

Inclusión de las personas no videntes al ámbito de la programación



Retamoso Granados, Lucas Lautaro

Ingeniería en Informática

2018

Inclusión de las personas no videntes al ámbito de la programación.

Profesores Guías:

Lic. Patricia Gabriela Aballay

Tribunal Evaluador:

Fecha de Presentación: ____ de _____ del 2018

Dedicatorias y Agradecimientos

Dedico y agradezco este trabajo a:

Dios por su fuerza otorgada en todo este trayecto, brindándome paciencia y energías para realizar este trabajo.

A mi profesora guía, la licenciada Patricia Aballay, por la paciencia y ser una profesional en la que confió plenamente en su criterio para que este proyecto.

A la universidad, por brindarme las herramientas necesarias para realizar el trabajo y brindarme su apoyo en todo momento.

A mis compañeros, a quienes ya considero mis amigos, por ayudarme en momento de duda y sacar lo mejor de mí en todo momento.

Por ultimo pero no menos importante, a mis familiares, pilares de mi salud y bienestar, