interfaz (06/12 - 13/12)
 - 2.1.4 - Administración de Órdenes de Compra (06/12 - 13/12)
 - 2.3 - Funcional (13/12 - 20/12)
 - 2.3.1 - Módulos Healthy (13/12 - 20/12)
 - 2.3.1.3 - Administración de Ord. Compra (13/12 - 20/12)
 - 2.7 - Validación del Diseño (20/12 - 27/12)
- SPRINT 2: 2 - Diseño del Sistema** (06/12 - 13/12)
 - 3 - Desarrollo del Sistema (13/12 - 20/12)
 - 3.1 - Interfaz (13/12 - 20/12)
 - 3.1.4 - Administración de Órdenes de Compra (13/12 - 20/12)
 - 3.4 - Funcional (20/12 - 27/12)
 - 3.4.1 - Módulos Healthy (20/12 - 27/12)
 - 3.4.1.3 - Administración de Ord. Compra (20/12 - 27/12)
- SPRINT 3: 3 - Desarrollo del Sistema** (13/12 - 20/12)
 - 3.6 - Validación de Desarrollo (27/12 - 03/01)
 - 4 - Pruebas del Sistema (03/01 - 10/01)
 - 4.3 - Funcionalidad del Sistema Web (03/01 - 10/01)
 - 4.3.3 - Administración de O.C. (03/01 - 10/01)
 - 4.4 - Datos (10/01 - 17/01)
 - 4.6 - Validación de Pruebas (17/01 - 24/01)

El proyecto concluye el 19/02 en estado 'HEALTHY'.

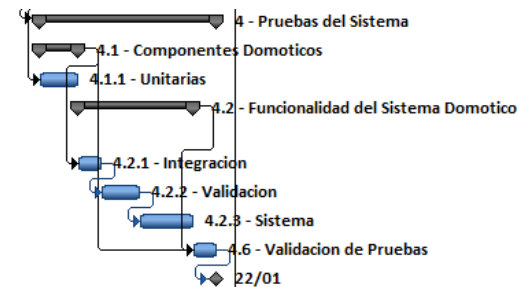




	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Gantt Chart Timeline											
						diciembre	enero	febrero	marzo								
1	Proyecto Final	121 días	mié 21/10/15	mié 06/04/16		22/11	29/11	06/12	13/12	20/12	27/12	03/01	10/01	17/01	24/01	31/01	07/02
2	1 - Analisis de Requerimientos	28 días	mié 21/10/15	vie 27/11/15													
21	HEALTHY	77 días	lun 30/11/15	mar 15/03/16	20												
132	SISTEMA DOMOTICO	40 días	lun 30/11/15	vie 22/01/16	20												
133	2 - Diseño del Sistema	10 días	lun 30/11/15	vie 11/12/15													
134	2.3 - Funcional	3 días	lun 30/11/15	mié 02/12/15													
135	2.3.2 - Arduino	3 días	lun 30/11/15	mié 02/12/15													
136	2.3.2.1 - Obtencion de Medicion	3 días	lun 30/11/15	mié 02/12/15	20												
137	2.3.2.1.1 - Obtención del pH	1 día	lun 30/11/15	lun 30/11/15													
138	2.3.2.1.2 - Procesamiento de Datos	1 día	mar 01/12/15	mar 01/12/15	137												
139	2.3.2.1.3 - Envio al Servidor	1 día	mié 02/12/15	mié 02/12/15	138												
140	2.4 - Sistema Domotico	4 días	jue 03/12/15	mar 08/12/15													
141	2.4.1 - Soporte	2 días	jue 03/12/15	vie 04/12/15	136												
142	2.4.2 - Posicionamiento de Componentes	2 días	lun 07/12/15	mar 08/12/15	141												
143	2.5 - Plan de Contingencias	3 días	lun 30/11/15	mié 02/12/15	20												
144	2.6 - Adquisicion de Componentes	0 días	mié 02/12/15	mié 02/12/15	143												
145	2.7 - Validacion de Diseño	3 días	mié 09/12/15	vie 11/12/15	134,140												
146	2.8 - Documentacion del Diseño	0 días	vie 11/12/15	vie 11/12/15	145												
147	3 - Desarrollo del Sistema	13 días	lun 14/12/15	mié 30/12/15	146												
148	3.3 - Sistema Domotico	5 días	lun 14/12/15	vie 18/12/15													
149	3.3.1 - Soporte	3 días	lun 14/12/15	mié 16/12/15	146												
150	3.3.2 - Posicionamiento de Componentes	2 días	jue 17/12/15	vie 18/12/15	149												
151	3.3.3 - Codificacion	5 días	lun 14/12/15	vie 18/12/15	146												
152	3.4 - Funcional	5 días	lun 21/12/15	vie 25/12/15													
153	3.4.2 - Arduino	5 días	lun 21/12/15	vie 25/12/15													
154	3.4.2.1 - Obtencion de Medicion	5 días	lun 21/12/15	vie 25/12/15	148												
155	3.4.2.1.1 - Obtencion del pH	2 días	lun 21/12/15	mar 22/12/15													
156	3.4.2.1.2 - Procesamiento de Datos	2 días	mié 23/12/15	jue 24/12/15	155												
157	3.4.2.1.3 - Envio al Servidor	1 día	vie 25/12/15	vie 25/12/15	156												
158	3.7 - Validacion de Sistema Domotico	3 días	lun 28/12/15	mié 30/12/15	148,152												
159	3.8 - Documentacion de Desarrollo	0 días	mié 30/12/15	mié 30/12/15	158												



160	4 - Pruebas del Sistema	17 días	jue 31/12/15	vie 22/01/16	159
161	4.1 - Componentes Domoticos	3 días	jue 31/12/15	lun 04/01/16	
162	4.1.1 - Unitarias	3 días	jue 31/12/15	lun 04/01/16	159
163	4.2 - Funcionalidad del Sistema Domotico	11 días	mar 05/01/16	mar 19/01/16	
164	4.2.1 - Integracion	3 días	mar 05/01/16	jue 07/01/16	161
165	4.2.2 - Validacion	3 días	vie 08/01/16	mar 12/01/16	164
166	4.2.3 - Sistema	5 días	mié 13/01/16	mar 19/01/16	165
167	4.6 - Validacion de Pruebas	3 días	mié 20/01/16	vie 22/01/16	161,163
168	4.7 - Documentacion de Pruebas	0 días	vie 22/01/16	vie 22/01/16	167



Nombre de tarea					Duración	Comienzo	Fin	Predecesoras	Gantt Chart Timeline (febrero a mayo)																							
1	Proyecto Final				121 días	mié 21/10/15	mié 06/04/16		Proyecto Final																							
2	1 - Analisis de Requerimientos				28 días	mié 21/10/15	vie 27/11/15																									
21	HEALTY				77 días	lun 30/11/15	mar 15/03/16	20	HEALTY																							
132	SISTEMA DOMOTICO				40 días	lun 30/11/15	vie 22/01/16	20	SISTEMA DOMOTICO																							
169	5 - Despliegue				16 días	mié 16/03/16	mié 06/04/16	21,132	5 - Despliegue																							
170	5.1 - Monitorizacion y Mantenimiento				5 días	mié 16/03/16	mar 22/03/16		5.1 - Monitorizacion y Mantenimiento																							
171	5.2 - Generacion de Informe Final				3 días	mié 23/03/16	vie 25/03/16	170	5.2 - Generacion de Informe Final																							
172	5.3 - Revision del Proyecto				3 días	lun 28/03/16	mié 30/03/16	171	5.3 - Revision del Proyecto																							
173	5.4 - Validacion del Despliegue				5 días	jue 31/03/16	mié 06/04/16	171,172	5.4 - Validacion del Despliegue																							
174	5.5 - Documentacion del Despliegue				0 días	mié 06/04/16	mié 06/04/16	173	06/04																							

Imagen 36: Diagrama de GANTT



Adquisiciones del Proyecto

La gestión de adquisiciones del proyecto, incluye todos aquellos procesos de compra o adquisición de los productos y/o servicios que son necesarios adquirir fuera del equipo del proyecto.

En este proyecto, los productos y servicios que se utilizarán son los siguientes:

- Servicios de transporte
- Servicios de internet
- Servicios de telefonía
- Insumos (lapiceras, papeles)
- Hardware:
 - Arduino UNO
 - Sensor de ultrasonidos HC-SR04
 - Sensor de colores TCS3200
 - Fuente de alimentación
 - ESP-8266
 - Servo
- Otros:
 - Protoboard
 - Transformador
 - Soporte para Arduino (Gabinete)
 - Pines macho y hembra
 - Cables de alimentación
 - Tiras reactivas
 - Notebooks

A la hora de hacer la planificación de las adquisiciones que se necesitan para el proyecto, hay que tener en cuenta el alcance, riesgos, cronograma y los costos que se tienen.

Se debe analizar y tomar una decisión. Si es necesario, comprar los elementos o se podrían hacer de manera satisfactoria por parte del equipo del proyecto. En nuestro caso, todo lo que es el hardware del sistema domótico y otras partes relacionadas con el mismo, se deben comprar y lo único que se realizó de forma manual es el soporte para Arduino (Gabinete).

La protoboard y los transformadores necesarios para el armado del circuito y alimentación del mismo, se pueden conseguir en casas de electrónica de la provincia. Las tiras reactivas se podrían conseguir en algunas farmacias, o algún comercio que haga la venta de elementos bioquímicos.

Lo que son servicios de internet y telefonía necesarios para la ejecución del proyecto, se realizan mediante un contrato de adhesión. Para la movilidad del equipo de proyecto, se hace el uso de servicios de transporte.

Otro punto importante es el criterio que se va a tener para la selección de los proveedores, a los cuales se les va a realizar la compra de estos insumos. Entre los criterios se debe tener en cuenta: costo de adquisición, formas de pago, tiempos de entrega, calidad del producto, calidad del proveedor, etc.



A la hora de efectuar las adquisiciones, ya se debe tener un listado de proveedores calificados según los criterios que se tienen a los cuales se podrá realizar la compra de los insumos. Este listado se obtiene mediante juicio de expertos, búsquedas en internet o publicidades relacionadas. Una vez establecida la comparativa entre los diferentes proveedores, se seleccionará el indicado para poder realizar el alquiler o compra de los insumos.

Como en la provincia de Salta no se encuentra muy desarrollado el tema de domótica, no existen locales o proveedores que hagan la venta de los componentes Arduino, así que los mismos se deberán solicitar desde otro lugar. Esto impacta directamente en los costos, calidad, riesgos y el cronograma del proyecto.

Los proveedores de insumos Arduino que tienen mejor reputación y cumplen con todo lo requerido para la realización del proyecto son:

- Patagonia Tec. CABA.
- uConnectME. Mendoza.
- DataSoft. CABA.

Todos estos proveedores fueron encontrados vía internet (Mercado Libre). Además de contar con un alto nivel de reputación en lo que es ventas y calidad de productos, nos ofrecen garantías por fallas en los dispositivos que nos están brindando. En lo que está relacionado a costos de adquisición, los precios son muy parecidos y no hacen una diferencia sustancial. Las formas de pago están dadas según las condiciones que impone Mercado Libre, y cada vendedor se adhiere o no a las mismas. Entre estas formas de pagos podemos encontrar: pago de contado, depósitos bancarios, pagos mediante tarjeta de crédito o mediante algún método electrónico (RapiPago, Pago Fácil). Uno de los puntos a tener en cuenta a la hora de adquirir el producto, es el tiempo de entrega que tienen los proveedores.

Los proveedores de insumos electrónicos se pueden encontrar en la provincia de Salta. Entre los más conocidos y que poseen los insumos necesarios contamos con:

- Gamma.
- Electrónica Urquiza.
- El Magnate.

Estos proveedores por conocimientos del equipo del proyecto trabajan de buena manera y venden productos de buena calidad. Se ofrecen garantías de insumos, buenos precios, distintas formas de pago y disponibilidad de materiales. Este punto de disponibilidad es de suma importancia ya que por algún desperfecto del insumo, se podría ir a cambiarlo y no perder tiempos en envíos al encontrarse dichos locales en la capital de la provincia.

Por investigaciones, el único proveedor que tenía disponibilidad de tiras reactivas en la capital de la provincia es la casa de venta de artículos bioquímicos “Cooperativa Bioquímica de Salta”. Existen otros proveedores de elementos de medición fuera de la provincia como ser:

- LAB Shop. Buenos Aires.
- Tu medición. Buenos Aires.



La relación de precios que existe entre el producto encontrado en la provincia y los encontrados en estas casas de venta de insumos químicos, no varían demasiado. Lo que sí varía, es la calidad del mismo ya que existen varios fabricantes de estas tiras reactivas. Pero a fines del proyecto, este punto no es de tanta importancia. Otro punto a tener en cuenta, es el tiempo que se demora el envío desde otra provincia.

Los servicios de telefonía e internet (Telecom Argentina y Arnet), ya se encuentran disponibles en el lugar donde se efectúa el desarrollo del proyecto, por lo que estos datos no serán necesarios para realizar una adquisición.

Teniendo toda esta información, tendremos un panorama sobre el calendario de recursos en relación a la disponibilidad de los elementos que se compraron. Al realizar la compra en otras provincias del país, tiene su tiempo de demora y a su vez gastos en el envío de los insumos.

En función de los recursos que se van a adquirir desde otra provincia, se eligió a Patagonia TEC como proveedor para el envío de todos los elementos relacionados con Arduino (Arduino UNO, sensor de ultrasonidos HC-SR04, sensor de colores TCS3200, servo, ESP-8266, Pines macho-hembra y cables de alimentación para Arduino), ya que poseían todo en stock, y se pudo tener contacto mediante distintos medios de comunicación.

Para los insumos electrónicos (Protoboard y Transformador), se eligió a Gamma para la compra. Ambos se encontraban en stock al momento de realizar la misma.

Las tiras reactivas fueron adquiridas en la provincia de Salta; se eligió al proveedor “Cooperativa Bioquímica de Salta”, ya que contaba con las tiras reactivas al momento de realizar la compra.

Una vez efectuadas estas adquisiciones se deben administrar. Esta administración se basa en gestionar las relaciones de las adquisiciones, los sistemas de pago de estos insumos, administrar los reclamos, supervisar el desempeño del contrato y efectuar cambios y correcciones según sea necesario.

Se debe asegurar que el producto que se compró, sea acorde a lo que el vendedor nos ofreció y a nuestras necesidades. Además de cumplir con todas las obligaciones contractuales tanto por parte del vendedor como nuestra.

El administrar las adquisiciones sirve también para monitorear el alcance, costo, cronograma, riesgos y la calidad de lo adquirido. También se podrían realizar peticiones de cambio por el incumplimiento en el contrato, calidad de adquisiciones, precios, tiempos de entrega, descripción de los productos adquiridos o resultados esperados.

Con el proveedor Patagonia TEC no hubo ningún inconveniente en términos contractuales. Todos los productos que nos envió cumplían con las especificaciones del mismo y la calidad es buena. El envío llegó acorde a los tiempos previstos por el servicio de correo (OCA Postal), y se estableció como forma de pago el uso de tarjeta de crédito.

Con el proveedor Gamma tampoco hubo inconvenientes y los productos adquiridos cumplían con las expectativas y la calidad necesaria para la realización del proyecto. Como



los productos se encontraban en stock, no hubo demoras en la entrega y a su vez, el pago de los mismos se efectuó de contado.

Con el proveedor Cooperativa Bioquímica de Salta no se encontraron inconvenientes, y el producto adquirido cumple con las normas necesarias para la realización del proyecto. No hubo demoras en la entrega y el pago se realizó de contado.

Como no hubo inconvenientes, no se necesitaron realizar peticiones de cambio por fallas o incumplimientos en el contrato. A su vez, se hicieron comparativas entre las líneas base que se tenían desde lo que se planificó y no hubo impactos significativos en el proyecto.

Si todo se cumple con lo pactado, se deberán cerrar las adquisiciones. Esto hace referencia a verificar la totalidad de los productos que se solicitaron, y que cumplan con todo lo necesario. Esto debería finalizar los reclamos abiertos (si existieran) y archivar esta información para un uso futuro.

Al verificar que todos los productos que se solicitaron a estos proveedores cumplen con las expectativas, se cierran todas las adquisiciones y se tienen los contactos de los mismos para una futura compra.

Riesgos del Proyecto

La gestión de riesgos, tiene por objetivo, disminuir la probabilidad y el impacto de eventos negativos que afecten al proyecto.

En este caso, se tratará de identificar y describir los riesgos que fueron identificados, como así también un plan de contingencia por si ocurriese.

- Disponibilidad horaria por parte del equipo del proyecto: ya que el proyecto será realizado en el horario de la tarde, los integrantes poseen un horario de trabajo matutino y esto podría perjudicar la correcta ejecución del proyecto.
Plan de contingencia: se tratará de convenir horarios por la tarde y si alguno de los integrantes no puede en el horario vespertino, habrá una comunicación para realizar consultas y temas relacionados con la problemática que se posee. A su vez, se realizarán reuniones para ver el avance del proyecto.
- Cambios en los requerimientos: se pueden provocar cambios en los requerimientos cuando se analizó el problema y esto puede resultar un inconveniente para los tiempos y costos del proyecto.
Plan de contingencia: se deben especificar correctamente y de manera detallada los requerimientos, por si ocurriese un cambio en alguno de ellos. De esta manera, a la hora de formular los requerimientos, lo más conveniente sería que fueran validados por todo el equipo del proyecto.
- Pérdida de personal del equipo del proyecto: este es un riesgo que puede ser perjudicial para el proyecto, en relación al rol que desempeñaba la persona que se pierde. Puede provocar un retraso en los tiempos que habían sido planificados.
Plan de contingencia: Se tratará de dar constantes informes de avances al equipo del proyecto, para que cada uno sepa en qué situación se encuentra y qué tarea está



realizando cada persona. Si alguna persona del equipo del proyecto se marchara, el equipo tiene un conocimiento sobre el proyecto en general.

- Falla en componentes hardware: este es un riesgo que puede ocurrir y puede desviar la planificación temporal y además generar un costo adicional.
Plan de contingencia: Se tratará de tener a disposición alternativas tecnológicas que puedan suplir este inconveniente, tanto en los componentes de Arduino como en las notebooks para el desarrollo del proyecto.
- Compra de componentes: como la mayoría de los componentes para realizar este sistema domótico se deben comprar en otras provincias, hay que tratar de buscar una confianza en el proveedor y precios razonables. El riesgo puede ser que al no conocer el vendedor, no te envíen los productos o que las especificaciones no están bien detalladas, y se realice mal la compra o alquiler del insumo.
Plan de contingencia: en mercado libre, que es el lugar donde se realizará la búsqueda del proveedor, existen comentarios y cada proveedor tiene una reputación. Esto será de gran ayuda para que no ocurra ningún problema a la hora del alquiler o compra de un insumo, seleccionando correctamente al proveedor.
- Disponibilidad de componentes: los componentes Arduino son de origen extranjero, por lo que con el problema de importaciones, puede ser que no se encuentren muchos proveedores que nos brinden disponibilidad de los elementos que se necesitan para la realización del proyecto. El riesgo puede ser que si no se encuentra alguno de los componentes necesarios, el alcance del proyecto no se pueda cumplir.
Plan de contingencia: existen otras tecnologías con las cuales se podría realizar el proyecto, aunque habría que investigar sobre el uso de la misma y esto llevaría a que el tiempo, calidad y costos del proyecto se vean modificados.
- Prueba técnicas en componentes: los distintos módulos de Arduino adquiridos, inclusive la misma placa Arduino, deben ser sometidos a una serie de pruebas con el fin de asegurar su correcto comportamiento. Esto incluye el riesgo de manipulación errónea de los componentes, lo cual puede ocasionar el deterioro de los mismos por falta de conocimiento o inexperiencia del personal asociado al proceso de desarrollo.
Plan de contingencia: para reducir este riesgo a su menor medida, se debe investigar cada módulo de hardware previamente a su manipulación. Esto se debe hacer consultando a la hoja de datos que viene asociada a cada componente, por parte del fabricante.
- Codificación del software: la complejidad del proyecto involucra inevitablemente acarrear errores en el código, debido a la gran cantidad de líneas codificadas. Esto tiene consecuencias muy relevantes en un sistema final, por lo cual no serán aceptables.
Plan de contingencia: realizar pruebas tanto de caja negra como de caja blanca para asegurar el correcto funcionamiento del código.
- Fluctuaciones Económicas: los componentes Arduino son de origen extranjero por lo que los precios de estos insumos están ligados en relación a una moneda extranjera (Dólar, Euros).
Plan de contingencia: en el caso de una fluctuación económica que den lugar a un aumento desmedido de los precios de los insumos, se deberá plantear esta situación y



proceder a replantear los costos asociados al proyecto, brindando alternativas tecnológicas para poder sobrellevar esta circunstancia.

- Pruebas no suficientes: a la hora de realizar las pruebas del sistema domótico, puede ser que no se tomen en cuenta todas las variaciones o posibilidades de los diferentes resultados esperados por dicho equipo.
Plan de contingencia: para subsanar los errores que podrían causar las pruebas insuficientes o casos de pruebas no representativas, luego de la implementación del proyecto, se proporcionará un mantenimiento correctivo al sistema domótico.

Matriz de Probabilidad e Impacto

Ahora que se tienen los riesgos, se va a realizar una matriz de probabilidades e impactos, para clasificar los riesgos por un orden de prioridad según las implicaciones potenciales que pueden perjudicar los objetivos del proyecto.

Se puede observar la matriz de probabilidad e impacto en la tabla 8.

Riesgo	Ocurrencia	Impacto	Valoración	Prioridad
Disponibilidad horaria por parte del equipo del proyecto	3	3	9	Media
Cambios en los Requerimientos	2	2	4	Baja
Pérdida de personal del equipo del proyecto	2	5	10	Media
Compra de Componentes	3	3	9	Media
Disponibilidad de Componentes	3	3	9	Media
Pruebas técnicas en componentes	3	3	9	Media
Codificación del software	3	4	12	Alta
Fluctuaciones Económicas	4	2	8	Media
Pruebas no Suficientes	3	5	15	Alta

Tabla 8: Matriz de Probabilidad e Impacto

Entre las columnas de la matriz de probabilidades e impacto se tiene:

- Riesgo: nombre del riesgo.
- Ocurrencia: el grado de ocurrencia puede obtener valores entre (1: más bajo y 5: más alto)
- Impacto: el grado del impacto puede obtener valores entre (1: más bajo y 5: más alto)
- Valoración: es el producto entre la ocurrencia y el impacto que posee el riesgo.



- Prioridad: según el valor que se obtuvo en la columna valoración, se tienen diferentes valores de prioridad.
 - Baja: valoración entre 1 y 5.
 - Media: valoración entre 6 y 10.
 - Alta: valoración igual o mayor a 11.

Comunicaciones del Proyecto

La gestión de comunicaciones en el proyecto garantiza que la generación, recopilación, distribución, almacenamiento, recuperación y disposición de la información del proyecto sea adecuada y oportuna.

Entre las diferentes comunicaciones con todo el equipo del proyecto y usuarios expertos se encuentran:

- Internas: entre todo el personal que está dentro del equipo del proyecto.
- Externas: con usuarios expertos.
- Formales: mediante informes.
- Informales: conversaciones, correos electrónicos.
- Escrita y orales

Calidad del Proyecto

La gestión de calidad, se aplica tanto para el proyecto como para el prototipo del producto. Se documenta en dicho trabajo final la manera en que el proyecto demostrará el cumplimiento de los requisitos y las normas de calidad aplicadas.

Esta documentación se basa en las metodologías empleadas como así también, en las formas de plantear una solución a los requerimientos que se poseen. A su vez, se debe monitorear que todas las actividades se ejecuten de una manera correcta, a fin de evaluar el desempeño y ver si se requieren cambios.

El no cumplir con los requisitos de calidad, puede tener consecuencias negativas en el proyecto, como ser:

- El equipo trabaje en exceso para cumplir con los requerimientos.
- No hacer un seguimiento detallado de las tareas a realizar, puede provocar errores en la ejecución de las mismas.



Integración del Proyecto

La gestión de integración del proyecto, incluye los procesos y actividades necesarios para identificar, definir, combinar, unificar y coordinar los diversos procesos y actividades de las otras ocho áreas de conocimiento. Entre las tareas que se llevaron a cabo por el equipo tenemos:

- Analizar y comprender el alcance.
- Modo en que se va a utilizar toda la información identificada y transformarla luego en un plan, según el enfoque estructurado que propone la guía del PMBOK.
- Realizar actividades para producir los entregables del proyecto.
- Medir y monitorear todos los aspectos de avance del proyecto, y realizar acciones apropiadas para cumplir con los objetivos del mismo.

Evaluación Económica - Financiera del Proyecto

Costos del Proyecto

La gestión de costos del proyecto, describe los procesos involucrados en planificar, estimar, presupuestar y controlar los costos, para que el proyecto se complete con el presupuesto aprobado.

Costos de Recursos Humanos

Según la gestión de los recursos que están involucrados en el desarrollo del proyecto y a su vez a la planificación temporal del mismo, en la tabla 9 se pueden observar los costos de los recursos humanos.

La tabla 9 cuenta con diferentes columnas que se detallan a continuación:

- Recurso: Persona del equipo de trabajo con un rol determinado.
- \$ / Hora: Representa el monto por honorarios estimado para cada uno de los recursos por hora hombre (según COPAIPA).
- Total Horas: Representa la cantidad de horas trabajadas en el proyecto.
- Subtotal: Representa el producto entre \$ / Hora y Total Horas.

Recurso	\$ / Hora	Total Horas	Subtotal
Analista	\$150,00	270	\$40.500
Diseñador	\$120,00	198	\$23.760
Desarrollador 1	\$120,00	392	\$47.040
Desarrollador 2	\$120,00	392	\$47.040
TOTAL			\$158.340

Tabla 9: Costos de Recursos Humanos



Costos de Adquisiciones

En las tablas 10, 11 y 12, se pueden observar diferentes costos relacionados con las adquisiciones del proyecto. Como se planteó en apartados anteriores, la duración del proyecto es de 121 días hábiles.

Adquisición	Costo
Servicios de Transporte	\$ 2000
Servicios de Internet	\$ 1200
Servicios de Telefonía	\$ 1200
Insumos	\$ 1000
TOTAL	\$ 5400

Tabla 10: Costos de Adquisiciones

Los costos relacionados con los componentes de Arduino son:

Adquisición	Costo
Arduino UNO	\$ 260
Sensor de Ultrasonidos	\$ 65
Sensor de Colores	\$ 230
ESP8266	\$ 200
Fuente de Alimentación	\$ 100
Servo	\$ 250
Pines Macho-Hembra	\$ 75
Cables de Alimentación	\$ 110
TOTAL	\$ 1290

Tabla 11: Costos de Adquisiciones (Arduino)



Los costos relacionados a otros insumos son:

Adquisición	Costo
Protoboard	\$ 90
Tiras Reactivas	\$ 200
Transformador	\$ 100
TOTAL	\$ 390

Tabla 12: Costos de Adquisiciones

Costos de Mantenimiento

Existen cuatro tipos de mantenimiento: correctivo, adaptativo, perfectivo y preventivo, pero a los fines del proyecto solamente se verán involucrados los siguientes:

- **Mantenimiento Preventivo:** se realizó la compra de algunos insumos extras por si había algún problema con los mismos y no se perdía tiempo, hasta que los proveedores nos hicieran la reposición de aquellos componentes que presentaban una falla y los cubría una garantía.
- **Mantenimiento Perfectivo:** una vez que fue presentado el sistema domótico y funciona de una manera correcta y aceptable según los objetivos que fueron planificados, podría existir la necesidad de profundizar dicho proyecto. Esto generaría el plantear nuevas tareas y objetivos específicos. Este tipo de mantenimiento podría ser útil para futuras investigaciones.

Costos Totales

En la tabla 13, se pueden observar los costos totales para la realización del proyecto.

Costo	Subtotal
RRHH	\$158.340
Adquisiciones	\$ 7.080
TOTAL	\$ 165.420

Tabla 13: Costos Totales



Flujo de Fondos

Un flujo de fondos está relacionado a los movimientos de entrada y salida de efectivo generados por las operaciones económicas, financieras y comerciales. En la tabla 14, se puede observar el flujo de fondos asociado al proyecto.

TNA	24%				Inflación anual	30%	Empleados	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	4		
TEM	0,02				Inflación mensual	0,025	K Ventas	5	7	10	15	18	20	22	24	25	30	33	35	35	38	40	45	48	50
Periodo	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
RRHH		-26390	-26390	-26390	-26390	-26390	-26390																		
Desarrollador								-14400	-14400	-14400	-14400	-14400	-14400	-14400	-14400	-28800	-28800	-28800	-28800	-28800	-28800	-28800	-28800	-28800	-28800
Analista								-18000	-18000	-18000	-18000	-18000	-18000	-18000	-18000	-18000	-18000	-18000	-18000	-18000	-18000	-18000	-18000	-18000	-18000
Soporte Tecnico																									-11000
Costos Fijos																									
Serv. Transporte		-333	-333	-333	-333	-333	-333	-400	-400	-400	-400	-400	-400	-400	-400	-600	-600	-600	-600	-600	-600	-600	-600	-600	-800
Serv. Internet		-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200
Serv. Telefonía		-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200	-200
Insumos		-167	-167	-167	-167	-167	-167	-200	-200	-200	-200	-200	-200	-200	-200	-300	-300	-300	-300	-300	-300	-300	-300	-300	-400
Alquiler Oficina								-3800	-3800	-3800	-3800	-3800	-3800	-3800	-3800	-3800	-3800	-3800	-3800	-3800	-3800	-3800	-3800	-3800	-3800
Muebles												-6000													-6000
PC												-13000													-13000
Costos Variables								-5000	-7000	-10000	-15000	-18000	-20000	-22000	-24000	-25000	-30000	-33000	-35000	-35000	-38000	-40000	-45000	-48000	-50000
TOTAL EGRESOS	-1680	-27290	-27290	-27290	-27290	-27290	-27290	-42200	-44200	-47200	-52200	-74200	-57200	-59200	-61200	-76900	-81900	-84900	-86900	-86900	-89900	-91900	-96900	-99900	-132200
TOTAL INGRESOS	0	0	0	0	0	0	0	25000	35000	50000	75000	90000	100000	110000	120000	125000	150000	165000	175000	175000	190000	200000	225000	240000	250000
Total	-1680	-27290	-27290	-27290	-27290	-27290	-27290	-17200	-9200	2800	22800	15800	42800	50800	58800	48100	68100	80100	88100	88100	100100	108100	128100	140100	117800
Total Fondo Desc	-1680	-28970	-56260	-83550	-110840	-138130	-165420	-182620	-191820	-189020	-166220	-150420	-107620	-56820	1980	50080	118180	198280	286380	374480	474580	582680	710780	850880	968680
Ajuste Inflación																									
Total	-1680	-26754,9	-26230,3	-25716	-25211,7	-24717,4	-24232,739	-14973,6	-7852,11	2342,915	18703,94	12707,36	33747,51	39270,05	44563,05	35739,01	49607,16	57204,42	61684,049	60474,55	67364,43	71321,77	82860,08	88845,24	73238,79
Total Fondo Desc	-1680	-28434,9	-54665,2	-80381,2	-105593	-130310	-154543,049	-169517	-177369	-175026	-156322	-143615	-109867	-70597	-26034	9705,037	59312,2	116516,6	178200,659	238675,2	306039,6	377361,4	460221,5	549066,7	622305,5

Tabla 14: Flujo de Fondos



Los **costos variables** hacen referencia a los componentes necesarios para la realización del equipo domótico, entre ellos se encuentran: Arduino UNO, Sensor de colores, sensor de ultrasonido, ESP8266, Fuente de alimentación, servo, pines macho-hembra, cables de alimentación, plaquetas, transformador y tiras reactivas. Los mismos son **variables** porque dependen del volumen de producción que se tiene por cada período de tiempo.

Los **totales de ingreso** están asociados a las cantidades estimadas de ventas mensuales del sistema domótico. Estas ventas empiezan en el **periodo 7**, que es el momento cuando el producto sale al mercado. El producto en sí incluye: equipo domótico (con el agregado de 100 tiras reactivas para la realización de pruebas y un manual de uso en conjunto al uso del sistema Healty para el paciente y su grupo familiar asociado. La adquisición del producto, también brinda un servicio de atención al cliente ante cualquier duda o inconveniente con el sistema domótico.

Todos los cálculos realizados en el flujo de fondos están realizados en base a las siguientes variables:

- TNA : Tasa Nominal Anual = 24%
- TEM : Tasa Efectiva Mensual = 0.02
- Inflación Anual = 30%
- Inflación Mensual = 0.025

La utilidad del flujo de fondos, es la posibilidad de poder calcular los indicadores económicos en la evaluación de un proyecto de inversión. Estos indicadores son:

- **VAN (Valor Actual Neto):** Este indicador permite conocer en términos de “pesos de hoy” el valor total de un proyecto que se extenderá por un determinado tiempo (meses o años), y que puede combinar flujos positivos (ingresos) y negativos (egresos).

El cálculo de la **VAN** del proyecto tiene un valor de: **\$622.305,5**. Como la VAN es mayor a 0 (cero) podemos decir que dicho **proyecto es rentable**.

- **TIR (Tasa Interna de Retorno):** Este indicador está relacionado con la VAN y puede entenderse como la tasa de interés máxima a la que es posible endeudarse para financiar el proyecto, sin que se generen pérdidas.

El cálculo de la TIR del proyecto tiene un valor del **5%**.



Inversión Inicial

Habiendo analizado todos los recursos relacionados al desarrollo del proyecto, se puede observar que se necesita una inversión inicial para la realización del mismo.

Los recursos de hardware de Arduino junto a algunos insumos que son necesarios para el armado del circuito (protoboard, pines macho-hembra, transformador, tiras reactivas, cables de alimentación) desde el inicio del proyecto, representan una inversión inicial que se puede observar en el flujo de fondos como un egreso (periodo cero) y es de \$1680.

Plan de Pago

El costo total del proyecto como pudo observarse en la tabla 13, está compuesto por los costos relacionados a los recursos humanos involucrados en la realización del proyecto y las adquisiciones del hardware y otros, que son gastos previstos por el equipo de proyecto para la realización de dicho trabajo.

Los componentes hardware fueron abonados mediante tarjetas de crédito y las tiras reactivas se abonaron en efectivo.

Beneficios del Proyecto

Los beneficios que va a producir el proyecto se podrían dividir en dos partes: beneficios cualitativos y beneficios cuantitativos:

Beneficios cualitativos

- Obtener una comprensión adecuada para la aplicación de manera correcta de la domótica.
- Obtener experiencia en la gestión de proyectos aplicando la guía de buenas prácticas como el PMBOK, y para el desarrollo de sistemas utilizando una metodología ágil como SCRUM.
- Sacar conclusiones en relación al estado de salud de las personas que hacen uso del sistema domótico.
- Obtención de hábitos para los usuarios finales que ayuden a llevar una vida saludable.
- Al ser un sistema amigable para la persona, esto permitirá a los usuarios finales no necesitar de un conocimiento avanzado en temas informáticos, para la utilización del mismo.
- Cuando un usuario necesite realizar su dieta, las compras a un supermercado podrá realizarla de manera directa mediante tarjeta de crédito.



Beneficios cuantitativos

En base a las estimaciones de ventas que se realizaron en el flujo de fondos, se podrían observar los ingresos y egresos monetarios para los diferentes periodos de tiempo que se plantearon. Con el cálculo de los indicadores económicos (TIR y VAN) se pudo determinar que el proyecto es rentable, por lo que es un buen indicio para la realización del mismo.

Lo que sí se podría garantizar es que este sistema domótico que se está desarrollando, va a ser de gran ayuda para el usuario final, ya que haciendo una comparativa entre el costo/beneficio, realizando una mínima inversión en aspectos monetarios obtendrá muchísimos beneficios para controlar su estado de salud y también ayudar de una manera práctica, para que las personas incluyan nuevas costumbres en su vida diaria para prevenir enfermedades.

Alternativas Tecnológicas

A continuación, se va a detallar, las distintas formas con las cuales se podría realizar este proyecto, en base a un análisis realizado por el equipo; cuáles podrían ser alternativas tecnológicas que son considerables para la ejecución y resolución del problema que se tiene. Además, se explican las justificaciones del porqué se eligió una alternativa y no otra.

Alternativas de Hardware: al hablar de alternativas de hardware, hacemos referencia a la tecnología utilizada para la realización del sistema domótico. Entre ellas nos encontramos con:

- **Arduino:** es una plataforma de prototipos electrónica de código abierto (*open – source*) basada en *hardware* y *software* flexibles y fáciles de usar. Es una de las tecnologías más utilizadas en la actualidad.
- **Raspberry PI:** Raspberry PI es una pequeña computadora basada en una placa de circuito impreso y equipada con un procesador ARM, sistema de video integrado HDMI, conexión Ethernet y entradas USB para conectar periféricos y pantallas. Es un miniordenador con el tamaño de una tarjeta de crédito, bastante económico (más que BeagleBone, pero menos que Arduino), desarrollada en Reino Unido por la Fundación Raspberry Pi, con el objetivo principal de estimular la enseñanza de ciencias de la computación en las escuelas. Al ser un miniordenador, podemos utilizarlo para desarrollar cosas bastante más complejas que con Arduino y puede utilizar lenguajes de programación de alto nivel como Python, C++ y Java.
- **BeagleBone:** Es un ordenador pequeño del tamaño de una tarjeta de crédito, donde se puede ejecutar un sistema operativo, como ser: Linux/Android 4.0. Su principal diferencia con Arduino, es que puede ejecutar un pequeño sistema operativo; es prácticamente un miniordenador donde se pueden ejecutar programas sobre estos sistemas operativos. BeagleBone. Está diseñado para funcionar a un nivel mucho más alto y tiene mucha más capacidad de proceso que Arduino.
- **Netduino:** es una placa de desarrollo *open source*, que utiliza el .net Micro Framework como *runtime* por lo que puede ser programado mediante el IDE Visual C# (C-Sharp) Express Edition. Es una plataforma hardware que facilita la implementación de robots y pequeños autómatas, con lenguajes de alto nivel



(C#). Es un dispositivo que contiene un microprocesador, un interfaz de salida y entrada, un led y un botón físico.

Alternativas de Software: en este punto, se tienen en cuenta diferentes software que pudieron ser utilizados para la ejecución correcta del proyecto.

- **Sistemas Operativos:**

- **Microsoft Windows:** familia de distribuciones de software para PC, smartphones y servidores, desarrollados y provistos por la empresa Microsoft, y disponibles para diferentes arquitecturas. Hoy en día, los Sistemas Operativos brindados por la empresa Microsoft (7, 8, 10) son de los más utilizados en el mundo en sus diferentes versiones (*Starter, Home Basic, Home Premium, Professional, Ultimate, Enterprise*).
- **Linux:** constituye una alternativa adicional, al tratarse de un sistema operativo normalmente distribuido bajo una licencia libre o de código abierto. Posee numerosas distribuciones, pero entre las más estables y reconocidas, se puede destacar a Ubuntu. Esta distribución puede ser instalada en dispositivos (*tablets*, televisores, teléfonos y computadoras tradicionales) de medianos recursos; sin embargo su uso no ha sido de gran difusión hasta el momento en nuestra región o provincia.
- **Mac OS:** constituye una alternativa adicional. Es el nombre del sistema operativo creado por Apple para la línea de computadoras Macintosh. Es conocido por haber sido uno de los primeros sistemas operativos en contar con una interfaz gráfica, compuesta por la interacción de mouse con ventanas, iconos y menús.

- **DBMS:**

- **MySQL:** a nivel global, se trata del DBMS relacional más utilizado actualmente. Es de distribución libre, es robusto y posee muy buen soporte de usuarios en la red. Es normalmente utilizado en el desarrollo de sistemas web al ser multiproceso.
- **PostgreSQL:** se trata de un DBMS orientado a objetos, publicado bajo la licencia PostgreSQL y con un alto crecimiento en los últimos tiempos. No es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores. Permite una alta concurrencia, es robusto y versátil, en general muy similar a MySQL.
- **Microsoft SQL Server:** se trata de un DBMS relacional desarrollado por la empresa Microsoft, que se caracteriza por su gran potencial para desarrollos de alta complejidad. Se destaca por su robustez y confiabilidad en la gestión de transacciones a las bases de datos.

- **Lenguajes de Programación:**

- **Ruby:** es un lenguaje de programación dinámico y de código abierto enfocado en la simplicidad y productividad. Es considerado un lenguaje flexible, ya que permite a sus usuarios alterarlo libremente.
- **PHP:** es un lenguaje de programación de uso general de código del lado del servidor, originalmente diseñado para el desarrollo web y de contenido dinámico. Es considerado uno de los lenguajes más flexibles, potentes y de alto rendimiento. Forma parte del software libre publicado bajo la licencia PHP.



- **ASP.Net:** es un lenguaje de programación de servidores para generar páginas web dinámicamente. Nos permite separar en las páginas webs la parte de diseño que contiene la página, no interviniendo para nada el código HTML. Es adecuado para el acceso a las bases de datos, ficheros, etc.
- **IDE para desarrollo:**
 - **Sublime Text:** Es un editor de texto avanzado para programadores, multiplataforma. Se necesita invertir tiempo en aprender a dominarlo, pero luego de aprenderlo, no se querrá dejarlo, ya que aumenta mucho la productividad a la hora de escribir código. No ofrece muchas de las características de ayuda al programador web que sí ofrecen la mayoría de los otros entornos, por lo que es recomendable sobre todo si vamos a trabajar mucho directamente con código, especialmente JavaScript. Lo podemos descargar y utilizar gratuitamente, pero si lo vamos a usar a menudo para trabajar, sus creadores nos solicitan que les paguemos una licencia, la cual tiene un precio razonable.
 - **Aptana:** Aptana Studio 3 es una herramienta de desarrollo de código abierto para programadores que tengan experiencia en Ruby. Permite a los desarrolladores probar sus aplicaciones web, usando un único entorno. Aptana soporta las últimas tecnologías para navegadores con HTML5, CSS3, JavaScript, Ruby, Rails, PHP y Python. Cuenta con la capacidad para publicar automáticamente tus aplicaciones Ruby & Rails a servicios de alojamiento como Heroku y Engine Yard.
 - **NetBeans:** NetBeans es un IDE de código abierto y una plataforma que ayuda a los desarrolladores a implementar rápidamente páginas web, aplicaciones de escritorio y aplicaciones para dispositivos móviles. En realidad, NetBeans está pensado para desarrollar en Java aunque también es compatible con JavaFX, PHP, JavaScript, Ruby & Ruby on Rails, Groovy and Grails y C / C ++.
 - **Microsoft Visual Studio Express for Web:** Está disponible en varios idiomas (incluyendo el español) y su única limitación, es que sólo funciona bajo Windows. Es decir, que si trabajas en Mac o Linux no podrás utilizarlo. Por lo demás es un entorno integrado de desarrollo (IDE) muy completo. Es una herramienta técnica orientada a programadores y con muchas otras posibilidades. Ofrece un soporte estupendo para diseño de páginas Web, con editores avanzados de HTML, CSS y JavaScript, capacidad de depuración, ayuda contextual para todos los lenguajes, validadores de código, validadores de accesibilidad, visualización real de cómo van a quedar las páginas, etc. Es totalmente gratuito.

Elección de Alternativas

Entre las diferentes alternativas que se plantean para poder realizar de forma adecuada y correcta, se seleccionaron las siguientes y se hace un detalle del porqué de esta elección:

- **Hardware:** Para la implementación del sistema domótico se seleccionó a la tecnología Arduino. Dicha elección se debe a varias razones:



- A la hora de realizar proyectos con Arduino, lo primero que se puede apreciar es la fácil curva de aprendizaje. Interactuar con los sistemas de Arduino no es un impedimento a la hora de desarrollar un proyecto. Todos los elementos están detallados completamente en la red, y son de fácil acceso. Sumado a esto se encuentra una gran cantidad de tutoriales con proyectos ya realizados, de los cuales uno puede tomar como base, mezclar, modificar, y hasta mejorar a su gusto. Sin ir más lejos, el IDE oficial de Arduino se puede descargar con una serie de ejemplos ya cargados.
- Arduino dispone de una gran comunidad *Open-Source*, lo que facilita el acceso a librerías ya creadas. Muchos proyectos de Arduino se encuentran disponibles de manera gratuita en internet, lo cual agiliza la creación de un proyecto.
- Un entorno de desarrollo minimalista, simple, sencillo, facilita la creación de proyectos que no involucran miles de líneas de código. Podemos descargar rápidamente el IDE de Arduino y comenzar a programar en no más de 5 minutos.
- Arduino es un sistema programable que no necesita estar conectado a una computadora para funcionar. El trabajo del programador es desarrollar la lógica del sistema, cargarlo a la placa, y automáticamente Arduino se independiza de la PC, permitiéndole a este funcionar solo con la alimentación de energía requerida.
- El equipo del proyecto ya contaba con un conocimiento previo en lo que es el uso de esta tecnología.
- Los componentes Arduino se pueden conseguir en el país a un bajo costo y con una buena disponibilidad. Estos componentes no están exactamente en la provincia, pero mediante envíos que no tienen mucha duración, se pueden adquirir.
- Software: Para la implementación del sistema web se seleccionaron las siguientes tecnologías.
 - IDE: se seleccionó Sublime Text para el desarrollo del sistema. Esta elección se debe a que ya se lo había utilizado anteriormente para desarrollos de sistemas y tiene un rendimiento bastante aceptable. Cuenta con muchas características que son de mucha utilidad para los desarrolladores y funciona tanto para Windows, Linux como Mac OS.
 - Lenguaje de Programación: se seleccionó el lenguaje de programación Ruby para el desarrollo y codificación del sistema web. Los integrantes del proyecto tienen un conocimiento en dicho lenguaje. Para hacer más fácil y ameno el uso de este lenguaje, se utilizará el *framework* Ruby on Rails.
 - DBMS: se seleccionó a PostgreSQL como motor de bases de datos. El mismo fue elegido, ya que cuenta con una gran comunidad y manuales que sirven de ayuda a la hora del desarrollo de aplicaciones.
 - Sistema Operativo: mediante el uso de una máquina virtual, se seleccionó utilizar Ubuntu 14.04. Esta elección se debe a que al instalar dependencias y

otras librerías utilizando Windows, generaban muchos problemas a la hora de configurar el entorno de programación.

Arquitectura de procesamiento y comunicación de la propuesta tecnológica

En base a la arquitectura de procesamiento y comunicación de la propuesta tecnológica que se presenta, se tiene el siguiente esquema que se observa en la imagen 37.

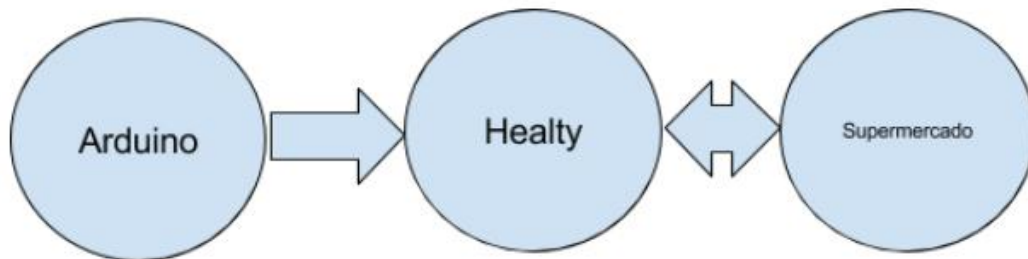


Imagen 37: Arquitectura de procesamiento y Comunicación.

En el gráfico se puede observar 3 partes o módulos dentro del sistema. El sentido de las flechas, representa el flujo de información y la comunicación entre los diferentes módulos.

- Arduino: El microcontrolador ATmega328P, es el que se encarga del procesamiento de toda la información dentro de Arduino UNO. A su vez, los diferentes módulos que se acoplan junto a Arduino UNO para conformar el sistema domótico, también realizan un procesamiento interno.

En temas de comunicación, el sistema domótico sería el Transmisor y el sistema web Healty sería el Receptor de la información. En este caso, el modo de transmisión sería Simplex, ya que la comunicación se realiza en un solo sentido sin poder invertir los roles entre ambas partes.

- Healty: El procesamiento se realiza mediante transacciones, en el cual se recolecta, almacena, modifica y recupera toda la información generada por las peticiones o transacciones realizadas en todos los sistemas que conforman este proyecto.

En aspectos de comunicación, como se dijo anteriormente, Healty se comporta como Receptor en su comunicación con el sistema Arduino. En tanto, el modo de transmisión entre Healty y el sistema web del Supermercado es Full-Duplex, ya que las transmisiones pueden realizarse en ambos sentidos al mismo tiempo y los sistemas comportarse tanto como Transmisor como Receptor.

- Supermercado: En este caso, al ser un sistema externo, no se sabe con exactitud si las peticiones se procesan mediante lotes o con transacciones. Se mencionó anteriormente que el sistema web del Supermercado se comporta tanto como Transmisor como Receptor.

5 - RESULTADOS O VERIFICACIÓN EXPERIMENTAL

Experimentos y pruebas realizadas

Durante el desarrollo del software y del hardware, se desarrollaron varias pruebas para verificar la veracidad y la funcionalidad del sistema propuesto. Las mismas tienen como objetivo encontrar errores, con el fin de solucionarlos y confirmar la calidad del producto final de este proyecto.

El fin del capítulo, es documentar las pruebas empíricas y técnicas que se desarrollaron, para proporcionar información objetiva sobre la calidad del producto. Las pruebas se fueron desarrollando a lo largo del proyecto, tanto en la parte de software como de hardware, por lo que, a fines de facilitar la comprensión del lector y la organización, se dividirá este capítulo en dos módulos, software y hardware. Los detalles a continuación:

Software

Las primeras pruebas que se desarrollaron en cuanto al software fueron dinámicas, ejecución del código por medio del navegador o de la consola de Linux. Las pruebas de la codificación se pueden dividir en los siguientes grupos:

Seguridad

- La seguridad se probó por dos métodos diferentes. En primer lugar fue necesario rever el código, buscando fallas, fue una verificación interna. La misma se desarrolló:
 - Verificando que los controladores permitan únicamente los parámetros que el usuario pueda modificar. Esto se realiza para evitar inyecciones HTML. Los controladores solo deben permitir parámetros que los usuarios puedan modificar libremente, ya que esto es muy manipulable desde el HTML del navegador, y así, un usuario podría acceder a ciertas columnas de la base de datos en las cuales no debería.

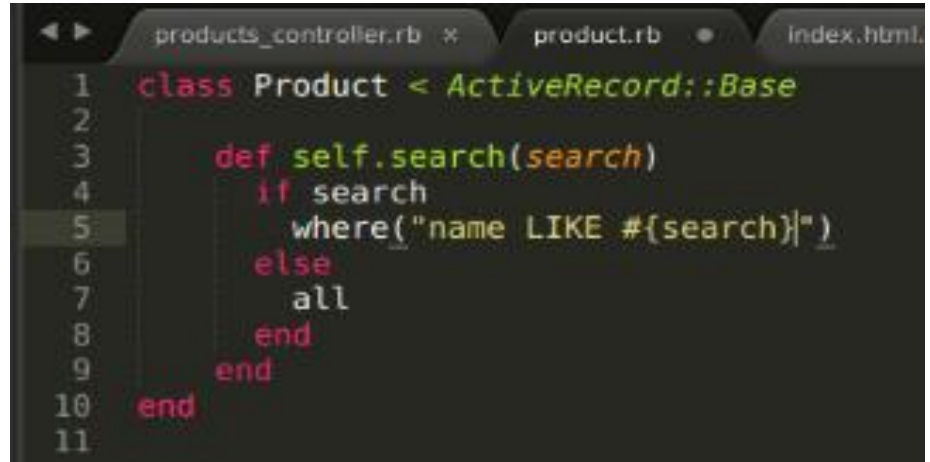


```
controller.rb - (test) - Sublime Text (UNREGISTERED)
182 end
183 end
184 end
185 # DELETE /purchaseorders/1
186 # DELETE /purchaseorders/1.json
187 def destroy
188   @purchaseorder.destroy
189   respond_to do |format|
190     format.html { redirect_to purchaseorders_url, notice: 'Purchaseorder was successfully destroyed.' }
191     format.json { head :no_content }
192   end
193 end
194
195 private
196 # Use callbacks to share common setup or constraints between actions.
197 def set_purchaseorder
198   @purchaseorder = Purchaseorder.find(params[:id])
199 end
200
201 # Never trust parameters from the scary internet, only allow the white list through.
202 def purchaseorder_params
203   params.require(:purchaseorder).permit(
204     :user_id, :total, :paid,
205     :active, :email, :phone, :neighborhood, :street,
206     :building, :department, :observation)
207 end
208
209 end
210
```

Imagen 38: Codificación - Seguridad

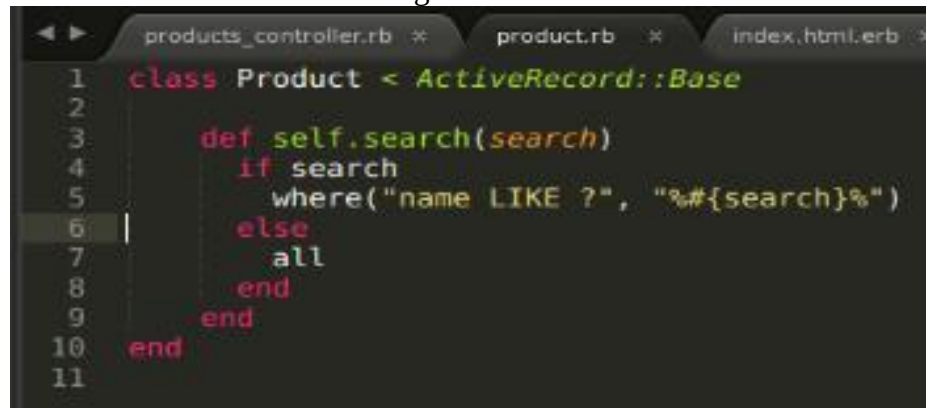
- Validar los modelos fue otro proceso necesario para garantizar la seguridad. En los mismos se realizaron validaciones de los atributos que se permitían en las vistas. Entre las validaciones podemos mencionar, a modo de ejemplo, la verificación de si un parámetro era numérico, si el parámetro estaba presente (!= nil), etc. Por medio de los modelos también se tuvo que tener en cuenta las inyecciones SQL en los parámetros. Se siguió los estándares propuestos por Ruby para

poder solucionar esta falla. Rails propone por medio de sus QUERY protección en contra de este tipo de inyecciones, dividiendo la consulta en dos parámetros. Por ejemplo, en vez de que la consulta quede estructurada como



```
1 class Product < ActiveRecord::Base
2
3   def self.search(search)
4     if search
5       where("name LIKE #{search}")
6     else
7       all
8     end
9   end
10 end
11
```

debería ser estructurada de la siguiente manera



```
1 class Product < ActiveRecord::Base
2
3   def self.search(search)
4     if search
5       where("name LIKE ?", "%#{search}%")
6     else
7       all
8     end
9   end
10 end
11
```

Imagen 39: Seguridad – Validación de Modelos

- Las validaciones en las **vistas** fueron las menos frecuentes. Las más comunes fueron evitar el uso de los hidden_fields (inputs ocultos), siempre que sea posible, para evitar la modificación de parámetros no deseados por parte del usuario, o también las validaciones de tipo de datos (integer, string, char, etc) en los inputs del HTML.
- Otro tipo de validaciones fueron las realizadas tanto en vistas, modelos y controladores. Las mismas hacen referencia a los tokens de los formularios, los tokens de las transacciones de MercadoPago, la encriptación y desencriptación de los atributos de los formularios, etc. Las encriptaciones de los datos, se realizaron para que los usuarios no puedan modificar el monto total de las órdenes de pago, ya que el monto se pasaba como parámetro a las vistas para ser enviado al supermercado y a MercadoPago por medio de get request. Para que esta encriptación sea efectiva, tuvo que ser realizada de la siguiente manera
 - Se encripta del lado del servidor de Healty. Se utilizó la encriptación Cipher, con la llave “EncryptDecryptGurudathBN”
 - Se envía a la vista del lado del cliente Healty, ya con el monto encriptado

- El usuario envía un *get request* con el parámetro del monto de la orden de compra encriptado.
- El lado del servidor del supermercado recibe el parámetro encriptado y lo desencripta con la misma *key* compartida. El resultado de la desencriptación, debe ser el monto de la orden de compra.

```
7 @purchaseorders = current_user.purchaseorders
8 end
9
10 # GET /purchaseorders/1
11 # GET /purchaseorders/1.json
12 def show
13   cipher = OpenSSL::Cipher.new('AES-128-ECB')
14   cipher.encrypt!
15   cipher.key = "EncryptDecryptGurudath8N"
16   @crypt = cipher.update(@purchaseorder.total.to_s) + cipher.final()
17   require 'json'
18   @po = @purchaseorder.to_json
19   @data = current_user.email
20   @data << ' ' + @purchaseorder.purchasedetails.count.to_s
21
22   @purchaseorder.purchasedetails.each_with_index do |pd,i|
23     @data << ' ' + pd.ingredient.name
24     @data << ' ' + pd.quantity.to_s
25   end
26 end
27 def add_detail
28 end
29 end
30
31 def pay
32   @purchaseorder = current_user.purchaseorders.find(params[:id])
33   payment_data = Purchaseorder.do_payment(params[:issuer],@purchaseorder.total,params[:token])
34   @response = $mp.post("/v1/payments", payment_data)
35   @errors = Purchaseorder.errors_response(response)
36   respond_to do |format|
37     if @response['response']['status'] == 'approved'
38       format.html { redirect to @purchaseorder, notice: @response }
39       format.json { render :show, status: :ok }
40     else
41       format.html { redirect to @purchaseorder, alert: @response }
42       format.json { render json: @purchaseorder.errors, status: :unprocessable_entity }
43     end
44   end
45 end
```

Imagen 40: Seguridad - Encriptación

- La verificación externa, se realizó básicamente intentando penetrar la seguridad del sitio web, especificada en el punto anterior, mediante ataques SQL, HTML, Sesiones, y modificación de parámetros.

Funcionalidad

- La verificación de funcionalidad se realizó del lado del cliente, haciendo pruebas de caja negra. Sobre este método podemos especificar, que se probaron valores sin tener en cuenta el funcionamiento interno, solo se ingresan valores de entrada y se verificó que los valores de salida sean los deseados. El objetivo de esta prueba fue encontrar:
 - Funciones incorrectas o ausentes.
 - Errores de interfaz.
 - Errores en estructuras de datos o en accesos a las Bases de Datos externas.
 - Errores de rendimiento.
- Se utilizaron dos técnicas de prueba de caja negra para verificar la funcionalidad del software:
 - Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.

Una partición equivalente, es una técnica de prueba de Caja Negra, que divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El diseño de estos casos de



prueba para la partición equivalente, se basa en la evaluación de las clases de equivalencia.

El diseño de casos de prueba para la partición equivalente, se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia, representa un conjunto de estados válidos o inválidos para condiciones de entrada.

Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica.

Las clases de equivalencia se definieron de acuerdo con las siguientes directrices: Si un parámetro de entrada debe estar comprendido en un cierto rango, aparecen 3 clases de equivalencia: por debajo, en y por encima del rango.

Si una entrada requiere un valor concreto, aparecen 3 clases de equivalencia: por debajo, en y por encima del rango. Si una entrada requiere un valor de entre los de un conjunto, aparecen 2 clases de equivalencia: en el conjunto o fuera de él. Si una entrada es booleana, hay 2 clases: si o no.

Los mismos criterios se aplican a las salidas esperadas: hay que intentar generar resultados en todas y cada una de las clases.

Aplicando estas directrices, se ejecutan casos de pruebas para cada elemento de datos del campo de entrada a desarrollar. Los casos se seleccionan de forma que ejerciten el mayor número de atributos de cada clase de equivalencia a la vez.

- Técnica del Análisis de Valores Límites: esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.

Consiste en ingresar valores que se encuentren por debajo y por encima de los valores aceptables por la clase de equivalencia. El sistema debe estar preparado para recibir valores incorrectos, y poder responder de manera correcta a los mismos.

Eficiencia

Para medir la eficiencia del sistema se verificó:

- Que el sistema sea totalmente amigable para el usuario común. Como uno de los objetivos principales del proyecto, se planteó que el producto resultante sea de uso muy intuitivo, siguiendo estándares de diseño y de transacciones. La eficiencia se logra al comprobar, que un usuario con conocimientos informáticos medios, pudo realizar la medición de su pH y la compra de los ingredientes de las dietas.



- Los tiempos de respuesta fue un punto a considerar, debido a los complejos procesos que se desarrollan, tanto en modelos como en controladores. Para realizar estas pruebas se cronometró el tiempo de acceso a las vistas, el tiempo de respuesta del servidor, etc.
- Se realizaron pruebas de cargas, que se utilizan para comprobar el comportamiento de la aplicación bajo una cantidad de peticiones esperadas. Si se monitorizan las bases de datos y/o el servidor de aplicaciones, se podría observar cual es el cuello de botella del sistema y mejorarlo.

Portabilidad

Para medir la portabilidad del sistema se verificó:

- El sistema fue diseñado y desarrollado mediante lo que se conoce como “*Responsive*” - adaptar las apariencias de las páginas web según el dispositivo con el cual se esté utilizando para visualizarla - para poder ser utilizado en diferentes plataformas como ser: computadoras convencionales, tablets, smartphones, etc. Esto se configura de una manera simple al utilizar el *framework* BOOTSTRAP, que es un conjunto de herramientas de software libre para el diseño de sitios y aplicaciones web.

Conectividad

Para medir la conectividad relacionada entre el Hardware y el Software:

- Se realizaron diferentes pruebas de conexión entre el sistema domótico y el sistema web, para enviar el valor del pH hacia el servidor que fue tomado por Arduino. Esta conexión se realizó mediante una red propia Wifi, enviando peticiones desde Arduino al Sistema Web.

Hardware

Entre las diferentes pruebas que se realizaron, en lo que es el hardware del sistema (Arduino y módulos asociados), fueron las siguientes:

Montaje: teniendo ideas de un diseño de cómo sería el soporte para mantener parte de los componentes que forman este sistema domótico, se tuvieron que realizar varios cambios en la estructura del mismo, para que cumpla con los requisitos necesarios para el correcto funcionamiento.

Entre las diferentes pruebas que se realizaron podemos encontrar:

- Material y forma del almacén: se probaron con diferentes materiales (metal, plástico). Este fue un punto importante, ya que este almacén no debería tener un gran peso y la forma tiene que ser con el mayor grado de detalle, para que quede visualmente aceptable para el usuario final.
A su vez, en el mismo se tendrían que colocar algunos de los componentes adicionales que forman parte del sistema domótico (Servo, Sensor de distancia, Sensor de colores, cinta transportadora). Esto también fue algo en lo que se realizaron diferentes tipos de pruebas por la ubicación de los mismos y tratando de encontrar la mejor posición.
- Cinta transportadora: se realizaron diferentes pruebas sobre el tamaño, material y ubicación de esta cinta transportadora. La misma debía ser de un elemento cuya base sea deslizable fácilmente, para que las tiras reactivas puedan recorrer el camino de una manera correcta, una vez que son tomadas por el rodillo que está colocado adicionalmente al Servo.

En relación al tamaño y la ubicación de esta cinta, se buscó uno acorde a las dimensiones que posee el armazón, y cuya ubicación sea conforme a un correcto funcionamiento del sistema.

- Rodillo: este rodillo es una de las partes importantes a la hora de tomar las tiras reactivas que ingresan al sistema, por lo que se buscaron diferentes alternativas que sean compatibles con el Servo. Se probaron diferentes rodillos que se consiguieron de diferentes componentes electrónicos que ya no eran utilizados, y serán de mucha utilidad para este diseño. Otro punto a tener en cuenta, fue la altura o distancia que se tenía entre la cinta transportadora y este rodillo, ya que si no se lo ubicaba de una manera correcta, no podría sujetar de manera correcta a las tiras reactivas.
- Sensor de distancias: este sensor tiene que ser colocado en una posición estratégica, para que cuando el paciente se aproxime a cierta distancia a orinar, sea detectado y el sistema domótico se active. Se realizaron muchas pruebas para asegurar donde debería ser colocado dicho sensor, y si era correcta la distancia a la cual se había programado el mismo.
- Sensor de colores: este sensor debe ser ubicado en un lugar donde sea eficiente a la hora de detectar el color que tiene la tira reactiva y que no tome otros colores. Estos otros colores hacen referencia a la luz del ambiente, como así también, a la luz artificial. Luego de probar diferentes posiciones, se determinó el lugar correcto donde deberá estar dicho componente.

Tiras reactivas: las tiras reactivas que fueron conseguidas en el mercado local, son las de uso convencional. Esto consiste en sumergir por completo la tira reactiva por un corto periodo de tiempo, en una muestra bien mezclada de orina. Luego se la extrae del recipiente esperando un tiempo para quitar excedentes de orina, y se espera un tiempo razonable hasta que se llevan a cabo las reacciones químicas.



Imagen 41: Tiras Reactivas

En nuestro caso, se realizaron diferentes pruebas con diferentes componentes químicos para que las tiras reactivas tomen valores diversos y comprobar si el sistema domótico mediante el sensor de colores enviaba la información correcta. A continuación en la imagen 42 se observa un listado relacionado con pH de diversos componentes.

ACIDO	0	HCl
	0,5	Ácido de batería
	1	Ácido sulfúrico
	2	Jugo gástrico
	2,4	Jugo de limón
	2,9	Vinagre
	3	Jugo de naranja
	5	Café
	5,6	Lluvia ácida
	5,5 - 6,5	Orina
	6,5	Leche
NEUTRO	7	Agua pura
	6,5 - 7,4	Saliva humana
	7,35 - 7,45	Sangre
	8	Agua de mar
	9 - 10	Jabón de manos
	11,5	Amoníaco
	12,5	Lejía
	13,5	Limpia tuberías
BASICO	14	NaOH

Imagen 42: pH de componentes

Se realizaron pruebas sumergiendo la tira reactiva en Lejía (lavandina), vinagre y en agua pura, para obtener distintos valores para ser probados por el sistema domótico.

Después de una serie de pruebas, lo que se propuso hacer fue armar tiras pintadas con un determinado color para que perduren en el tiempo, y no hacer un gasto excesivo de tiras reactivas.

Otro punto a tener en cuenta para realizar las pruebas, es que el tamaño de las tiras reactivas al ser genéricas, tienen un ancho y un largo determinado, por lo que eran muy pequeñas para que el sensor de colores detecte dicho valor.

Conectividad: respecto a los temas de conectividad entre el sistema domótico y Healty, se tuvieron que hacer muchas pruebas para comprobar que los datos que estaba tomando el sistema domótico, llegaba de una manera correcta al sistema Web. Estos datos luego se convierten en información, la cual es útil para darle un sentido a esta medición que se realiza.

Esta conexión que se establece entre ambos sistemas, se realiza mediante el ESP-8266 (Módulo Wifi) el cual se debe configurar de manera que se conecte a una red Wifi para lograr esta comunicación entre partes. Se realizaron diferentes pruebas con varias redes para lograr una configuración correcta del módulo. Dicha configuración se realiza mediante los comandos AT.

Para poder comprobar si se producía la conexión entre el sistema Arduino y el sistema web, lo que se realizó fue lo siguiente:

- Se tomaba un valor mediante el sensor de colores, y se comprobaba mediante el monitor que presenta el IDE de Arduino, si dicha información era correcta.



- Mediante la configuración de un script en el código programado, se realizaba la conexión con el sistema web y se enviaban los datos.
- Mediante la consola en Ubuntu, se podía observar el estado de la petición (esto era un parámetro para saber si se había enviado dicha información), y en el caso de que la petición sea correcta, se devuelve una respuesta estándar “200 - OK” y cuando no se podía realizar esta conexión se devuelve un error “400 - Bad Request”.

Codificación Arduino: a la hora de realizar la programación del sistema Arduino, se dividieron las pruebas en dos etapas, las cuales se detallan a continuación:

- Primer Etapa: se realizaron pruebas de los diferentes componentes de Arduino de manera separada para ver si funcionaban de una manera correcta y respondían a las peticiones que se realizaban. Estas pruebas se hicieron siguiendo los manuales de configuración de los diferentes componentes, y a través del IDE de programación probar los diferentes códigos.
- Segunda Etapa: una vez que ya se probaron todos los componentes de forma unitaria, se buscó realizar pruebas de integración entre las partes, para comprobar que todo funcionaba de manera correcta. Un aspecto a tener en cuenta a la hora de realizar la integración, es que el código escrito de forma unitaria iba a sufrir muchas modificaciones, al tener que cumplir con todos los requerimientos necesarios según las especificaciones del producto final y su funcionamiento.

Al hablar de requerimientos necesarios, nos referimos entre ellos:

- Que el sensor de ultrasonidos tome la distancia correcta entre el paciente y el sistema Arduino para que dé inicio al proceso de medición de pH.
- Que el Servo realice su giro durante cierto tiempo, para poder colocar la tira reactiva y que el paciente pueda orinar; luego regrese la misma a que el sensor de color pueda detectar dicho valor y se realice el envío de información al sistema Web. A su vez, se realizó una mejora a la hora de la colocación de la tira reactiva, ya que, si la misma no se encontraba bien ubicada, el servo puede realizar movimientos en ambos lados para colocarla de manera correcta y así tomar dicho color.
- El envío de información mediante el ESP-8266 se realiza de manera automática, una vez que el sensor de colores tomó el valor de la tira reactiva.
- Una vez que se tomó dicha información, el servo debería girar para tirar dicha tira reactiva al inodoro y dejar el sistema listo para una próxima medición.

Se realizó esta división en etapas, porque como sabemos, no es lo mismo probar componentes de manera unitaria que realizando una integración entre diferentes partes, y que todo funcione de la misma forma.



Resultados obtenidos

Software

Realizando las pruebas detalladas en el punto anterior se pudieron observar los siguientes resultados.

- No se pudo realizar ninguna inyección HTML, los parámetros configurados en los controladores fueron robustos en todas las pruebas. El único controlador que no se probó fue el de usuario, debido a que el mismo fue realizado por medio de una gema basada en Warden. La seguridad de la misma ya se encuentra comprobada por medio de numerosas pruebas que se pueden encontrar en la red. La autenticación (*cookies* de sesión) de usuarios fue requerida siempre y cuando se necesite modificar un campo de la base de datos.
- Los modelos fueron validados con éxito. Solo se necesitó realizar modificaciones para permitir relaciones entre las tablas de producto e ingredientes, debido a la presencia de un *nested form* (formularios anidados). En cuanto a seguridad, podemos afirmar que el resultado de validar los modelos, fue un proceso que permitió conocer más a fondo la interacción del *framework* con el lenguaje de programación. Se pudo observar también, la versatilidad del *framework* en cuanto a la validación de atributos de los objetos. El único inconveniente que se presentó a la hora de la validación, fue la dificultad de verificar la presencia de un parámetro A basándose en el contenido de un parámetro B. Se buscó documentación en la guía oficial de Ruby y no se encontró ninguna explicación, como tampoco se encontró en *Stackoverflow*, foro principal de ayuda para programación en este proyecto.
- En cuanto a la validación de las vistas, pudimos observar, que es la parte sensible a falla del sistema. El principal defecto es la falta del SSL para las transacciones por MercadoPago y en general. Al no contar con el certificado SSL, las *request* podían ser interceptadas y leídas con facilidad, observando datos totalmente sensibles. En cuanto a la encriptación realizada, tanto en vistas como en controladores, es segura hasta determinado punto, ya que con solo conocer la llave del encriptador se puede modificar el parámetro “total” de la orden de compra. También se observó que sin conocer la llave, copiando el valor encriptado de otra orden de compra con un total menor, se podría replicar en el formulario actual y ser enviada, sin ninguna detección de error, por parte del sistema.

Funcionalidad

- Las verificaciones que se realizaron durante la codificación del sistema, no se documentaron por razones obvias. Las verificaciones que se tratarán en esta sección, solo hacen referencia a las pruebas realizadas una vez finalizada la codificación de los módulos.
 - Funciones incorrectas o ausentes:
 - Cuando se realizó esta prueba, se pudo determinar que la encriptación de los datos sensibles en los modelos de órdenes de



compras, no estaba presente. Fue cuando se implementó el sistema de seguridad descrito en este proyecto.

- En la fase de desarrollo se utilizó la gema “*paperclip*” para realizar la carga de imágenes y luego relacionarlas con los productos. Una vez que se implementó el sistema en Heroku (pasando del ambiente de desarrollo a producción), pudimos observar que el servidor no permite la carga de imágenes del lado del cliente, por razones de seguridad. Por este motivo se tuvo que eliminar la gema en uso, y seleccionar una nueva llamada “S3_amazon”. Servicio de alojamiento de imágenes en la nube.
- Otra función incorrecta que se pudo observar es que al eliminar productos, los detalles en las órdenes de compras se eliminaban. Este error se tuvo que solucionar para poder llevar un correcto registro de la facturación y para conservar la integridad de los datos.
- Al probar el módulo de dietas con un pH neutro (=7), el sistema no mostraba ninguna dieta.

Errores de interfaz:

- El diseño *responsive* no funcionaba para determinadas resoluciones. Generalmente ocurre este error con resoluciones comunes en monitores antiguos.
- Existían vistas a las cuales se podía acceder, que no contenían información, o contenían datos generados por el lenguaje por medio de scaffolds.
- Se encontraron vistas donde no se encontraban disponibles los botones básicos de la misma, entiéndase paginación, botones de redirección a visitas anteriores, a pasos siguientes, etc.
- La forma de mostrar las fechas seguía el formato de PostgreSQL, por lo que se tuvo que realizar una internalización de datos al formato español.
- Las vistas, que trabajaban con JavaScript, debían ser recargadas para poder hacer un uso correcto del lenguaje. Esto se debía a la presencia de turbolinks.

Errores en estructuras de datos o en accesos a las Bases de Datos externas.

- No se encontraron errores en esta sección, pero se pueden desarrollar mejoras a futuro en caso de que la concurrencia al sistema sea más requerida.

Errores de rendimiento.

- No se encontraron errores de rendimiento que puedan afectar de manera significativa al sistema, pero si se notó que existían procesos que necesitaban más tiempo del esperado.
- El servidor otorgado por Heroku, posee un 95% de disponibilidad. Esto quiere decir que no siempre que uno desee hacer uso del sistema, el mismo estará disponible. Esto es causa del plan contratado.

Eficiencia

Para medir la eficiencia del sistema se verificó:



- El registro del procedimiento se realizó con un total de 5 usuarios sin una previa capacitación. La prueba se realizó solamente indicando al usuario tester los objetivos del proyecto, se le otorgaron los componentes necesarios (sistema domótico y sistema web), y se verificó que el sistema sea totalmente intuitivo. Afirmamos que el proceso fue exitoso, ya que sin ningún inconveniente, los usuarios pudieron lograr los objetivos previstos.
- En cuanto al rendimiento del servidor, se obtuvieron buenos resultados debido a la calidad de servicio que ofrece Heroku, aunque no siempre fueron satisfactorios. Una de las debilidades que presentó fue que con el plan elegido del servidor web no llegó a optimizar los tiempos de respuesta de consultas, ya que no se tiene un gran procesamiento ni memoria. Otro punto a considerar es que el servidor solamente se mantiene despierto 8 horas al día con el plan contratado. Una vez que se ingresa al servidor, este se despierta y permanece en este estado hasta media hora después de la última consulta realizado al mismo. Si el servidor permanece 8 horas despierto, este pasa a estado de dormido por un día.

Portabilidad

- Para obtener los resultados de la portabilidad de la interfaz en distintas resoluciones, se utilizó el inspector de elementos de Mozilla Firefox. El mismo permite simular las diferentes resoluciones. El resultado fue el esperado en la mayoría de las pruebas. Para resoluciones utilizadas por monitores antiguos, el sistema presentaba un desfase en determinados módulos.

Conectividad

- Para medir el rendimiento del sistema, se utilizó un complemento de Ubuntu llamado SIEGE. Se puede observar en la imagen 43 los resultados de la prueba.

```
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Configurando siege (3.0.5-1) ...
sa@ubuntu:~$ siege --concurrent=50 --reps=100 http://localhost:3000
** SIEGE 3.0.5
** Preparing 50 concurrent users for battle.
The server is now under siege...^C
Lifting the server siege...      done.

Transactions:      814 hits
Availability:      100.00 %
Elapsed time:      169.42 secs
Data transferred:  9.88 MB
Response time:     9.52 secs
Transaction rate:  4.80 trans/sec
Throughput:        0.06 MB/sec
Concurrency:       45.75
Successful transactions: 814
Failed transactions: 0
Longest transaction: 23.59
Shortest transaction: 0.16

FILE: /var/log/siege.log
You can disable this annoying message by editing
the .siegerc file in your home directory; change
```

Imagen 43: Rendimiento en Conectividad

Hardware

Los resultados que se obtuvieron por parte de la implementación del sistema domótico fueron los siguientes:

Montaje: a la hora elegir el material con el cual se hizo el armazón del sistema, se terminó seleccionando a una base metálica que no posee mucho peso, y un tamaño acorde para brindar las prestaciones del sistema de una manera correcta.



La cinta transportadora brindó excelentes resultados con el material elegido (plástico). Este componente permite un deslizamiento efectivo, sin inconvenientes para las tiras reactivas y hace el proceso mucho más fácil.

El rodillo seleccionado está compuesto por goma, y cubre todo el ancho de la cinta transportadora. A su vez, mediante una parte adicional que se coloca en el servo, se lo pudo mantener fijo para que su giro sea regular y constante, permitiendo no tener problemas con la toma y deslizamiento de las tiras reactivas.

En cuanto a las ubicaciones de los sensores de distancia y colores, se obtuvieron los resultados esperados. El sensor de distancia tomó correctamente los valores según la proximidad de la persona que se acercaba al sistema domótico, y el sensor de colores detectó los colores de la tira reactiva de manera correcta.

Tiras Reactivas: los resultados de las tiras reactivas fueron los esperados según las tablas de pH y colores que deberían adquirir las mismas, al estar en contacto con algún químico o sustancia particular.

Conectividad: en relación a la conectividad entre el sistema domótico y el sistema web Healty se obtuvieron los resultados esperados. Se pudieron enviar correctamente los datos y lograr una comunicación efectiva entre ambos sistemas.

Codificación Arduino: los resultados obtenidos de las pruebas unitarias de los diferentes componentes que forman el sistema domótico, nos brindaron resultados esperados.

A la hora de realizar la integración de las diferentes partes, luego de muchas pruebas, se pudo llegar a los resultados esperados y que el proceso de funcionamiento del dispositivo sea el correcto.

Calidad de los resultados en comparación con otras soluciones similares

La calidad de los resultados esperados fue buena y aceptable para los requerimientos que tiene este proyecto, tanto en hardware como en software. Lo que hay que tener en cuenta, es que no se conocen otras soluciones similares que realicen la misma funcionalidad. En base a esto no se tienen métricas o variables en las cuales se pueda realizar una comparación entre diferentes sistemas.

Evaluación general de procedimientos, recursos y facilidades

En los procedimientos que se utilizaron para la realización del presente proyecto podemos destacar los siguientes aciertos, errores y debilidades:

Proyecto

- La selección del tema en el que basamos nuestro proyecto de grado es considerado un acierto, ya que pudimos encontrar basta información en internet. Dicha información no era de difícil comprensión, aun no teniendo los conocimientos básicos de medicina, lo cual facilitó en gran medida el desarrollo del sistema.



- Otro punto a considerar con respecto al tema seleccionado, es que podemos diseñar, idear un prototipo el cual permite mejorar la vida de las personas, lo cual fue muy motivador durante todo el proyecto.
- La previa planificación de las metodologías a utilizar fue un acierto, que nos permitió encarar un proyecto organizado y documentado.
- La previa investigación de la calidad de los productos utilizados, y de los proveedores seleccionados, es considerado como un punto a favor del proyecto, ya que gracias a esta investigación, se pudo concretar un proyecto sin realizar ninguna devolución, ya que todos los productos contaban con la calidad deseada.
- El control y seguimiento de todo lo que se planificó, haciendo comparaciones con las líneas bases (alcance, costos, RR.HH., adquisiciones) no tuvo grandes desviaciones, por lo que se podría considerar como un acierto en el proyecto.
- Definir un alcance coherente en el proyecto, fue uno de los motivos por el cual pudimos concretar este trabajo. Los criterios de aceptación fueron totalmente razonables y alcanzables para un proyecto desarrollado por dos personas.
- Una de las fallas que podemos mencionar, es la que se presentó a la hora de desarrollar un proyecto con un EDT de alto nivel. Consideramos que haber desarrollado más este punto, podría haber ayudado de manera considerable a la organización de los procesos del proyecto.
- La falta de investigación a la hora de la selección de uno de los productos (Motor Shield Arduino) derivó en un gasto innecesario para el proyecto, ya que el mismo no se utilizó.
- Otra falla que podemos mencionar, es que durante un periodo no mayor a 2 meses, no se respetaron las daily meeting, lo cual produjo un retraso en el cronograma, ya que tuvimos que rever los procesos realizados durante este tiempo.

Producto

- En cuanto a aciertos del producto podemos mencionar la utilidad que presenta el mismo. Se pudo obtener un sistema totalmente amigable para el usuario, lo cual cumple con los criterios de aceptación previamente planteados.
- La selección del lenguaje también fue un punto a favor del proyecto, ya que el mismo presenta documentación de fácil acceso y entendimiento, sumado a la gran cantidad de librerías disponibles para la realización del sistema web. En cuanto al lenguaje de Arduino no tuvimos más opción que utilizar el lenguaje predefinido del hardware, ya que no permite otro tipo de interacción. Aun así, podemos decir que Arduino (lenguaje) fue de simple comprensión, y encontramos gran cantidad de proyectos en internet que facilitaron la realización de nuestros requerimientos.
- La selección y utilización del *framework* de CSS Bootstrap fue un punto a favor, que permitió agilizar el desarrollo del diseño de la página web del producto.
- La selección del hardware para armar el sistema domótico, fue un acierto en cuanto a la facilidad de adquisición del mismo en nuestro país. Si bien las compras del mismo se realizaron a proveedores de Buenos Aires por medio de internet, pero no fue un impedimento a la hora de realizar el producto.
- Una debilidad que se encuentra en el sistema, es que el mismo está pensado para ser utilizado por personas de sexo masculino. Esto podría mejorarse



planteando una reforma en la solución para el uso de personas de sexo femenino.

- Una debilidad dentro del sistema web, es la falta de documentación del código fuente. Esto debería ser una buena práctica a la hora del desarrollo de un sistema para ayudar a explicar qué hace cada función, y a su vez si ocurriese un error encontrarlo más rápido dentro del código.
- Otra debilidad que se encuentra dentro del sistema, es la seguridad con respecto a los envíos de datos de datos entre sistemas. Por falta de tiempo, no se pudo aplicar de una mejor forma la seguridad en dicho producto.
- Una falla del producto domótico es la conexión a una red de internet. La misma se encuentra “*Hardcodeada*” con los datos de la conexión. Esto es una mala práctica y considerada un anti-patrón de diseño.

Dificultades no previstas encontradas y sus soluciones

Software

Entre las dificultades que no fueron previstas y se presentaron en el desarrollo del sistema web podemos mencionar las siguientes:

- **Problema:** JavaScript no funciona cuando la vista es redirigida por el sistema. Esto quiere decir que si, por ejemplo, previamente ingresamos a una orden de compras, la cual tiene en su vista código de JavaScript, y volvemos a ingresar luego por algún motivo, la gema turbolinks produce una interferencia con la codificación
- **Solución:** Investigando sobre la gema de turbolinks, se pudo observar que la misma funciona evitando que las páginas recarguen su contenido cuando las consultas ya fueron realizadas. Esta gema se crea por defecto al crear una aplicación en Ruby on Rails. La solución fue eliminar este comportamiento, quitando la gema del Gemfile y de application.js, para así evitar la interferencia no deseada. Como observación podemos decir que esto también afectó al rendimiento del sistema, ya que la gema borrada producía una mejora en cuanto a las solicitudes que se realizan del lado del cliente.
- **Problema:** A la hora de enviar el monto total de una orden de compra desde el sistema Healty hacia el Supermercado, esta información podría ser modificada mediante código HTML.
- **Solución:** La solución a dicho problema, fue implementar el proceso de encriptación, utilizando la encriptación CIPHER con la llave “EncryptDecryptGurudathBN”. Esto aumentó la seguridad del sistema, pero no fue una solución 100% efectiva ya que con conocer la llave, se podría realizar transacciones maliciosas o con simplemente copiar la encriptación de una orden de compra que posee un monto total menor, y enviarla, por medio de una modificación al código HTML, en una orden de compra de un monto total mayor.
- **Problema:** Incompatibilidad de Heroku con la base de datos (MySQL) inicial del sistema. Por defecto, al crear una aplicación nueva en RoR y no especificar ninguna base de datos, por defecto se crea asociándose a MySQL. Para cambiar la misma hay que especificar con un comando específico.
- **Solución:** Se tuvo que migrar la base de datos hacia otro DBMS PostgreSQL que era la base de datos permitida por Heroku.



- **Problema:** Heroku no permite la carga de imágenes del lado del cliente.
- **Solución:** Se tuvo que utilizar un sistema de almacenamiento de imágenes externo llamado “Amazon Web Service S3 Edge”. Se tuvo que migrar desde la gema Paperclip a las gemas Fog y S3_direct_upload. Se tuvo que crear una cuenta en Amazon para poder hacer uso de este servicio.

Hardware

Entre las dificultades que no fueron previstas y se presentaron en lo que es el armado del sistema domótico, nos podemos referir a las siguientes:

- **Montaje:** Uno de los primeros inconvenientes a la hora de realizar el montaje del sistema domótico, fue el no pensar que la persona cuando iniciaba el proceso para la toma del pH, ya no debería intervenir en el mismo a la hora de tirar la tira reactiva. La solución a esto fue mejorar el código desarrollado para que después de cierto tiempo, que en este caso es hasta que se hace el envío de la información al sistema web Healty, para que la tira reactiva sea expulsada de manera automática por parte del sistema domótico.

Otro problema surgido fue a la hora de colocar el servo en el armazón del sistema domótico, por el tamaño y la forma que posee el mismo. La solución a esto fue modificar el armazón que se había seleccionado, para que cumpla con las expectativas iniciales.

La organización del cableado, fue otro problema que surgió a la hora de realizar todas las conexiones de los componentes que forman parte del sistema domótico. No se había pensado en que habría tantos ensambles y esto perjudicó a la hora de dejar de manera presentable el prototipo final del sistema. La solución a esto fue organizar las posiciones de los diferentes componentes y sus conexiones a los diferentes pines de la placa Arduino Uno de manera que no sea obstrusivo para el correcto uso del dispositivo y que tenga una estética aceptable.

- **Tiras reactivas:** Como las tiras reactivas conseguidas no son de alta calidad, después de un determinado tiempo el color que era adquirido por las mismas iba cambiando por su contacto con el aire, así que las pruebas deben realizarse de manera instantánea. Esto se debe a que las tiras reactivas que fueron adquiridas, no están siendo usadas como corresponde. Esto es un problema y la solución que se planteó, fue realizar para las pruebas en el sistema unas tiras pintadas con diferentes colores, para que el sensor de colores las detecte y poder hacer el uso del sistema de una manera correcta.

Otro inconveniente fue el tamaño que poseían las tiras reactivas adquiridas, ya que eran muy pequeñas para la función que se las necesitaba. Por esto, la solución que se planteó, fue armar diferentes tiras con un tamaño mayor para no tener inconvenientes a la hora de que el rodillo las tome y se pueda completar el proceso de forma correcta.

- **Conectividad:** Este fue uno de los puntos donde más inconvenientes se tuvo. En primer lugar, se tuvo problemas en buscar manuales para la configuración del componente para la conexión mediante Wifi (ESP-8266). Otro problema que surgió fue que a la hora de tener todos los parámetros correctos para realizar la conexión con



el sistema web no se podía efectuar, y por último un error del módulo ESP-8266 debido a la falta de tensión y corriente necesaria para un correcto funcionamiento de dicho componente.

Las soluciones fueron las siguientes: además de buscar mucha documentación sobre ESP-8266, se tuvo que consultar con otras personas que estaban capacitadas en la tecnología Arduino, para que nos puedan ayudar a resolver dicho inconveniente según sus experiencias.

En cuanto al error de la conexión entre los diferentes sistemas, después de muchas pruebas y búsquedas en internet sobre este error, llegamos a la conclusión que el *firmware* que venía instalado en el módulo ESP-8266 ya era obsoleto y esto es lo que provocaba dicho dilema. La solución fue buscar descargar el nuevo *firmware* y un manual en algún foro o página especializada, en dónde nos ayuden a instalarlo. Luego de esto, se pudo realizar la conexión de una forma correcta y las peticiones llegaban de manera precisa.

Por la falta de tensión y corriente que nos brindaban los pines de salida del Arduino UNO, se tuvo que realizar la adquisición de una fuente de alimentación externa para protoboard, la cual nos ofrece los valores de tensión y corriente necesarios para que el módulo ESP-8266 funcione de manera correcta.

- **Codificación Arduino:** Uno de los problemas que ocurrieron a la hora de realizar la codificación para el funcionamiento del sistema domótico, fue la programación del giro del Servo. La solución a dicho inconveniente, fue realizar excesivas pruebas a la hora de definir la cantidad de grados que debería girar el servo para cumplir la funcionalidad.

Otro problema fue que a la hora de configurar el ESP-8266 mediante los comandos AT, se produjo un error que provocó una falla interna tanto en Arduino Uno como en el módulo cuyo efecto fue el no permitir ingresar comandos como así también salir de la pantalla. Esto se pudo solucionar formateando ambos componentes dejándolos en modo de fábrica.



6 - CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

Como conclusión del presente proyecto, podemos destacar que se cumplieron con los estándares esperados. Se llegó a desarrollar un sistema de complejidad elevada, gracias a la continua investigación y al constante desarrollo de los participantes del mismo. Podemos asegurar la funcionalidad del producto resultante y la importancia e impacto que puede tener en nuestra sociedad una vez perfeccionado. En este proyecto se desarrolló la base de un sistema que puede ayudar a las personas a tomar conciencia sobre su salud. Sin tener conocimientos informáticos fuera del estándar, se puede hacer uso de este sistema, lo cual destacamos como un punto a favor del mismo.

En cuanto a las pruebas relacionadas al producto final de este trabajo, podemos mencionar que fue un punto que no pudimos desarrollar por limitaciones de tiempo. Tuvimos que considerar que las pruebas del producto final (sin hacer referencia a las pruebas del hardware y del software) son realizadas por el uso diario del sistema por parte del usuario, lo cual requiere un tiempo prolongado para verificar si el correcto uso del sistema lleva a mejorar la salud del usuario según lo esperado. Cabe destacar también, que cada persona tiene diferentes reacciones a los distintos tipos de alimentación, por lo cual se debería considerar un estándar en cuanto a las pruebas del sistema domótico.

Como futuras líneas de investigación, se podría buscar el diseñar e implementar un sistema adaptable con una inversión de dinero razonable. Esto hace referencia a mejorar todo lo que es el armado del sistema domótico, como así también las prestaciones del mismo, y que pueda ser utilizado por personas del sexo femenino.

Si bien no podemos asegurar que el uso de un producto perfeccionado sea capaz de evitar enfermedades a quien lo use, podemos afirmar que es la base para futuras líneas de investigación y perfeccionamiento en cuanto a la salud de las personas. Este proyecto tuvo como base investigaciones científicas realizadas por Otto Heinrich, lo cual agrega cierto grado de confianza a lo realizado en este proyecto, ya que desde la parte informática cumplimos con una solución tecnológica a lo propuesto por el Dr. Heinrich.



7 - BIBLIOGRAFÍA

- Arduino. *Arduino*. <https://www.arduino.cc/en/Main/ArduinoBoardUno>
Página vigente al 21/01/2016
- Dory Martínez. *Mi herbolario*. <http://www.miherbolario.com/articulos/salud/64/estres-pandemia-de-actualidad>
Página vigente al 11/01/2016
- Eco Vida. *Eco Vida*. <http://ecovida.fundacioncodigos.org/alimentos-alcalinos-y-acidos-los-efectos-en-el-equilibrio-del-ph-en-el-cuerpo-humano/>
Página vigente al 11/01/2016
- EFD. *EFD*. <https://elfeniciodigital.wordpress.com/2013/11/23/la-verdad-indeseable-de-la-quimioterapia-en-el-cancer/>
Página vigente al 18/01/2016
- Hector Torres. *HETPRO*. <http://hetpro-store.com/TUTORIALES/arduino-vs-microcontrolador/>
Página vigente al 12/01/2016
- Juana Villalva. *Historias y Biografías*. <http://historiaybiografias.com/cancer/>
Página vigente al 18/01/2016
- Luis Fernando Atilano Martínez. *Scribd*. <http://es.scribd.com/doc/286199284/Contaminacion-ambiental#scribd>
Página vigente al 11/01/2016
- Matsumoto, Y. “. (s.f.).
- Susana Rodriguez. Lista de Alimentos Alcalinos, Acidos y Neutros.
<http://curacancernatural.org/alimentos-alcalinos/>
Página vigente al 23/02/2016
- OMS. *Organizacion Mundial de la Salud*.
<http://www.who.int/mediacentre/factsheets/fs297/es/>
Página vigente al 23/01/2016
- OMS. *Organizacion Mundial de la Salud*.
<http://www.who.int/features/qa/15/es/>
Página vigente al 11/01/2016
- OMS. *Organizacion Mundial de la Salud*.
<http://www.who.int/mediacentre/news/releases/2015/cancer-red-meat/es/>
Página vigente al 11/01/2016
- PROMETEC. *PROMETEC*. <http://www.prometec.net/servos/>
Página vigente al 21/01/2016
- Warburg, O. H. (1931). *El Metabolismo de los tumores*.
- Matriz FODA. Matriz FODA. <http://www.matrizfoda.com/>
Página vigente al 21/05/2016
- Honorarios de Informática. Honorarios Informática. <http://www.copaipa.org.ar/informatica/>
Página vigente al 22/05/2016
- Organización Mundial de la Salud (WHO).
- Martel C, Ferlay J, Franceschi S, et al. Global burden of cancers attributable to infections in 2008: a review and synthetic analysis. *The Lancet Oncology*.
- Informe mundial sobre el cáncer 2014, IARC
- Cura Cancer Natural. <http://curacancernatural.org/alimentos-alcalinos/>
Página vigente al 23/02/2016



- Salud. Enciclopedia de Salud, Dietética y Psicología
José L. Fresquet. Instituto de Historia de la Ciencia y Documentación (Universidad de Valencia - CSIC). Septiembre de 2007.
- Ruby on Rails. <https://www.ruby-lang.org/es/>
Página vigente al 21/01/2016
- Domótica, el futuro de las viviendas. <https://domoticaudem.wordpress.com/componentes-de-un-sistema/>
Página vigente al 21/01/2016
- Patagoniatec. <http://saber.patagoniatec.com/>
Página vigente al 25/01/2016
- Tutorial ESP8266 Parte I. http://www.naylampmechatronics.com/blog/21_Tutorial-ESP8266-Parte-I.html
Página vigente al 25/01/2016
- Arduino – Módulo Ultrasónico. <https://labyteacora.wordpress.com/2015/07/31/tutorial-arduino-módulo-de-ultrasonidos-hc-sr04/>
Página vigente al 26/01/2016
- Arduino – Sensor de distancia. <http://electronilab.co/tienda/sensor-de-distancia-de-ultrasonido-hc-sr04/>
Página vigente al 26/01/2016
- Arduino – Sensor de color. <http://www.sigmaelectronica.net/gy-31-p-2137.html>
Página vigente al 26/01/2016
- Arduino – Servos. <http://www.prometec.net/servos/>
Página vigente al 26/01/2016
- Heroku. <https://www.heroku.com>
Página vigente al 02/02/2016
- Project Management Institute, Inc. 2008. Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK) – Cuarta Edición. ISBN: 978-1-933890-72-2.
- ANMAT. http://www.anmat.gov.ar/webanmat/institucional/que_es_la_ANMAT.asp
Página vigente al 29/05/2016
- Flujo de Fondos. <http://www.buenosnegocios.com/notas/247-van-y-tir-cuanto-vale-una-inversion>.
Página vigente al 24/06/2016
- Mg. Bazzocco, Javier. 2014. Gestión Ágil de Proyectos de Software. Escuela de Administración Pública. Salta – Argentina.
- Porter, Michael E. 2009. Ser competitivo. Grupo Planeta (GBS). ISBN: 8423426955.
- Pressman, Roger S. 2005. Ingeniería del Software – Un enfoque Práctico. McGraw Hill / Interamericana de México. ISBN: 9789701054734.
- Romero Morales, Vázquez Serrano, Castro Lozano, R. 2007. Domótica e Inmótica Viviendas y Edificios Inteligentes. Editorial Alfaomega. ISBN: 978-970-15-1248-7.
- Sáez Vacas, F, Domínguez Hugo M. 2006. Domótica: Un enfoque sociotécnico. Fundación Rogelio Segovia. Madrid. ISBN: 84-7402-335-1.
- Evans, Brian W. 2008. Arduino Programming Notebook. ASIN: B002AD5Q8Q
- Chávez Cárdenas, Mario Alberto. 2015. Aprendiendo Ruby on Rails. México.



8 - ANEXOS

Domótica

La domótica es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía, que aporta seguridad y confort, además de comunicación entre el usuario y el sistema.

Un sistema domótico, es capaz de recoger información proveniente de sensores o entradas, procesarla y emitir órdenes a actuadores o salidas. El sistema puede acceder a redes exteriores de comunicación o información.

La domótica permite dar respuesta a los requerimientos que plantean estos cambios sociales y las nuevas tendencias de nuestra forma de vida, facilitando el diseño de casas y hogares más humanos, más personales, poli funcionales y flexibles.

El sector de la domótica ha evolucionado considerablemente en los últimos años, y en la actualidad ofrece una oferta más consolidada. Hoy en día, la domótica aporta soluciones dirigidas a todo tipo de viviendas, incluidas las construcciones de vivienda oficial protegida. Además, se ofrecen más funcionalidades por menos dinero, más variedad de producto, que gracias a la evolución tecnológica, son más fáciles de usar y de instalar. En definitiva, la oferta es mejor y de mayor calidad, y su utilización es ahora más intuitiva y perfectamente manejable por cualquier usuario. Paralelamente, los instaladores de domótica han incrementado su nivel de formación, y los modelos de implantación se han perfeccionado. Asimismo, los servicios posventa garantizan el perfecto mantenimiento de todos los sistemas. En definitiva, la domótica de hoy contribuye a aumentar la calidad de vida, hace más versátil la distribución de la casa, cambia las condiciones ambientales creando diferentes escenas predefinidas, y consigue que la vivienda sea más funcional al permitir desarrollar facetas domésticas, profesionales, y de ocio bajo un mismo techo.

La red de control del sistema domótico se integra con la red de energía eléctrica, y se coordina con el resto de redes con las que tenga relación: telefonía, televisión, y tecnologías de la información, cumpliendo con las reglas de instalación aplicables a cada una de ellas. Las distintas redes coexisten en la instalación de una vivienda o edificio. La instalación interior eléctrica y la red de control del sistema domótico, están reguladas por el Reglamento Electrotécnico para Baja Tensión (REBT). En particular, la red de control del sistema domótico, está regulada por la instrucción ITC-BT-51 Instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad para viviendas y edificios.

Beneficios de la domótica

Como principal característica de este tipo de sistemas, podemos nombrar el aporte que hace sobre la calidad de vida del usuario. Actualmente la domótica, en su mayoría, sirve para satisfacer el fin antes mencionado. Permite facilitar tareas diarias, trabajos engorrosos, y hasta suplantar la mano del hombre en determinadas situaciones.

Entre los rubros más sobresalientes de la domótica, podemos mencionar el campo de la ayuda a personas con diferentes tipos de discapacidad. Si buscamos en la web, podemos encontrar innumerables ejemplos de la ayuda que provee para personas con falta de

movilidad, discapacidad mental, falta de miembros del cuerpo, etc. Esta tecnología, permite integrar a la vida diaria a las personas que, por distintos motivos, no tienen las mismas capacidades que otras. Desde el traslado automatizado por el hogar a personas que no poseen movilidad en sus piernas de manera teledirigida, hasta alertas y recordatorios a personas que tienen problemas de memoria.

Otro rubro que podemos mencionar en los campos de la domótica, es el del ahorro energético. Esta tecnología está adaptada al hogar para iluminación, climatización, agua caliente sanitaria, riego, electrodomésticos, etc. Esto permite aprovechar de manera eficiente el consumo de recursos diariamente, reduciendo así los costos en comparación a una casa común, permitiendo obtener estadísticas de consumo, para luego optimizarlas modificando hábitos.

La seguridad es un tema que se presenta frecuentemente en aplicaciones de domótica. Permite aportar una mayor seguridad tanto al hogar como a las personas que lo habitan. Sus ejemplos son muy variados, podemos mencionar, entre otros, sistemas domóticos de monitorización del hogar, alertas de humo, de fuego, de gas, manejo de cerraduras y accesos, etc.

Convertir el hogar en un lugar más confortable, forma parte también de los objetivos de esta tecnología. Aporta a la gestión diaria de elementos, electrodomésticos, alertas personalizadas, climatización, etc. En este campo podemos mencionar, que la domótica permite en gran parte, el reemplazo de la mano del hombre para la realización de actividades diarias.

Optimiza y garantiza las comunicaciones, mediante el control remoto y monitorización por medio de distintos dispositivos. Esto permite sustituir la presencia humana por la domótica para la realización de actividades en el hogar, totalmente manejable a distancia.

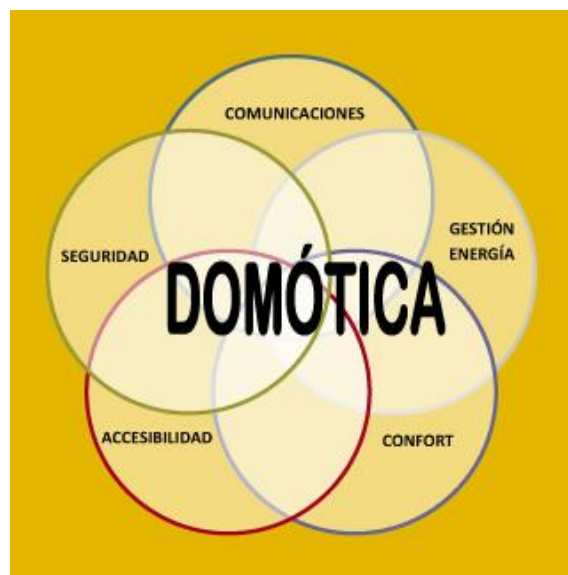


Imagen 44: Beneficios de la domótica¹

¹ Asociación Española de Domótica e Inmótica - CEDOM



Características del sector

Cuando hablamos del sector de la domótica, podemos mencionar los siguientes aspectos:

- Lo novedoso e inexplorado del tema: cuando hablamos de domótica, podemos experimentar un mayor grado de desinformación en países en vías de desarrollo o poco desarrollados. Esto se debe generalmente a que la domótica es un lujo que hoy en día no es accesible a toda la población, ya que sus aplicaciones son comúnmente cotizadas a un precio muy elevado en el mercado. Cabe mencionar, que la domótica está muy ligada al rubro de la construcción, ya que su implementación se facilita cuando el hogar está construido considerando que en el mismo se implementará esta tecnología. Esto está ligado con la dificultad que presenta la domótica para acceder al sector de la construcción.
- Su posición en el mercado: como mencionamos anteriormente, la domótica es considerada un lujo para la mayoría de la población. El acceso a esta se produce generalmente en países desarrollados, que es donde los agentes proveedores de domótica se están especializando. El avance tecnológico aporta a que los costos de creación, desarrollo e implementación de la domótica sean cada vez menores. Si bien la aplicación de esta tecnología se está conociendo en el mercado, es un proceso lento que requiere mayor desarrollo, lo cual se espera en un futuro no muy lejano.
- Las normas de seguridad necesarias para su implementación: gran parte de los sistemas domóticos que podemos encontrar en la actualidad, interactúan con personas con condiciones delicadas, lo que significa un gran control de calidad y seguridad en un producto final. Un sistema domótico de este tipo no puede tener errores, ya que los mismos podrían causar un gran daño en el usuario. Cabe mencionar que estos procesos de control aumentan en gran cantidad el costo final de la tecnología.
- El avance de la tecnología. Este es un tema esperanzador para los agentes de domótica. Permite una mayor inserción en el mercado, como aumentar la facilidad de creación e implementación. Esto se puede ver reflejado, tanto en tamaño como en complejidad de los distintos sistemas. En la actualidad existen placas programables, como Arduino (con la que trabajamos en este proyecto), Raspberry pi, Netduino, etc., que facilitan de manera significativa el crecimiento de la domótica.

Enfermedad

Conceptos

Enfermedad

El término enfermedad proviene del latín infirmitas, que significa literalmente «falta de firmeza». La definición de enfermedad según la Organización Mundial de la Salud (OMS), es la de “Alteración o desviación del estado fisiológico en una o varias partes del cuerpo, por causas en general conocidas, manifestada por síntomas y unos signos característicos, y cuya evolución es más o menos previsible”

El concepto de enfermedad, es una aproximación que orienta sobre el tipo de problema de salud en cuestión, y ayuda a su entendimiento. Toda enfermedad tiene



unas características concretas que la categoriza, y proporciona un punto de referencia para identificar qué puede tener en común o diferenciarse una entidad nosológica de otra.

Salud

La salud es un estado de completo bienestar físico, mental y social, no solamente la ausencia de enfermedad o dolencia, según la definición presentada por la Organización Mundial de la Salud (OMS) en su constitución aprobada en 1948.

Este concepto se amplía a: «La salud es un estado de completo bienestar físico, mental y social, y no solamente la ausencia de afecciones o enfermedades». En la salud, como en la enfermedad, existen diversos grados de afectación y no debería ser tratada como una variable dicotómica. Así se reformularía de la siguiente manera: «La salud es un estado de bienestar físico, mental y social, con capacidad de funcionamiento, y no sólo la ausencia de afecciones o enfermedades». También puede definirse como el nivel de eficacia funcional o metabólica de un organismo, tanto a nivel micro (celular) como a nivel macro (social). En 1992 un investigador amplió la definición de la OMS, al agregar: "y en armonía con el medio ambiente".

Cáncer

«Cáncer» es un término genérico que designa un amplio grupo de enfermedades que pueden afectar a cualquier parte del organismo; también se habla de «tumores malignos» o «neoplasias malignas». Una característica del cáncer, es la multiplicación rápida de células anormales que se extienden más allá de sus límites habituales, y pueden invadir partes adyacentes del cuerpo o propagarse a otros órganos, proceso conocido como metástasis. Las metástasis son la principal causa de muerte por cáncer.

Datos y cifras

- El cáncer es una de las principales causas de morbilidad y mortalidad en todo el mundo; en 2012 hubo unos 14 millones de nuevos casos y 8,2 millones de muertes relacionadas con el cáncer.
- Se prevé que el número de nuevos casos aumente en aproximadamente un 70% en los próximos 20 años.
- En 2012, los cánceres diagnosticados con más frecuencia en el hombre fueron los de pulmón, próstata, colon y recto, estómago e hígado.
- En la mujer fueron los de mama, colon y recto, pulmón, cuello uterino y estómago.
- Aproximadamente un 30% de las muertes por cáncer son debidas a cinco factores de riesgo conductuales y dietéticos: índice de masa corporal elevado, ingesta reducida de frutas y verduras, falta de actividad física, consumo de tabaco y consumo de alcohol.
- El consumo de tabaco es el factor de riesgo más importante: y el 20% de las muertes mundiales por cáncer en general, y alrededor del 70% de las muertes mundiales por cáncer de pulmón.
- Los cánceres causados por infecciones víricas, tales como las infecciones por virus de las hepatitis B (VHB) y C (VHC) o por papilomavirus humanos (PVH), son



responsables de hasta un 20% de las muertes por cáncer en los países de ingresos bajos y medios.¹

- Más del 60% de los nuevos casos anuales totales del mundo se producen en África, Asia, América Central y Sudamérica. Estas regiones representan el 70% de las muertes por cáncer en el mundo
- Se prevé que los casos anuales de cáncer, aumentarán de 14 millones en 2012 a 22 millones en las próximas dos décadas.

El problema

El cáncer es la principal causa de muerte a escala mundial. Se le atribuyen 8,2 millones de defunciones ocurridas en todo el mundo en 2012. Los principales tipos de cáncer son los siguientes:

- Pulmonar (1,59 millones de defunciones);
- Hepático (745 000 defunciones);
- Gástrico (723 000 defunciones);
- Colorrectal (694 000) defunciones;
- Mamario (521 000 defunciones);
- Cáncer de esófago (400 000 defunciones).

¿Cuál es la causa del cáncer?

El cáncer comienza en una célula. La transformación de una célula normal en tumoral es un proceso multifásico, y suele consistir en la progresión de una lesión precancerosa a un tumor maligno. Estas alteraciones son el resultado de la interacción entre los factores genéticos del paciente y tres categorías de agentes externos, a saber:

- carcinógenos físicos, como las radiaciones ultravioleta e ionizantes;
- carcinógenos químicos, como los asbestos, los componentes del humo de tabaco, las aflatoxinas (contaminantes de los alimentos) o el arsénico (contaminante del agua de bebida);
- carcinógenos biológicos, como las infecciones causadas por determinados virus, bacterias o parásitos.

A través de su Centro Internacional de Investigaciones sobre el Cáncer, la OMS mantiene una clasificación de los agentes cancerígenos.

El envejecimiento es otro factor fundamental en la aparición del cáncer. La incidencia de esta enfermedad aumenta muchísimo con la edad, muy probablemente porque se van acumulando factores de riesgo, de determinados tipos de cáncer. La acumulación general de factores de riesgo se combina con la tendencia que tienen los mecanismos de reparación celular a perder eficacia con la edad.



Factores de riesgo del cáncer

El consumo de tabaco y alcohol, *la dieta malsana* y la inactividad física, son los principales factores de riesgo de cáncer en todo el mundo. Algunas infecciones crónicas también constituyen factores de riesgo, y son más importantes en los países de ingresos medios y bajos.

Los virus de las hepatitis B (VHB) y C (VHC) y algunos tipos de papilomavirus humanos (PVH), aumentan el riesgo de cáncer de hígado y cuello uterino, respectivamente. La infección por el VIH también aumenta considerablemente el riesgo de algunos cánceres, como los del cuello uterino.

Modificación y prevención de los riesgos

Más del 30% de las defunciones por cáncer podrían evitarse modificando o evitando los principales factores de riesgo, tales como:

- El consumo de tabaco;
- El exceso de peso o la obesidad;
- Las dietas malsanas con un consumo insuficiente de frutas y hortalizas;
- La inactividad física;
- El consumo de bebidas alcohólicas;
- Las infecciones por PVH y VHB;
- Radiaciones ionizantes y no ionizantes;
- La contaminación del aire de las ciudades;
- El humo generado en la vivienda por la quema de combustibles sólidos.

El consumo de tabaco es el factor de riesgo más importante, y es la causa de aproximadamente un 22% de las muertes mundiales por cáncer en general, y de acerca el 70% de las muertes mundiales por cáncer de pulmón. En muchos países de ingresos bajos, hasta un 20% de las muertes por cáncer son debidas a infecciones por VHB o PVH.

Estrategias de prevención

- intensificar la práctica de los factores de riesgo recién enumerados;
- vacunar contra los PVH y el VHB;
- controlar los riesgos ocupacionales;
- reducir la exposición a la radiación no ionizante solar (ultravioleta);
- reducir la exposición a la radiación ionizante (ocupacional o pruebas médicas radiológicas).



Detección temprana

La mortalidad por cáncer se puede reducir si los casos se detectan y tratan a tiempo. Las actividades de detección temprana tienen dos componentes:

- El diagnóstico temprano: el conocimiento de los síntomas y signos iniciales (en el caso de cánceres como los de la piel, mama, colon y recto, cuello uterino o boca) es fundamental para que se puedan diagnosticar y tratar precozmente. El diagnóstico temprano es especialmente importantes cuando no hay métodos de cribado eficaces o, como ocurre en muchos entornos con escasos recursos, no se aplican intervenciones de cribado y tratamiento. En ausencia de intervenciones de detección temprana o de cribado y tratamiento, los pacientes son diagnosticados en estadios muy tardíos, cuando ya no son posibles los tratamientos curativos.
- El cribado: el cribado tiene por objeto descubrir a los pacientes que presentan anomalías indicativas de un cáncer determinado o de una lesión precancerosa, y así poder diagnosticarlos y tratarlos prontamente. Los programas de cribado son especialmente eficaces en relación con tipos de cáncer frecuentes, para los cuales existe una prueba de detección costo eficaz, asequible, aceptable y accesible a la mayoría de la población en riesgo.

Respuesta de la OMS

En 2013, la OMS puso en marcha el Plan de Acción Global para la Prevención y el Control de las Enfermedades No Transmisibles 2013-2020, que tiene como objetivo reducir la mortalidad prematura por el 25% de cáncer, enfermedades cardiovasculares, diabetes y enfermedades respiratorias crónicas. Algunas de las metas de aplicación voluntaria son especialmente importantes para la prevención del cáncer, como la que propone reducir el consumo de tabaco en un 30% entre 2014 y 2025.

La OMS y el Centro Internacional de Investigaciones sobre el Cáncer, colaboran con otras organizaciones que forman parte del Equipo de Tareas Interinstitucional de las Naciones Unidas, sobre la Prevención y el Control de las Enfermedades No Transmisibles y con otros asociados a el fin de:

- Aumentar el compromiso político con la prevención y el control del cáncer;
- Coordinar y llevar a cabo investigaciones sobre las causas del cáncer y los mecanismos de la carcinogénesis en el ser humano;
- Efectuar un seguimiento de la carga de cáncer (como parte de la labor de la Iniciativa Mundial sobre Registros Oncológicos);
- Elaborar estrategias científicas de prevención y control del cáncer;
- Generar y divulgar conocimientos, para facilitar la aplicación de métodos de control del cáncer basados en datos científicos;



- Elaborar normas e instrumentos para orientar la planificación y la ejecución de las intervenciones de prevención, detección temprana, tratamiento y atención;
- Facilitar la formación de amplias redes mundiales, regionales y nacionales de asociados y expertos en el control del cáncer;
- Fortalecer los sistemas de salud locales y nacionales, para que presten servicios asistenciales y curativos a los pacientes con cáncer;
- Prestar asistencia técnica para la transferencia rápida y eficaz de las prácticas óptimas a los países en desarrollo.

Dietas

Los alimentos alcalinos son la base para una buena salud y en los que respecta al cáncer, la acidez en la sangre produce cáncer y también la mayoría de las enfermedades que se producen en nuestro cuerpo. Así lo demostró el Premio Nobel Dr. Otto Warburg que pudo comprobar, como las células cancerígenas crecen en el ambiente ácido cuando las células del cuerpo se quedan sin oxígeno y no en el alcalino, ya que en estas circunstancias no son capaces de reproducirse.

Alimentos alcalinos y pH

Los alimentos alcalinos contra el cáncer son los más beneficiosos, ya que proporcionan un medio alcalino en el cuál las células del cáncer no pueden proliferar. Las células cancerígenas crecen en los medios ácidos y por lo tanto es muy importante diferenciar entre alimentos alcalinos y ácidos, porque sabiendo qué alimentos son ácidos y cuáles son los que alcalinizan nuestro cuerpo, tenemos mucho a favor de nuestra salud.

Los alimentos alcalinos o alimentos bases son los que elevan el pH y son alimentos no ácidos y por el contrario los alimentos ácidos son lo que bajan el pH de nuestra sangre. Los alimentos alcalinos son los que debemos consumir para tener una buena salud; los alimentos neutros también deben formar parte de la dieta y por último los alimentos ácidos son los que debemos de evitar en la medida de lo posible.

Recordemos que nuestro cuerpo siempre tiende hacia la acidez, por eso es tan importante conocer bien qué alimentos comemos e independientemente de sus aportes nutricionales el conocer si son ácidos, neutros o alcalinos es importantísimo, no solo para prevenir e incluso curar el cáncer como muchas personas ya han hecho, sino para prevenir cualquier tipo de enfermedad.

Ahora vamos a ver cuáles son los alimentos que nos proporcionarán un medio alcalino en nuestro organismo.



En la tabla 15 se observa un listado de verduras alcalinas, en la tabla 16 un listado de frutas alcalinas y en la tabla 17 otro listado de alimentos alcalinos. A su vez, en la tabla 18 observamos un listado de alimentos neutros y en la tabla 19 alimentos ácidos.

Lista de alimentos alcalinos

Verduras Alcalinas						
Acelga	Remolacha	Zanahoria	Pepino	Hojas de mostaza	Nabo	Semillas Germinadas
Alfalfa	Ejotes	Repollo	Col rizada	Cebolla	Rábano	Zapallo
Apio	Lechuga	Coliflor	Dientes de león	Batatas	Tomate	Papa
Berenjena	Pimientos	Ajo	Hongos	Trigo	Espinaca	Berro
Vegetales fermentados	Hierba de cebada	Calabaza	Guisantes	Chirivía	Spirulina	Pepino

Tabla 15: Verduras Alcalinas

Frutas Alcalinas				
Manzana	Pomelo	Guayaba	Melón	Sandia
Albaricoque	Pera	Piña	Palta	Banana
Mango	Moras	Cerezas	Frambuesas	Coco
Pasas	Dátiles	Bayas	Higos	Limón
Lima	Uvas	Nectaria	Durazno	Naranja
Mandarina	Melocotón	Frutilla	Frutas tropicales	Ruibarbo

Tabla 16: Frutas Alcalinas

Otros alimentos alcalinos				
Almendras	Aceite de oliva virgen	Castañas	Tempeh	Tofu
Mijo	Chile	Jengibre	Canela	Curry
Suero de leche	Mostaza	Sal marina	Melaza	Vinagre de manzana
Jugo de frutas	Jugo de verduras	Polen de abeja	Lecitina	Tamari
Miso	Probióticos	Agria lácteos	Agua mineral	Cesio

Tabla 17: Otros Alimentos Alcalinos



Lista de alimentos neutros

Alimentos neutros				
Miel	Huevos	Papas	Aceites	Porotos
Leche de soja	Maíz	Lentejas	Aceitunas	Moras
Arándanos	Ciruelas	Pasas	Cebada amaranto	Avena
Salvado de avena	Salvado de trigo	Maicena	Harina de semillas de cáñamo	Kamut
Harina de avena	Quínoa	Arroz	Pasteles de arroz	Centeno
Espelta	Trigo	Germen de trigo	Fideos	Espaguetis
Harina de trigo	Frijol negro	Garbanzos	Judías	Lentejas
Frijoles pintos	Soja	Leche de soja	Judías blancas	Leche de arroz

Tabla 18: Alimentos Neutros

Lista de alimentos ácidos

Alimentos ácidos				
Mantequillas	Queso	Helados de leche	Crema	Anacardos
Maní	Nueces	Mantequilla de maní	Tahini	Carne
Tocino	Carpa	Pescado	Almeja	Bacalao
Atún	Merluza en alta mar	Salmón	Sardinas	Mariscos
Camarones	Mejillones	Vieiras	Langosta	Ostras
Cordero	Conejo	Carne de venado		

Tabla 19: Alimentos Ácidos

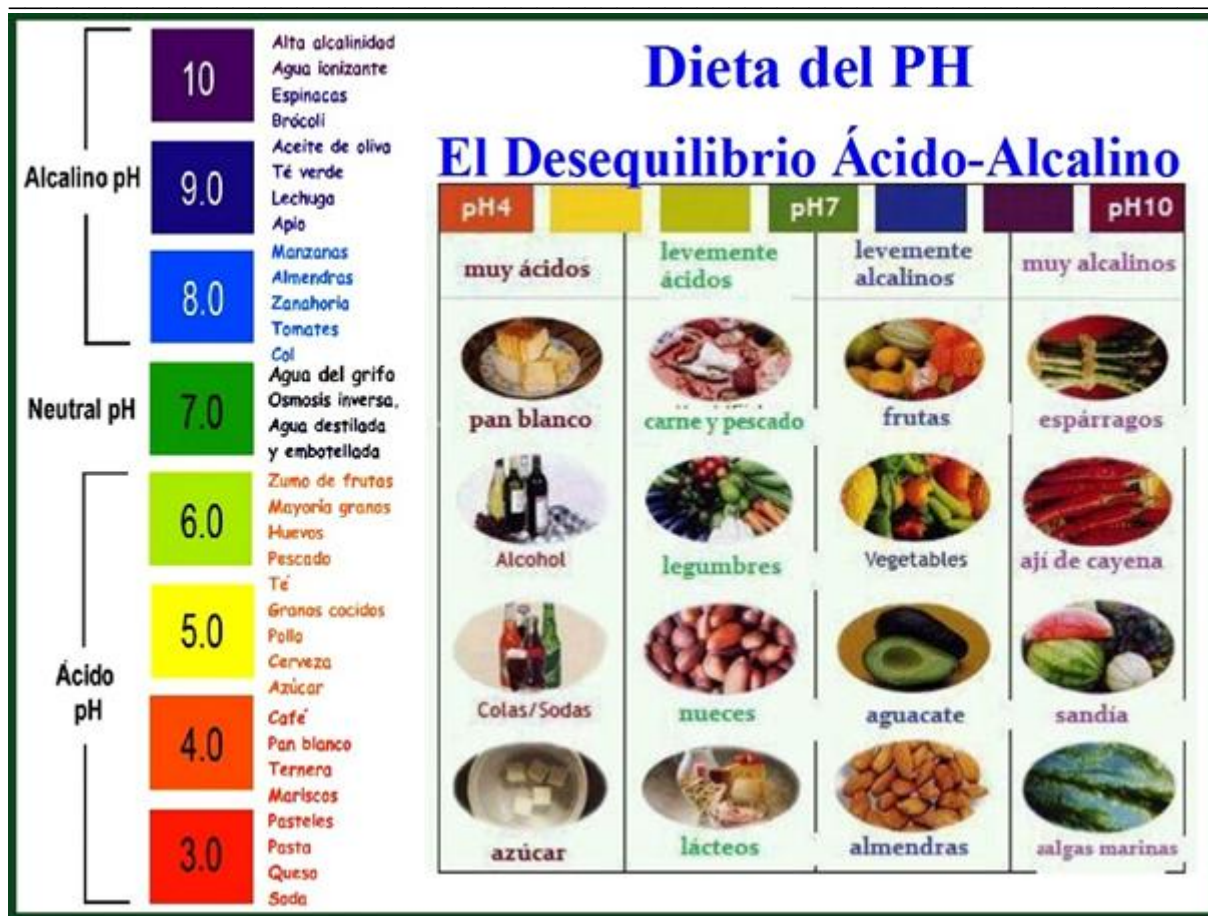


Imagen 45: Dietas por pH

pH

El pH es una forma de representar el carácter ácido, neutro o básico de una solución acuosa.

El agua pura está levemente dissociada en iones H^+ e iones OH^- por partes iguales. Cuando la cantidad de iones H^+ es mayor a la cantidad de iones OH^- nos encontramos con una solución ácida. En caso inverso ($OH^- > H^+$) nos encontramos con una solución básica.

La escala del pH

La forma más cómoda de medir la concentración de iones, es mediante una escala logarítmica, que es el llamado pH. La escala va de 0 a 14, y el punto neutro es el 7. El pH neutro 7 (agua pura) es equivalente a una concentración de 10^{-7} iones-gramo por litro.

El pH del ácido clorhídrico (un ácido fuerte) es 1. El pH del hidróxido sódico (una base fuerte) es 13,5. Un pH inferior a 7 representa una solución ácida. Un pH superior a 7 representa una solución básica o alcalina.



La sangre es ligeramente alcalina (pH de 7'35 a 7'45), mientras que en el estómago hay un ambiente ácido (para permitir la digestión), de pH 2 a 3 en personas adultas, y 4 (más suave) en bebés lactantes.

Homeostasis y regulación del nivel de pH en el cuerpo

Hay varios sistemas mediante los cuales nuestro organismo regula los niveles de acidez o alcalinidad del cuerpo:

- Mediante la respiración, que hace que el anhídrido carbónico que espiramos, de un carácter ácido a la sangre.
- Mediante los riñones, ya que nuestro organismo forma una orina más o menos ácida según las necesidades de equilibrio.
- Mediante la alimentación. Una dieta errónea puede provocar acidosis. El cuerpo también dispone de un "almacén" de bases orgánicas que utiliza para restablecer el equilibrio en caso de que el pH del plasma sanguíneo varíe. Una mala alimentación provoca que estas reservas disminuyan y que pueda provocarse acidosis.

¿Cómo se mide el pH?

Sangre

Por lo general la medición del pH se hace a través de la sangre, ya que es más sencillo de medir que el de otras partes del cuerpo. Lo que se determina a través del análisis de la sangre son los niveles de: potasio, sodio, cloro, dióxido de carbono y bicarbonato.

Orina

Algunos practicantes de la medicina alternativa miden también la orina. Ellos piensan que el cuerpo busca permanentemente eliminar los excesos de acidez que irritan los tejidos y le impiden absorber los minerales; y puesto que uno de los sistemas principales de eliminación del organismo es el renal (riñones), ellos consideran que al medir el nivel de acidez de la orina, se puede determinar si el organismo está eliminando o no la cantidad de ácidos que debería. Si la eliminación de ácidos a través de la orina es más alta de lo normal, el pH de la orina será más ácido. Esto también indicaría que el cuerpo está saturado y por eso tiene una acidez alta.

Saliva

Dentro de la medicina alternativa se mide también la saliva, pues se considera que si el cuerpo tiene las suficientes reservas de minerales; esta abundancia de minerales debe aparecer en la saliva. No consideran sin embargo, que medir la orina o la saliva, sean un modo completo de medir el pH del cuerpo, sino que más bien son indicadores.



¿Que origina un pH ácido?

Un pH ácido puede ocurrir por el estrés emocional, sobrecarga de toxinas, las reacciones inmunes o cualquier proceso que prive a las células de oxígeno y otros nutrientes. Obviamente, una dieta que es muy ácida, puede cambiar también los niveles de pH hasta cierto punto.

Nuestro pH en la sangre, tiene una gama muy estrecha de alrededor de 7,35 a 7,45. Si el pH de nuestro cuerpo se desvía de esta gama y cae por debajo de 6,8 o por encima de 7,8, las células dejan de funcionar correctamente y nuestro cuerpo enfermará o comenzará a mostrar síntomas de enfermedad. Por ejemplo, la capacidad de contracción de los músculos declina y la hormona adrenalina aumenta cuando el cuerpo se vuelve ligeramente más ácido. Las partes del cuerpo tienen distintos niveles de acidez y alcalinidad.

Ante una desviación del pH, el cuerpo trata de compensar un nivel ácido mediante la utilización de minerales alcalinos que ha almacenado. Si la dieta no contiene suficientes minerales para compensar, se producirá una acumulación de ácidos en las células, privándolas de oxígeno.

La alteración del pH, es fomentada por dietas altas en azúcar o en hidratos de carbono (carbohidratos), así como por beber agua contaminada y respirar aire contaminado, de lo cual muchas veces ni nos damos cuenta. También se provoca alteración, cuando nuestra flora intestinal se deteriora y hasta se destruye a causa del uso de antibióticos u otros medicamentos fuertes, como la quimioterapia.

Pedrajas, naturópata energético, aporta su punto de vista, asintiendo también una acidez como consecuencia de una mala gestión de las emociones y sentimientos. "El estrés es un mecanismo que crea acidez en el organismo, e incluso hay emociones que también lo hacen, como por ejemplo la rabia, la ira, la impotencia, la falta de serenidad, la represión..."

No podemos hacerlos cargo de las enfermedades a las bacterias y hongos, ya que sin ellos no podríamos vivir, son los que nos dan vida. La causa está en el desequilibrio que se genera cuando creamos un cuerpo ácido, por una intoxicación de elementos ajenos a nuestro organismo, como son las toxinas que ingerimos de forma gratuita a través de la respiración, los alimentos, y medicamentos de origen sintético, etc., y aun así, si hiciéramos una gestión inteligente de nuestras emociones y sentimientos, esos tóxicos se oxidarían sin llegar a mas consecuencia, pero si además de un cuerpo toxico, alimentado con tóxicos, tratado con tóxicos, lo alimentamos con emociones tóxicas, entonces la batalla está perdida.



Por lo tanto, las principales fuentes de acumulación de ácido son:

1. El metabolismo y/o desglose incompleto (oxidación) de los productos alimenticios o metabólicos “residuos” producido como subproducto de actividad celular. Durante la respiración celular y la producción de energía, los ácidos producidos como parte de productos de “desechos”. Estos ácidos deben ser “equilibrados”, neutralizados, o eliminados por el cuerpo y los sistemas de desintoxicación a través de los intestinos, riñones, pulmones, el hígado y la sangre.
2. El consumo de ácido presente en el suministro de alimentos, aire y agua. Las emisiones de nitrógeno de los automóviles y las plantas industriales, colorantes alimentarios, aerosoles, ceras, conservantes, aditivos alimentarios, los edulcorantes artificiales, los fertilizantes, contaminantes del agua, e incluso cloruro y fluoruro en el agua del grifo, son sólo algunos de los productos químicos altamente ácidos que son ingeridos por millones de personas cada día.
3. La gestión de las emociones y sentimientos. Desde la inquietud hasta un sentimiento de rabia continuado, generan mecanismos de estrés liberando adrenalina aumentando el nivel de glucosa en la sangre, y al igual que cuando ingerimos en exceso alimentos ricos en glucosa, y carbohidratos, esta, si no es consumida o quemada por ejercicio físico, queda como residuo tóxico, generando un ambiente ácido en nuestro cuerpo, rompiendo su equilibrio bacteriano.

Como afecta un pH ácido a nuestro organismo

Los hábitos basados en estos alimentos, además de la acidez, crean adicción, y es entonces cuando sentimos la urgencia de ingerir algo dulce, o carbohidratos, que se convierten en glucosa, o en alcohol. Para poder procesar dichas toxinas, el hígado las convierte en alcohol (ácido) y ese exceso de alcohol en nuestro organismo, nos produce una sensación como la de estar borracho... mareado, desorientado, mentalmente confundido

La acidificación conlleva a una acumulación excesiva de bacterias y hongos, que reduce la provisión de potasio y magnesio del cuerpo. Esto puede disminuir la capacidad del organismo para absorber minerales adicionales y otros nutrientes, disminuye la producción de energía en las células, y su capacidad para reparar las células dañadas; disminuye su capacidad de desintoxicar los metales pesados, permitiendo que las células tumorales crezcan y proliferen, haciéndolo más susceptibles a la fatiga y la enfermedad.

Otros síntomas son los ataques de pánico, ansiedad, depresión, irritabilidad, dolores de cabeza, dolores en las articulaciones, inflamación en vías respiratorias, sinusitis, estrés



glandular y problemas menstruales, quita el entusiasmo, la ambición, la stamina, y causa la liberación de radicales libres, los cuales colaboran en el proceso de envejecimiento.

Sintomatología de un pH ácido

La acumulación de ácido puede provocar acidosis, que a su vez afecta al hígado, riñón, corazón, la insulina y la regulación de la salud celular. Estos problemas pueden llevar rápidamente a un accidente cerebrovascular, infarto de miocardio, aneurisma, problemas cardiovasculares, la osteoporosis, la obesidad y la hipertensión arterial.

La acidosis generalmente altera los lípidos y ácidos grasos, los cuales están involucrados en la función nerviosa y cerebral. Esta alteración provoca problemas neurológicos como la esclerosis múltiple, MD, así como problemas con el equilibrio hormonal en el sistema endocrino.

Es muy probable que un pH ácido, a partir de una dieta desequilibrada, produzca una condición que estimula la respuesta genética predeterminada a la inanición y el hambre. Cuando esto sucede, el cuerpo comienza a acumular todas las calorías consumidas y la almacena como grasa, por temor a que se muera de hambre y no sabe dónde ni cuándo tendrá su próxima comida. Para contrarrestar esto, muchos nutricionistas y atletas recomiendan el consumo de numerosas comidas pequeñas y saludables durante todo el día con el fin de mantener al cuerpo seguro. Cuando el cuerpo está en este cómodo estado y sin preocupaciones, quemará esas calorías (un aumento del metabolismo), lo que ayuda a adelgazar la cintura, crear masa muscular y evitar muchos de los problemas que se asocian con la acidosis, la obesidad y la regulación de insulina.

Otto Heinrich Warburg

Como una de las ciencias básicas de la medicina, la fisiología explica las funciones orgánicas bien como procesos energéticos, o bien como procesos materiales. El primer caso se basa en la física y, en el segundo, en la química. Históricamente ambas tendencias surgieron por separado pero después acabaron convergiendo. Uno de los capítulos más desarrollados de la fisiología es el correspondiente a las bases químicas de las funciones orgánicas. El apoyo de la fisiología en la química se ha producido en tres fases.

En un principio se apoyó en la química orgánica con el fin de aclarar la composición de la materia viva. Después lo hizo en la constitución de la química fisiológica como explicación de la dinámica material de los procesos orgánicos. Por último, en la bioquímica como disciplina biológica autónoma de carácter básico.



La bioquímica, tal como la entendemos hoy, se constituyó en las primeras décadas del siglo XX y surgió como un área interdisciplinar, que se centró en la investigación de las reacciones químicas que tienen lugar en los seres vivos y de los enzimas que las regulan. Una de sus características es que son específicos, es decir, catalizan la reacción entre dos moléculas o dos tipos determinados de moléculas.

La investigación de los enzimas ha ocupado una buena parte de los trabajos de los bioquímicos del siglo XX: técnicas para aislarlos, estudio de sus propiedades físicas y químicas, etc. Uno de los conceptos centrales de la bioquímica es el de metabolismo: suma total de reacciones enzimáticas que tienen lugar en la célula; se trata de una actividad muy integrada y "pletórica" de propósitos, en la que participan multitud de conjuntos de sistemas multienzimáticos, con el fin de intercambiar materia y energía entre la célula y su entorno.

Es en este contexto en el que debemos situar la obra de Otto Warburg, que significó un hito en la comprensión bioquímica de numerosos procesos orgánicos y que algunos consideran como el más original y productivo del siglo XX. Uno de sus grandes hallazgos, fue demostrar que las células utilizan oxígeno, lo que ayudó a clarificar numerosos procesos metabólicos.

Se sabía que varias reacciones químicas estaban implicadas en los procesos biológicos. Warburg aportó perspectivas desde la física, nuevas técnicas de medición precisa y estudió las reacciones moleculares in vitro. Estos acercamientos fueron muy fructíferos para la investigación médica.

Otto Heinrich Warburg nació en Friburgo, Breisgau, Baden, el 8 de octubre de 1883. Realizó sus estudios universitarios en las Universidades de Berlín, Múnich y Heilderberg. Uno de sus principales maestros fue Emil Fischer, que le enseñó a seguir los estándares científicos más rigurosos en la planificación y desarrollo de experimentos.

Warburg se fue decantando hacia la investigación médica básica. Tras recibir su doctorado en química en 1906, marchó a Heilderberg para estudiar medicina; obtuvo el grado en 1911.

En 1912 Warburg postuló la existencia de un enzima respiratorio activador del oxígeno, descubrió su inhibición por el cianuro y demostró que el hierro era indispensable en la respiración. Utilizó recursos de la física y de la química que, hasta entonces, todavía seguían al servicio de la exploración del mundo inorgánico.



Cuando tenía 30 años marchó al Káiser Wilhelm Institute (1913), con una buena reputación como científico. Comenzó la Primera Guerra Mundial y sirvió como oficial. Cuando en 1918 regresó al Instituto, decidió dedicarse de lleno a la investigación.

En 1930, ya había identificado un enzima como responsable de la canalización de las reacciones de oxidación celular. En 1931 le fue otorgado el Premio Nobel de fisiología y medicina, por su descubrimiento de la naturaleza y modo de acción del enzima respiratorio.

A este hallazgo habría que añadir, la creación de técnicas de laboratorio muy sofisticadas. Ideó lo que se llaman “cortes supervivientes”, para el estudio del metabolismo intermediario de los tejidos animales o vegetales.

Más fructíferas fueron sus investigaciones sobre el cáncer. Observó que las células cancerígenas podían reproducirse sin oxígeno, y lanzó la hipótesis de que la privación de oxígeno era la causa de la enfermedad. Según esta teoría, las células corporales que no utilizan oxígeno, no desarrollarían cáncer como efectivamente ocurre. Estas son los glóbulos rojos, la córnea, el cristalino y ciertas regiones de la retina. Estas células no tienen mitocondrias y dependen sólo de la glicólisis.

Estas ideas fueron bien recibidas. Según él, había que introducir en la dieta, varias enzimas respiratorias, como vitaminas B y hierro, lo que ayudaría a prevenir la aparición del cáncer.

Warburg tenía una personalidad difícil y algo excéntrica. Los últimos años de su vida pensaba que la enfermedad se debía a la contaminación. No comía nada que no pudiera controlar.

Sus estudiantes y colaboradores, le estuvieron siempre agradecidos por sus contribuciones y lo reconocieron como uno de los grandes científicos contemporáneos.

Lenguajes de programación

Ruby

Ruby es un lenguaje de programación creado por Yukihiro Matsumoto, quien mezcló diferentes idiomas (Perl, Smalltalk, Eiffel, Ada y Lisp) para lograr, lo que es hoy, uno de los principales referentes en cuanto a la programación web.

Este lenguaje posee numerosas características, que hacen que la programación adquiera una simpleza que no se logra fácilmente, y aun así, permite explorar a un nivel tan bajo como el programador lo desee.



Ruby es un lenguaje de programación basado en el sistema MVC (Modelo – Vista – Controlador), orientado a objetos y totalmente libre. Con metodologías DRY fuertemente establecidas en los estándares.

Desde su liberación pública en 1995, Ruby ha atraído devotos desarrolladores de todo el mundo. En el 2006, Ruby alcanzó reconocimiento masivo, formándose grupos de usuarios activos en las ciudades más importantes del mundo, llenando las capacidades de las conferencias relacionadas a Ruby.

El índice TIOBE, que mide el crecimiento de los lenguajes de programación, ubica a Ruby en la posición #13 del ranking mundial. Refiriéndose a su crecimiento, predicen: “Todo indica que Ruby llegará a estar entre los 10 primeros en menos de 6 meses”. Gran parte de su crecimiento, se atribuye a la popularidad alcanzada por aplicaciones desarrolladas con Ruby, en particular el *framework* de desarrollo web Ruby on Rails.

Ruby es totalmente libre. No sólo gratis, sino también libre para usarlo, copiarlo, modificarlo y distribuirlo.

Paradigma

“Quería un lenguaje que fuera más poderoso que Perl, y más orientado a objetos que Python” (Matsumoto).

Como dijimos anteriormente, Ruby es un lenguaje orientado a objetos. En Ruby todo es un objeto. Se le puede asignar propiedades y acciones a toda información y código. Esto nos facilita la reutilización y la extensibilidad del código, permitiéndonos llevar al cabo el estándar DRY (*Don't repeat yourself* – no te repitas a ti mismo), optimizando tanto los tiempos de programación, la calidad del *software* final, y la comprensión de los programadores.

La flexibilidad de Ruby

Ruby es considerado un lenguaje flexible, ya que permite a sus usuarios alterarlo libremente. Las partes esenciales de Ruby, pueden ser extraídas o redefinidas a placer. Se puede agregar funcionalidad a partes ya existentes. Ruby intenta no restringir al desarrollador.

Los operadores de Ruby, son simples conveniencias sintácticas para los métodos. Se pueden redefinir como y cuando quieras.

Los Bloques

Los bloques de Ruby son también vistos como una fuente de gran flexibilidad. El desarrollador puede anexas una cláusula a cualquier método, describiendo cómo debe actuar. La cláusula es llamada “bloque”, y se ha convertido en una de las más famosas funcionalidades para los recién llegados a Ruby, que vienen de otros lenguajes imperativos como PHP o Visual Basic.



Los bloques están inspirados por los lenguajes funcionales. Matsumoto dijo, “*en las cláusulas de Ruby, quise respetar la cultura de Lisp3*”.

```
motores_de_busqueda =  
  %w[Google Yahoo MSN].map do |motor|  
    "http://www." + motor.downcase + ".com"  
  end
```

En este código, el bloque está descripto entre la construcción `do... end`. El método `map` aplica el bloque a la lista de palabras provistas. Muchos otros métodos en Ruby dejan abierta la posibilidad al desarrollador para que escriba su propio bloque, describiendo los detalles de qué debe hacer ese método.

Ruby y el Mixin

A diferencia de otros lenguajes de programación orientada a objetos, Ruby se caracteriza por su intencional herencia simple. Sin embargo, Ruby incorpora el concepto de módulos (llamados categorías en Objective-C), que son colecciones de métodos.

Las clases pueden mezclar (Mixin) un módulo e incorporar todos sus métodos gratuitamente. Por ejemplo, cualquier clase que implemente el método `each`, puede incorporar el módulo `Enumerable`, que le agrega un conjunto de métodos que usan `each` para recorrer sus elementos.

```
class MiArray  
  include Enumerable  
end
```

La apariencia visual de Ruby

A pesar de que Ruby utiliza la puntuación muy limitadamente, y se prefieren las palabras claves en inglés, se utiliza algo de puntuación para decorar el código. Ruby no necesita declaraciones de variables. Se utilizan convenciones simples, para nombrar y determinar el alcance de las mismas.

- `var` puede ser una variable local.
- `@var` es una variable de instancia.
- `$var` es una variable global.

Estos detalles mejoran la legibilidad permitiendo que el desarrollador identifique fácilmente los roles de las variables. También se hace innecesario el uso del molesto `self` como prefijo de todos los miembros de instancia.

Ruby on Rails

Ruby on Rails es un entorno de desarrollo web de código abierto, que está optimizado para la satisfacción de los programadores y para la productividad sostenible. Te permite escribir un buen código, evitando que te repitas y favoreciendo la convención antes que la configuración.



¿Qué es Rails?

Rails es un almacén para construir aplicaciones web que acceden a bases de datos.

Un conjunto de librerías, automatismos y convenciones destinados a resolver los problemas más comunes a la hora de desarrollar una aplicación web, para que el programador pueda concentrarse en los aspectos únicos y diferenciales de su proyecto, en lugar de los problemas recurrentes.

¿Para qué se utiliza?

Para ayudar a construir aplicaciones modernas de internet como: Twitter, Scribd, Hulu, Xing, Soundcloud, Basecamp, Github, etc.

¿Cuál es su Origen?

Rails fue creado en 2003 por David Heinemeier Hansson y desde entonces, ha sido extendido por el *Rails core team*, más de 2.100 colaboradores y soportado por una extensa y activa comunidad.

Base de datos

PostgreSQL

PostgreSQL es un administrador de base de datos de código abierto. Ruby on Rails es un web framework de código abierto escrito en Ruby. Rails no se apeg a ninguna base de datos, permite la elección de distintos administradores para comodidad del programador. Por defecto, Rails utiliza MySQL, pero es muy simple cambiar este motor.

Requerimientos de Instalación

Usando RVM

La forma más simple de instalar Rails es utilizando RVM, que también permite instalar Ruby. Para instalar RVM, necesitamos asegurarnos de que tenemos curl instalado en nuestro sistema (como hacer esto depende de cada sistema operativo).

RVM puede instalar Ruby y Rails automáticamente en un mismo comando:

```
\curl -L https://get.rvm.io | bash -s stable -rails
```



RVM se instalará así mismo en el sistema. Ahora podremos utilizarlo para administrar las distintas versiones de Ruby. Esto es muy útil para poder cambiar a distintas versiones, así como nuestros proyectos lo exijan.

Instalando Postgres

El método para instalar Postgres depende de cada sistema operativo. Se puede obtener una lista completa en la siguiente dirección: **<http://postgresql.org/download>**. Generalmente es más simple usar un administrador de paquetes como apt-get en Ubuntu, o Homebrew en OS X.

Una vez instalado Postgres, debemos instalar la gema perteneciente al mismo en nuestro proyecto de Ruby on Rails. Para esto debemos acceder al Gemfile de nuestro proyecto y añadir una nueva línea con el código:

```
gem 'pg'
```

O también podemos hacerlo por medio del siguiente código en la consola de nuestro SO:

```
gem install pg
```

Configurando Postgres

Para crear un usuario de Postgres para la aplicación desarrollada en Rails, se utiliza el siguiente comando:

```
su - postgres
```

Luego, debemos crear un rol. Esto se realiza mediante el siguiente comando:

```
create role myapp with createdb login password 'password1';
```

Creando la aplicación en Rails

Para crear la aplicación en rails, utilizando Postgres, debemos correr este comando en la consola:



```
rails new myapp -d postgresql
```

Esto crea un directorio llamado “myapp” que contiene la aplicación. Rails espera que el usuario de la base de datos, coincida con el nombre de la aplicación. Esta configuración es fácilmente editable.

Ahora configuraremos la base de datos con la que Ruby interactuará. Esto se hace utilizando el archivo `database.yml`, localizado en el directorio `/myapp/config/`

El archivo `database.yml`, es usado por Rails para conectar con la base de datos, apropiada para el ambiente en el cual se trabaja (producción, desarrollo, prueba). Se utiliza YAML, un estándar en la serialización de datos.

Se debe configurar todos los campos necesarios y expuestos en el archivo `.yml`. Una vez realizado esto, el archivo debería quedar como sigue:

```
development:
  adapter: postgresql
  encoding: unicode
  database: myapp_development
  pool: 5
  username: myapp
  password: password1

test:
  adapter: postgresql
  encoding: unicode
  database: myapp_test
  pool: 5
  username: myapp
  password: password1
```

Luego debemos correr el siguiente comando en la consola

```
rake db:setup
```

Esto creará las base de datos de desarrollo y prueba, asignara sus correspondientes usuarios, y creará el esquema de migraciones (`schema.rb`), ubicado en **RAILS_ROOT/db/**

Levantar la aplicación

Podemos iniciar la aplicación corriendo el siguiente comando en consola:



```
rails server o rails s
```

Si accedemos desde nuestro navegador a **localhost:3000**, podremos ver una aplicación estándar de rails, que indica que todo el proceso de creación fue exitoso.

Podemos especificar distintos puertos a la hora de levantar el servidor, para esto especificamos en puerto con el siguiente comando en consola:

```
rails s -p 4000
```

Esto levantará la aplicación, a la cual podremos acceder por el puerto 4000, mediante la siguiente dirección en nuestro navegador: **localhost:4000**.

Hardware

Arduino

Arduino es una plataforma de prototipos electrónica de código abierto (*open-source*) basada en *hardware* y *software* flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como *hobby* y para cualquiera interesado en crear objetos o entornos interactivos.

Arduino puede “sentir” el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el “*Arduino Programming Language*” (basado en Wiring) y el “*Arduino Development Environment*” (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con *software* en ejecución en un ordenador (por ejemplo con Flash, Processing, MaxMSP, etc.).

Las placas se pueden ensamblar a mano o encargarlas pre ensambladas; el *software* se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD)

están disponibles bajo licencia open-source, por lo que somos libres de adaptarlas a nuestras necesidades. Arduino recibió una mención honorable en la sección Digital Communities del Ars Electrónica Prix en 2006.

Elementos de un sistema domótico

Los sistemas domóticos, están compuestos principalmente de uno o varios elementos. Se pueden clasificar los dispositivos de un sistema en:

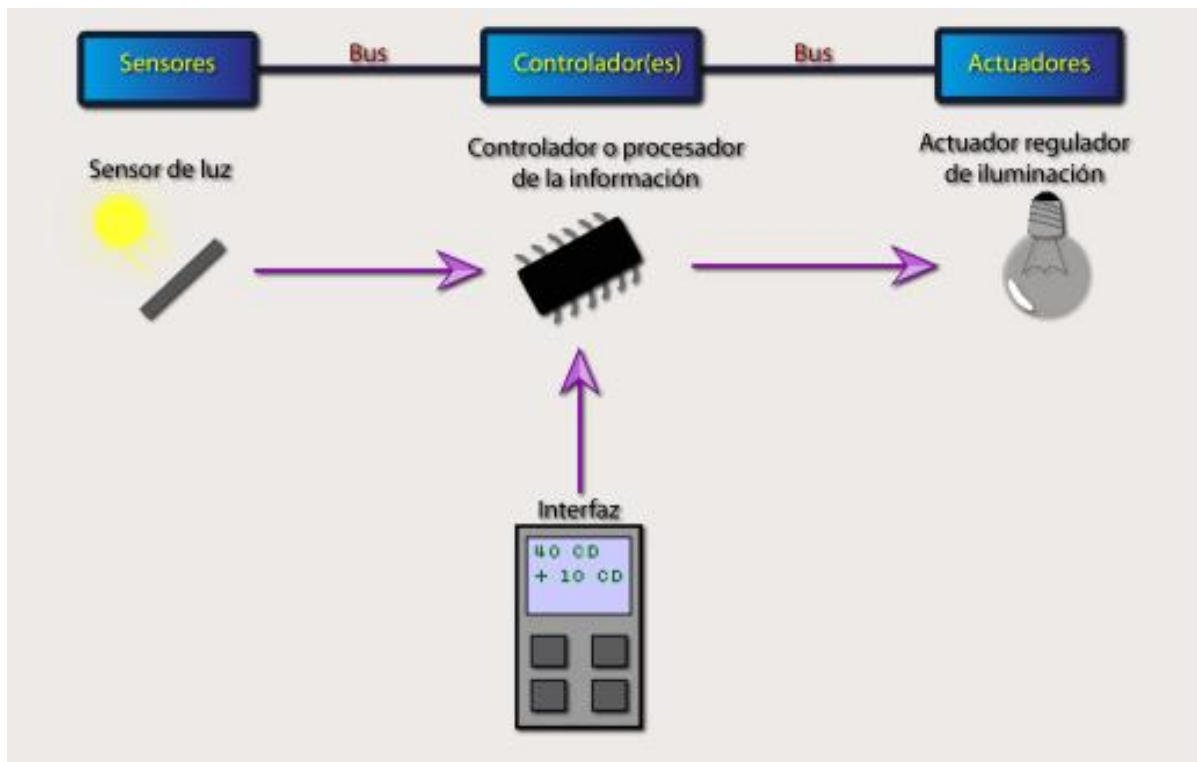


Imagen 46: Elementos de un Sistema Domótico

- **Sensor** – El sensor es el dispositivo que monitoriza el entorno captando información que transmite al sistema (sensores de agua, gas, humo, temperatura, viento, humedad, lluvia, iluminación, etc.).
- **Controlador** – Los controladores son los dispositivos que gestionan la información que reciben del sistema y “deciden” que hacer según la programación que tengan. Puede haber un controlador solo, o varios distribuidos por el sistema.
- **Actuador** – El actuador es un dispositivo capaz de recibir una orden y ejecutarla, así cambiando las características del entorno domótico (encendido/apagado, subida/bajada, apertura/cierre, etc.).
- **Bus** – Es el medio de comunicación que transporta la información entre los distintos dispositivos, ya sea por una red propia, o por la redes de otros sistemas (red eléctrica, red telefónica, red de datos); también puede ser de forma inalámbrica.
- **Interfaz** – La interfaz son los dispositivos (pantallas, móvil, Internet, interruptores) en que se muestra la información del sistema para los usuarios, y donde ellos mismos pueden interactuar con el sistema.



Elementos del prototipo del sistema domótico

ESP-8266

De todos los productos hoy en día en el mercado pensados para IOT (Internet de las Cosas), este es sin dudas uno de los mejores. En orden de importancia vamos a destacar su costo, que es lo que lo hizo tan popular; por unos 200 pesos argentinos, le podemos dar comunicación Wifi a cualquier microcontrolador.

Cabe mencionar, que el ESP-8266, ya es un micro en si con 16 pines GPIO (*General Purpose Input Output*). Otra característica en la teoría es su bajo consumo de energía, en modo *sleep* estamos hablando de uA. Decimos en teoría, porque para iniciar, el módulo llega a consumir unos 200mA y mas también, corriente que el Arduino no puede brindarle, y esto puede convertirse en un problema, aunque solucionable de momento, con una fuente para protoboard.

Este módulo, incluye todo lo necesario, para conectarse a un punto de acceso Wifi mediante comandos de texto AT, vía una puerta serie, que puede ser configurada a diferentes velocidades. Una vez que lo programamos para conectarse a nuestra red Wifi, el módulo es capaz de enviar información que le remitimos vía la puerta serie, a una dirección IP y puerto que deseemos.

Cuando se trata de recibir, limpia todo el empaquetado TCP/IP y nos reenvía por la puerta serie la información de datos, con lo que tiene la enorme virtud de permitirnos olvidarnos de la gestión del TCP/IP, y de las demandas de procesador y memoria que suponen.

Conexionado

Es importante mencionar que con este módulo, no se puede hacerlo funcionar sin la fuente para protoboard, ya que en el momento en que se enciende, puede llegar a tener picos de consumo de más de 200mA, corriente que el Arduino no puede entregar.

El módulo se alimenta estrictamente con 3.3V, los pines de señal soportan tensiones de 5v. En varios foros colocan resistencias para su protección y funcionamiento, pero el módulo funciona sin problemas sin estas.

Como último dato de importancia, como muestra el circuito, debemos compartir la tierra de nuestro Arduino con la del resto del circuito.

En la imagen 47 se puede ver una descripción de los pines del módulo.

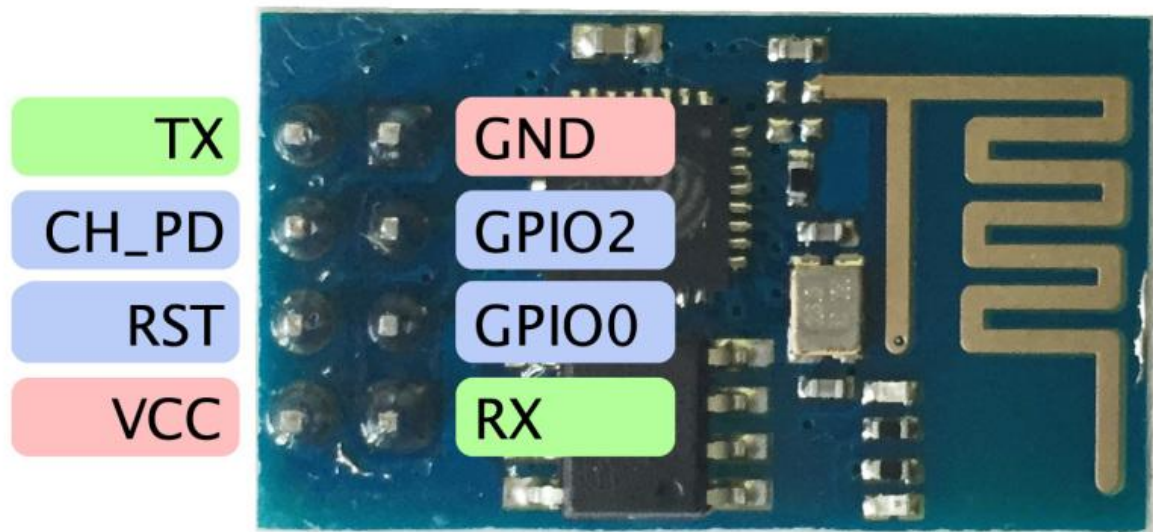


Imagen 47: Conexión ESP-8266

Características Técnicas

- Protocolos soportados: 802.11 b/g/n
- Wifi Direct (P2p), Soft Access Point
- Stack TCP/IP integrado
- PLL, reguladores y unidades de manejo de energía integrados
- Potencia de salida: +19.5dBm en modo 802.11b
- Sensor de temperatura integrado
- Consumo en modo de baja energía: <10 uA
- Procesador integrado de 32 bits, puede ser utilizado como procesador de aplicaciones
- Wifi 2.4 GHz, soporta WPA/WPA2
- Tamaño ultra reducido (11.5mm x 11.5mm)
- Conversor analógico a digital de 10-bit
- Soporta variedad de antenas
- Integrated low power 32-bit MCU
- SDIO 2.0, SPI, UART, I2C
- Encendido y transmisión de datos en menos de 2ms
- Rango de operación -40C° ~ 125C°

Usando el Firmware AT

El *firmware* AT que viene por defecto en el ESP-8266 es el ai-thinker V0.9.2.4.

Utilización de los comandos

Tipo	Ejemplo	Descripción
Prueba	AT+CMD=?	Retorna los posibles valores que puede tomar
Consulta	AT+CMD?	Retorna el valor actual del parametro
Establecer	AT+CMD=Parametro	Establece el valor del parametro de usuario
Ejecutar	AT+CMD	Ejecuta comandos sin parametros de usuario



Listado de Comandos

Comando	Respuesta	Función
AT - Probar iniciación correcta		
AT	OK	Prueba si el módulo responde correctamente
AT+RST - Reinicia el módulo		
AT+RST	OK	Resetéa el módulo
AT+CWMODE - Modo Wifi		
AT+CWMODE=?	+CWMODE:(1-3) OK	Lista los modos validos
AT+CWMODE?	+CWMODE:modo OK	Pregunta en que modo AP esta actualmente el módulo
AT+CWMODE=modo	OK	Establece el módulo en el modo dado 1 = Modo estación (cliente) 2 = Modo AP (huésped) 3 = Modo AP + Estación (modo dual)
AT+CWLAP - Lista APs disponibles		
AT+CWLAP	AT+CWLAP:ecn,ssid,rssi,mac OK	Lista los Acess Points disponibles para conectarse. ecn: codificación, puede ser: 0 = Abierto 1 = WEP 2 = WPA PSK 3 = WPA2 PSK 4 = WPA WPA2 PSK ssid: String que contiene el SSID del AP rssi: Fuerza de la señal mac: String que contiene la dirección MAC
AT+CWLAP=ssid,mac,ch	+CWLAP:ecn,ssid,rssi,mac OK	Busca Acess Points disponibles para conectarse con las condiciones especificadas
AT+CWJAP - Unirse a un Access Point		
AT+CWJAP?	+ CWJAP:ssid OK	Imprime el SSID al que el módulo esta conectado
AT+CWJAP=ssid,pwd	OK	El módulo se conecta a la red con el nombre ssid indicado y la contraseña pwd suministrada
AT+CWQAP - Desconectarse de una Access Point		
AT+CWQAP	OK	Se desconecta de la red que esta actualmente conectado
AT+CWSAP - Configurar el softAP del módulo		
AT+CWSAP?	+CWSAP:ssid,pwd,ch,ecn OK	Pregunta la configuración actual del softAP
AT+CWSAP=ssid,pwd,ch,ecn	OK	Configura el softAP con ssid: String con el nombre de la red pwd: Contraseña, no mayor a 64 caracteres ch: Canal inalámbrico ecn: Tipo de codificación 1 = Abierto 2 = WPA_PSK 3 = WPA2_PSK 4 = WPA_WPA2_PSK



AT+CIPSTATUS - Información acerca de la conexión		
AT+CIPSTATUS	STATUS:status +CIPSTATUS:id,type,addr,port,tetype OK	status: 2 = Se obtuvo IP 3 = Conectado 4 = Desconectado id: ID de la conexión en caso de multiconexión (1-4) type: Tipo de conexión, "TCP" o "UDP" addr: Dirección IP de la conexión port: Número del puerto tetype: 0 = El módulo corre como cliente 1 = El módulo corre como servidor
AT+CIPMUX - Habilitar o deshabilitar múltiples conexiones		
AT+CIPMUX=mode	OK	mode: 0 = Conexión única 1 = Múltiples conexiones, hasta 4
AT+CIPMUX?	+CIPMUX:mode OK	Imprime el mode, el modo de conexión actual
AT+CIPSTART - Establece una conexión TCP o registra un puerto UDP e inicia la conexión		
AT+CIPSTART=type,addr,port	OK	Empieza una conexión como cliente (en modo conexión única) type: puede ser "TCP" o "UDP" addr: String que contenga la dirección IP remota port: String que contenga el puerto remoto
AT+CIPSTART=id,type,addr,port	OK	Empieza una conexión como cliente (En modo conexión múltiple) id: ID de la conexión (1-4)
AT+CIPSTART=?	[+CIPSTART:(id)("type"),("ip address"), (port)] OK	Lista los posibles comandos
AT + CIPCLOSE - Cierra la conexión TCP o UDP		
AT+CIPCLOSE=?	OK	
AT+CIPCLOSE=id	OK	Cierra la conexión TCP o UDP con el ID "id" en modo conexión múltiple
AT+CIPCLOSE	OK	Cierra la conexión TCP o UDP para modo de conexión única
AT+CIPSEND - Enviar datos		
AT+CIPSEND=?	OK	
AT+CIPSEND=length	SEND OK	Establece la longitud de datos a enviarse (máximo 2048). Para un envío normal (modo conexión única)
AT+CIPSEND=id,length	SEND OK	Establece la longitud de datos a enviarse en la conexión número "id". Para un envío normal (modo conexión múltiple)
AT+CIPSEND		Envía datos sin adornos cada 20ms. El módulo retorna ">" después de ejecutar el comando, si se recibe el comando "+++" se regresa al modo comando.
AT+CIFSR - Obtener la dirección IP local		
AT+CIFSR=?	OK	
AT+CIFSR	+CIFSR:ip OK	Retorna la dirección IP local del módulo como cliente.
AT+CIFSERVER - Configurar como servidor		
AT+CIPSERVER=mode[,port]	OK	Configura el módulo como servidor donde el modo: 0 = Borrar servidor 1 = Crear servicio puerto: número del puerto, por defecto es el 333

AT+CI0BAUD Cambiar la velocidad de transmisión serial		
AT+CI0BAUD=?	+CI0BAUD:(9600-921600) OK	Nos informa que las velocidades de transmisión permitidas están en este rango
AT+CI0BAUD?	+CI0BAUD:baudrate OK	Nos indica que el módulo está actualmente configurado a 'baudrate'
AT+CI0BAUD=baudrate	OK	Configura la velocidad de transmisión a 'baudrate'

Tabla 20: Comandos ESP-8266

Fuente para Protoboard

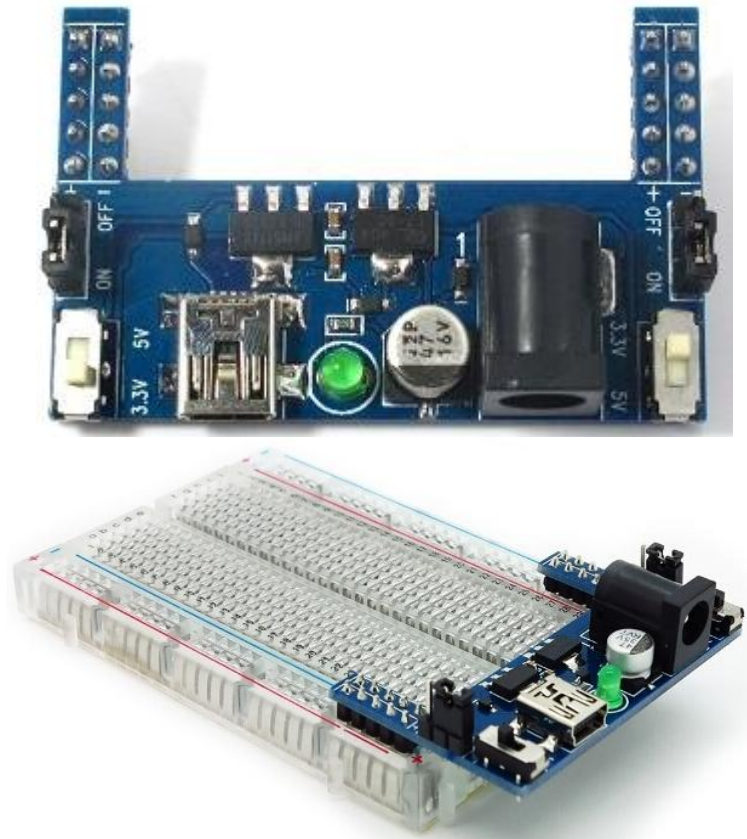


Imagen 48: Fuente de Alimentación

Esta fuente tiene 2 salidas independientes una de la otra para cada línea de alimentación del protoboard, una de 5V y otra de 3.3V. Un switch para intercalar entre ambas y un jumper para anularlas. Está basada en el regulador de tensión AMS1117. Tiene 2 opciones de alimentación, una por Jack y otra por mini USB.

La entrada de tensión por Jack debe ser de entre 7V y 12V, positivo al centro. Por USB obviamente va a ser de 5V. Esta placa puede entregar hasta 1A de corriente, y puede trabajar en rangos de temperatura muy amplios desde -40° y 150°

Sensor de ultrasonidos – HC-SR04



Imagen 49: Sensor Ultrasónico HC-SR04

Este módulo posee 4 pines (**Vcc**, **GND**, **Trig** y **Echo**). Los pines de alimentación van conectados a los pines GND y 5V de nuestro Arduino, y los pines Trig y Echo irán conectados a cualquier pin digital. Para medir distancia, lo que hace este módulo es enviar un ultrasonido (que no se puede percibir con el oído humano), a través del ping Trig y cronometrar el tiempo que tarda en rebotar en el obstáculo más cercano y ser recibido por el pin Echo; debemos tener en cuenta, que la velocidad del sonido es de 340m/s.

El HC-SR04 es un sensor de distancias por ultrasonidos, capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos, y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo, como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, el HC-SR04 se destaca por su bajo consumo, gran precisión y bajo precio.

De fácil uso y programación con las placas de Arduino y microcontroladores.

Características

- Dimensiones del circuito: 43 x 20 x 17 mm
- Tensión de alimentación: 5 Vcc
- Frecuencia de trabajo: 40 KHz
- Rango máximo: 4.5 m
- Rango mínimo: 1.7 cm
- Duración mínima del pulso de disparo (nivel TTL): 10 μ S.
- Duración del pulso eco de salida (nivel TTL): 100-25000 μ S.
- Tiempo mínimo de espera entre una medida y el inicio de otra 20 mS.

Pines de conexión

- VCC
- Trig (*Disparo del ultrasonido*)

- Echo (*Recepción del ultrasonido*)
- GND

Distancia = {(Tiempo entre Trig y el Echo) * (V.Sonido 340 m/s)}/2

Sensor de Color - TCS3200



Imagen 50: Sensor de Colores TCS3200

El TCS3200 es un convertidor de color a frecuencia, en un circuito integrado CMOS. La salida es una onda cuadrada (ciclo de trabajo del 50%), con una frecuencia directamente proporcional a la intensidad de la luz (irradiación). La frecuencia de salida, a escala completa, se puede escalar por uno de tres valores preestablecidos a través de dos pines de entrada de control. La habilitación de salida (OE) coloca la salida en el estado de alta impedancia, para el intercambio de unidades múltiples de una entrada del microcontrolador.

La escala de frecuencia de salida, se puede manejar mediante los dos pines S0 y S1 (*Selectable Options 2%, 20%, 100% frequency*). Los pines S2, S3 controlan el filtro RGB. Todos los pines digitales I/O se pueden conectar directamente al micro, ya que trabajan en el mismo voltaje. *Output enable* (OE) coloca la salida en estado de alta impedancia, para el intercambio de unidades múltiples de una línea de entrada del microcontrolador. Por último, el usuario puede calcular el color de la luz por los valores RGB.

Características

- El TCS3200 es TCS230 actualizado mejor versión.
- 3-5v Alimentación.
- Resistencia a la interferencia de la luz.
- LED blanco se puede controlar encendido, apagado.
- Puede detectar el color del objeto no luminoso.
- Mejor detección de distancia de 1 cm.

RGB (sigla en inglés de *red*, *green*, *blue*, en español «rojo, verde y azul») es la composición del color en términos de la intensidad de los colores primarios de la luz.

Uso de RGB en HTML y en lenguajes de programación

Para indicar con qué proporción es mezclado cada color, se asigna un valor a cada uno de los colores primarios, de manera que el valor "0" significa que no interviene en la mezcla y, a medida que ese valor aumenta, se entiende que aporta más intensidad a la mezcla. Aunque el intervalo de valores podría ser cualquiera (valores reales entre 0 y 1, valores enteros entre 0 y 37, etc.), es frecuente que cada color primario se codifique con un byte (8 bits).

Así, de manera usual, la intensidad de cada una de las componentes, se mide según una escala que va del 0 al 255 y cada color, es definido por un conjunto de valores escritos entre paréntesis (correspondientes a valores "R", "G" y "B") y separados por comas.

De este modo, el rojo se obtiene con (255, 0, 0), el verde con (0, 255, 0) y el azul con (0, 0, 255), obteniendo, en cada caso un color resultante monocromático. La ausencia de color, es decir el color negro, se obtiene cuando las tres componentes son 0: (0, 0, 0). La combinación de dos colores a su máximo valor de 255 con un tercero con valor 0 da lugar a tres colores intermedios. De esta forma, aparecen los colores amarillo (255, 255, 0), cian (0, 255, 255) y magenta (255, 0, 255). El color blanco se forma con los tres colores primarios a su máximo valor (255, 255, 255).

Servo



Imagen 51: Servo

Normalmente los motores habituales, lo que hacen es transformar la energía eléctrica (*O química*) en un giro continuo, que podemos usar para desarrollar trabajo mecánico.

Los servos son también motores de corriente continua, pero en lugar de diseñarse para obtener un giro continuo que podamos aprovechar (para mover una rueda por ejemplo), se diseñan para que se muevan un ángulo fijo en respuesta a una señal de control, y se mantengan fijos en esa posición.

Estos servos o servomotores son muy frecuentes en robótica, por la capacidad que presentan para moverse a un ángulo concreto y mantenerse allí. De hecho se suelen diseñar



para que giren un ángulo proporcional a una señal PWM, de forma que su control es muy preciso.

Un servo suele estar formado por:

- Una reductora. Un circuito de control.
- Un motor de CC.
- Un circuito de control

En la práctica se comporta como un bloque funcional, que posiciona su eje en un ángulo preciso en función de la señal de control

Habitualmente los servos tiene un margen de operación, es decir, pueden moverse entre 0° y ángulo dado, que suele ser de 180°, pero existen modelos comerciales de todas la características imaginables (incluyendo servos de 360°).

Normalmente estos pequeños servos funcionan sobre 5V, y el control se realiza mediante una señal de control PWM, en la que el ancho el pulso indica el ángulo que deseamos adopte el eje.

Un servo tiene un conector de 3 hilos, 5V (rojo), GND (negro o marrón) y el otro Control (amarillo o blanco).

Podemos manejar un servo sin preocuparnos para nada de cómo realizar el control de las señales, o de la operación interna, ya que se cuenta con librerías que se instalan para obtener los programas y que los servos funcionen de manera correcta.

- Una librería es un conjunto de definiciones y funciones escritas en C++ de Arduino, que podemos utilizar en nuestros programas. Podemos ver el código de estos programas, aunque normalmente se requiere un nivel medio alto en C++ para comprender su funcionamiento.

La ventaja de trabajar con una plataforma tan bien establecida como Arduino, es que hay cantidad de gente en el mundo desarrollando librerías.

Heroku

Heroku es una plataforma como servicio de computación en la Nube, que soporta distintos lenguajes de programación.

Heroku es propiedad de Salesforce.com. Heroku, es una de las primeras plataformas de computación en la nube, que fue desarrollada desde junio de 2007, con el objetivo de soportar solamente el lenguaje de programación Ruby, pero posteriormente se ha extendido el soporte a Java, Node.js, Scala, Clojure y Python y (no documentado) PHP. La base del sistema operativo es Debian o, en la nueva plataforma, el sistema basado en Debian Ubuntu.



Dynos

Los Dynos son piezas fundamentales del modelo de arquitectura de Heroku, son las unidades que proveen capacidad de cómputo dentro de la plataforma. Están basados en Contenedores Linux.

Cada Dyno está aislado del resto, por lo que los comandos que se ejecutan y los archivos que se almacenan en un Dyno, no afectan a los otros. Además cada Dyno provee el ambiente requerido por las aplicaciones para ser ejecutadas.

Los posibles comandos a ser ejecutados en los Dynos incluyen procesos web, o cualquier otro tipo de proceso definido en el archivo Procfile de la aplicación. Este es un archivo de texto ubicado en el directorio raíz de la aplicación, y es el mecanismo provisto para la declaración de comandos que luego correrán los Dynos. Básicamente, consiste de una lista de tipos de procesos de la aplicación. Cada tipo de procesos constituye una declaración de un comando.

Principales características

- Elasticidad y crecimiento: La cantidad de Dynos asignados a una aplicación se puede cambiar en cualquier momento a través de la línea de comandos o el *dashboard*.
- Tamaño: Heroku ofrece diferentes tipos de Dynos, cada uno con diferentes capacidades de procesamiento y memoria.
- Routing: Internamente los routers realizan un seguimiento de la ubicación de los Dynos que estén corriendo, y redirigen el tráfico de acuerdo a la misma.
- Seguimiento: Existe un manejador de Dynos, el cual monitorea de forma continua los Dynos que se estén ejecutando. En caso de una falla en un Dyno, este es eliminado y creado nuevamente.
- Distribución y redundancia: Los Dynos se encuentran aislados uno de otros. Esto implica que de existir fallos en la infraestructura interna de alguno de ellos, los otros Dynos no se ven afectados, y consecuentemente tampoco la aplicación.



	Free \$0	Hobby \$7/dyno per month	Standard 1x \$25/dyno per month	Standard 2x \$50/dyno per month	Performance M \$250/dyno per month	Performance L \$500/dyno per month
			PROFESSIONAL			
What is it good for?	Ideal for experimenting with cloud applications in a limited sandbox.	Perfect for small scale personal projects and hobby apps.	Enhanced visibility, performance, and availability for powering your production applications.		Superior performance when it's most critical for your super scale, high traffic apps.	
Deploy from Git	•	•	•	•	•	•
Automated patching	•	•	•	•	•	•
Self healing apps	•	•	•	•	•	•
Unified logs	•	•	•	•	•	•
Number of process types	2	10	Unlimited	Unlimited	Unlimited	Unlimited
Always on	Must sleep 6 hours a day	•	•	•	•	•
Custom domains	•	•	•	•	•	•
"Slide to scale" out			•	•	•	•
Preboot			•	•	•	•
Application metrics			•	•	•	•
Faster builds			•	•	•	•
Mix and match			Mix with 2x and Performance M/L	Mix with 1x and Performance M/L	Mix with Standard 1x/2x and Performance L dynos	Mix with Standard 1x/2x and Performance M dynos
RAM	512MB	512MB	512MB	1GB	2.5GB	14GB
Dedicated	No	No	No	No	Yes	Yes

Imagen 52: Alternativas de Dynos

Seguridad (SSL)

Heroku brinda el soporte necesario para poder otorgar, a todas las aplicaciones generadas y *hosteadas* en el mismo un sistema de encriptación, para fortalecer la seguridad de las transacciones entre *client-side* y *server-side*. Nos presenta esta característica como un AddOn (complemento), el cual se puede adquirir fácilmente desde el dashboard de cada usuario.

La instalación de esta función se realiza de la siguiente manera:

1. Adquirir un certificado SSL.

- a. Generar la clave privada.

Antes de solicitar un certificado SSL necesitamos generar nuestra clave privada en nuestro ambiente local. Para esto utilizamos OpenSSL, aplicación que está incluida en el paquete de instalación de Ubuntu, pero que debemos actualizar en caso de no poseer la última versión de la misma. Si se posee la última versión, lo único que debemos hacer para generar la clave es escribir el siguiente comando en consola:

```
openssl genrsa -des3 -out server.pass.key 2048
```

La contraseña debe ser separada de la clave privada, para no generar la necesidad de introducirla manualmente. Por esto corremos el siguiente comando:



```
openssl rsa -in server.pass.key -out server.key
```

Una vez realizados estos pasos, ya disponemos de nuestra server.key alojada en nuestro servidor.

b. Generar el CSR

El CSR es un certificado de petición de logueo, que es requerido cuando vamos a adquirir nuestro SSL. Usando la clave privada generada en el paso anterior, procedemos a generar el CSR. Esto requiere que ingresemos información personal, como ser el nombre de nuestra organización.

```
openssl req -nodes -new -key server.key -out server.csr
```

El resultado de esta operación, será un server.csr en nuestro directorio local.

c. Enviar el CSR al proveedor de SSL

Ahora comenzamos el proceso de crear nuestro propio certificado SSL con el proveedor elegido. El proceso dependerá mucho del proveedor, pero en general lo que debemos hacer es enviarle nuestro CSR

2. Crear el AddOn

- a. Este paso debemos ejecutarlo ya que nuestro servidor es de acceso público. Debemos agregar el servicio que provee Heroku desde consola con el siguiente comando.

```
heroku addons:create ssl:endpoint
```

3. Añadir certificado

- a. Añadimos nuestro certificado con el siguiente comando en consola.

```
heroku certs:add server.crt server.key
```

Una vez realizados todos los pasos anteriores, dispondremos de un certificado SSL. A fines de este proyecto, se optó por no incluir en el alcance la implementación de un certificado SSL, aunque si la investigación del mismo (detallada anteriormente). El SSL es fundamental a la hora del manejo de tarjetas de crédito, si bien es una característica recomendable a todas las aplicaciones webs con acceso público, en el caso del manejo de datos sensibles es esencial.

MercadoPago, en su API, obliga a sus developers (desarrolladores) a utilizar un certificado SSL para la puesta en producción de las transacciones. Esta regla no se aplica para la etapa de homologación-testeo. Por este motivo también es primordial, la aplicación de este certificado a un producto con salida al mercado real.



Metodología para la gestión del proyecto

La metodología utilizada en el proyecto final es la brindada por PMI (*Project Management Institute*), la cual es una de las principales organizaciones internacionales sin fines de lucro, que asocia a profesionales relacionados con la Gestión de Proyectos.

La guía de fundamentos para la dirección de proyectos (Guía del PMBOK), es una norma reconocida en la profesión de la dirección de Proyectos. Por norma se hace referencia a un documento formal que describe normas, métodos, procesos y prácticas establecidos. La misma evolucionó a partir de las buenas prácticas reconocidas por profesionales dedicados a la dirección de proyectos, quienes contribuyeron a su desarrollo.

¿Por qué utilizar PMBOK?

- Porque los procesos que marca el PMBOK, al igual que cualquier otro enfoque estándar como PRINCE2 o ISO21500, han sido descritos como resultado del trabajo y experiencia que aporta un gran número de profesionales de la gestión de proyectos.
- Adaptable, no es necesario que la empresa cambie para amoldarse a un enfoque como PMBOK, sino que éste tiene la suficiente flexibilidad como para adaptarse a la empresa, pudiendo seleccionar los procesos a aplicar, el modo de hacerlo, las técnicas concretas a utilizar, etc.
- Hablar todos el mismo idioma. Esto facilitará la comunicación con el exterior en el caso de proyectos multi-empresa y también de cara al cliente, ya que es muy probable que fuera conozcan las prácticas estándar como las que propone el PMI en su guía PMBOK, pero por el contrario es casi seguro que no conocerán nuestro método interno.
- Puede facilitar el periodo de adaptación de nuevas incorporaciones al equipo, ya que es probable que ya lo conozcan.

Esta metodología divide a la gestión del proyecto, en nueve áreas de conocimiento claves. A continuación, en la imagen 53, se observa la relación de las distintas áreas de conocimiento para poder cumplir con el objetivo propuesto en el proyecto.

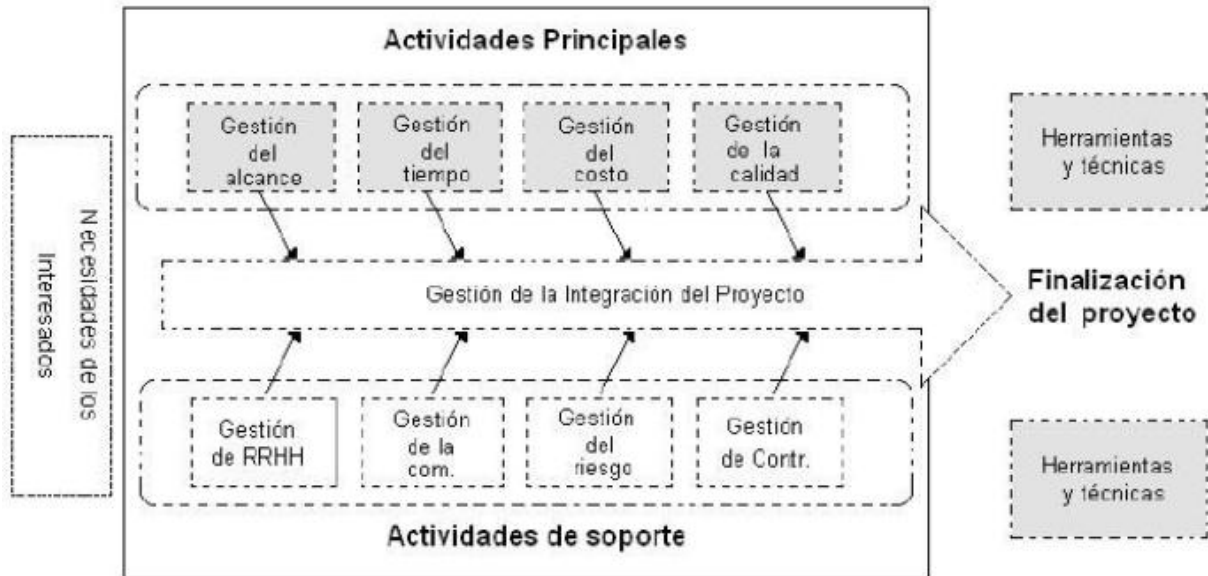


Imagen 53: Áreas de Conocimiento PMBOK

A continuación, se hace una breve explicación de cada una de ellas:

- Área de Integración: Es un área que es influida por las otras ocho áreas de conocimiento. Es la que se encarga de coordinar a todas las demás dentro del desarrollo del proyecto. Implica tomar decisiones en cuanto a la asignación de recursos, balancear objetivos y alternativas contrapuestas, y manejar las interdependencias entre las áreas de conocimiento de la dirección de proyectos.
- Gestión del Alcance del Proyecto: Es un área que incluye los procesos necesarios para garantizar que el proyecto incluya todo (y únicamente todo) el trabajo requerido para completarlo con éxito. El objetivo principal, es definir y controlar que se incluye y que no se incluye en el proyecto.

Se deben recopilar requisitos de los interesados, a fin de cumplir con los objetivos del proyecto, desarrollar una descripción detallada del proyecto y el producto, crear la EDT (Estructura de desglose de Trabajo) junto a su diccionario, verificar los entregables y controlar dicho alcance para monitorear el estado del alcance del proyecto.

- Gestión del Tiempo del Proyecto: Es un área que incluye los procesos para administrar la finalización del proyecto a tiempo. Se deben definir las actividades a realizar en el proyecto y secuenciarlas mediante interrelaciones. A su vez, se deben estimar los recursos y duraciones de las actividades, para finalmente desarrollar el cronograma y realizar un control del mismo mediante el seguimiento al estado del proyecto.
- Gestión de los Costos del Proyecto: Es un área que incluye los procesos involucrados en estimar, presupuestar y controlar los costos, de modo que se complete el proyecto dentro del presupuesto aprobado.



- Gestión de la Calidad del Proyecto: Es un área que incluye los procesos y actividades que determinan responsabilidades, objetivos y políticas de calidad, a fin de que el proyecto satisfaga las necesidades para las cuales fue emprendido.
- Gestión de los Recursos Humanos del Proyecto: Es un área que incluye los procesos que organizan y conducen el equipo del proyecto. El equipo de proyecto, está conformado por aquellas personas a las que se les han asignado roles y responsabilidades para completarlo.

Se identifican y documentan los roles dentro del proyecto, las responsabilidades y las relaciones de comunicación. Se confirman los recursos humanos disponibles y se forma el equipo, para completar con las actividades del proyecto. Dicho equipo debe mejorar las competencias, interacción y el ambiente, para lograr un mejor desempeño. Para ello, debe haber un seguimiento de los miembros del equipo, para la resolución de problemas y gestionar cambios.

- Gestión de las Comunicaciones del Proyecto: Es un área que incluye los procesos requeridos para garantizar que la generación, recopilación, distribución, almacenamiento, recuperación y disposición final de la información del proyecto sean adecuados y oportunos.

Estas comunicaciones pueden ser:

- Internas (dentro del proyecto) y externas (cliente, otros proyectos, medios de comunicación, público).
 - Formal (informes, instrucciones) e informal (correos electrónicos, conversaciones ad hoc).
 - Escrita y oral
 - Verbal y no verbal (inflexiones de voz, lenguaje corporal)
- Gestión de los Riesgos del Proyecto: Es un área que incluye los procesos relacionados para llevar a cabo la planificación de la gestión, identificación, análisis, respuesta a los riesgos, como así también su monitoreo y control. Se debe tratar de aumentar la probabilidad e impacto de eventos positivos, y disminuir la probabilidad y el impacto de eventos negativos para el proyecto.
 - Gestión de las Adquisiciones del Proyecto: Es un área que incluye los procesos de compra o adquisición de los productos, servicios o resultados que son necesarios obtener fuera del equipo de proyecto.

Análisis de Porter

El análisis de las cinco fuerzas de Porter, es un modelo estratégico que establece un marco para analizar el nivel de competencia dentro de una industria, y poder desarrollar una estrategia de negocio. Este análisis deriva en la respectiva articulación de las 5 fuerzas que determinan la intensidad de competencia y rivalidad de una industria, y por lo tanto, en cuanto atractiva es esta industria en relación a oportunidades de inversión y rentabilidad.

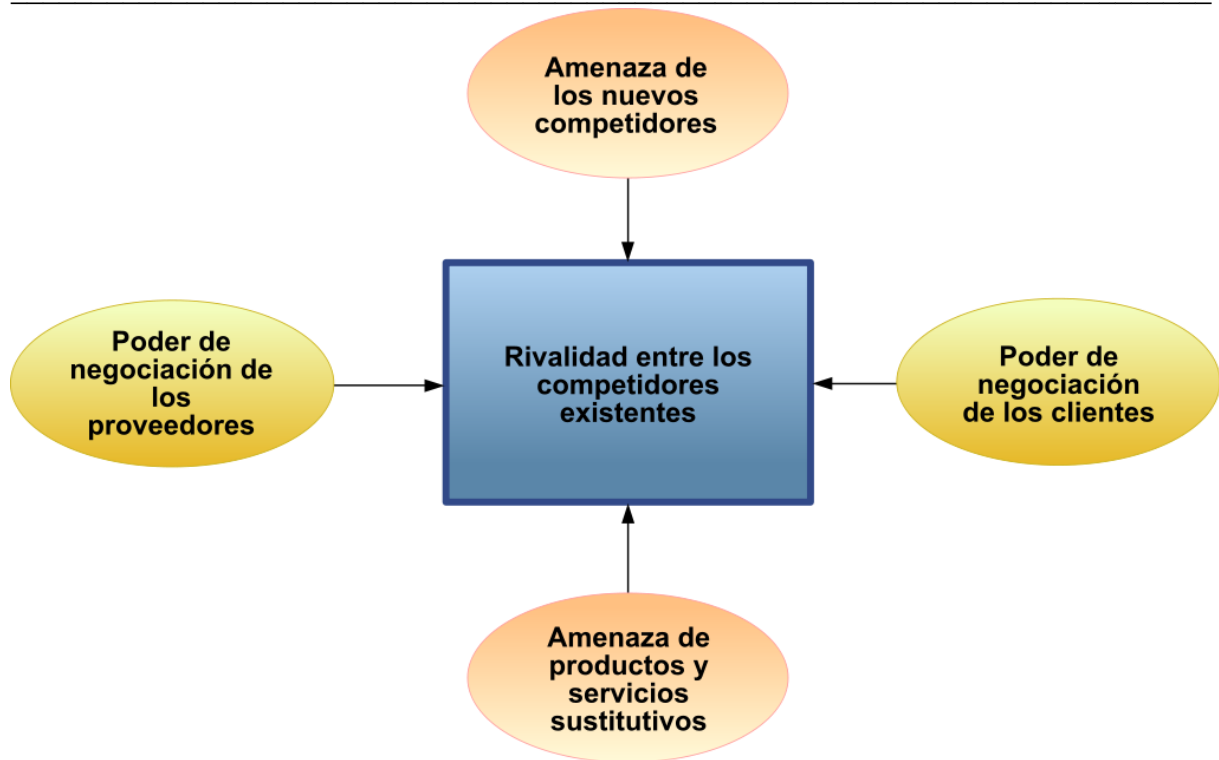


Imagen 54: Esquema de las 5 fuerzas de Porter

A continuación, se hace una breve explicación de cada una de las fuerzas de Porter:

- Poder de negociación de los Compradores o Clientes: se tienen problemas cuando los clientes cuentan con un producto que tiene varios sustitutos en el mercado, o que puede llegar a tener un costo más alto que otros productos si tu producto llega a tener un costo más alto que otros similares en el mercado. Si los compradores están bien organizados, esto hace que sus exigencias sean cada vez más altas, incluso una reducción de precios notable.
- Poder de negociación de los Proveedores: se refiere a una amenaza impuesta sobre la industria por parte de los proveedores a causa del poder que estos disponen, ya sea por su grado de concentración, por las características de los insumos que proveen, por el impacto de estos insumos en el costo de la industria, etc.
- Amenaza de productos y servicios sustitutos: una empresa comienza a tener serios problemas, cuando los sustitutos de los productos comienzan a ser reales, eficaces y



más baratos que el que vende la empresa inicial. Esto hace que dicha empresa tenga que bajar su precio, lo que lleva una reducción de ingresos en la empresa.

- Amenaza de los nuevos competidores: Es una de las fuerzas más famosas y que se usa en la industria, para detectar empresas con las mismas características económicas o con productos similares en el mercado.

A su vez, para las fuerzas (Amenazas de productos y de nuevos competidores), se tienen una serie de barreras de entrada que son muy útiles si se usan de una forma correcta.

- La economía de escalas: Los volúmenes altos en las empresas permiten que los costos se reduzcan, lo que ofrece la posibilidad de volver a ser competitivos en el mercado.
- La diferenciación de productos: Si se es capaz de posicionar el producto claramente en el mercado ofreciendo algo diferente, se puede revalorizar ante los ojos de los compradores, buscando tu producto al verlo de mejor calidad y buscar una mejor calidad en sus productos.
- Las inversiones de capital: En caso de problemas, la empresa puede mejorar su posición con una inyección de capital en sus productos, lo que puede hacer que sobreviva ante empresas más pequeñas similares.
- Desventaja de Costos: Esta barrera juega a nuestro favor, cuando las otras empresas no pueden emular el precio de nuestros productos por que cuentan con costos más elevados.
- Acceso a los Canales de Distribución: Cuando una empresa cuenta con varios canales de distribución, es complicado que puedan aparecer competidores y sobre todo que los proveedores acepten el producto. Esto implicaría para las empresas, tener que compartir costos de promoción de distribución y reducción de precios en general.



- Política Gubernamental: Este punto puede jugar a tu favor, ya que en muchos puntos, las políticas gubernamentales son las que impiden la llegada de nuevos competidores en todos los sentidos. Esto está regulado por leyes muy estrictas.
- Rivalidad entre los competidores existentes: se puede competir directamente con otras empresas de la industria que te dan el mismo producto. La rivalidad se caracteriza por que los competidores están enfrentados y que ambos usen grandes estrategias de negocios. Esta rivalidad puede dar como resultado:
 - Cantidad de competidores más grandes y que todos estén equilibrados.
 - Que el crecimiento de la industria sea mucho más lento.
 - Que los costos y almacenamiento sean más elevados.
 - Que el producto no llegue a los clientes o no pueda diferenciar realmente su utilidad.
 - Que se tengan que buscar nuevas estrategias con costos mucho más elevados.
 - Que el mercado se sature.
 - Que existan competidores muy diversos.

ANMAT

La Administración Nacional de Medicamentos, Alimentos y Tecnología Médica (ANMAT), es un organismo descentralizado de la Administración Pública Nacional creado en agosto de 1992, mediante decreto 1490/92.

Colabora en la protección de la salud humana, garantizando que los medicamentos, alimentos y dispositivos médicos a disposición de los ciudadanos, posean eficacia (que cumplan su objetivo terapéutico, nutricional o diagnóstico) seguridad (alto coeficiente beneficio/riesgo) y calidad (que respondan a las necesidades y expectativas de la población).

Para ello, se encarga de llevar adelante los procesos de autorización, registro, normatización, vigilancia y fiscalización de los productos de su competencia en todo el territorio nacional.

La ANMAT depende técnica y científicamente de las normas y directivas que le imparte la Secretaría de Políticas, Regulación e Institutos del Ministerio de Salud, con un régimen de autarquía económica y financiera.



Una de las dependencias de la ANMAT es la **Dirección de Tecnología Médica**, la cual se encarga de controlar, fiscalizar y vigilar la calidad de los productos médicos como así también las actividades, procesos y tecnologías que se realicen en función de la elaboración, fraccionamiento, importación o exportación y depósito de dichos productos, a fin de asegurar a la población la utilización de productos médicos de comprobada calidad, eficacia y seguridad.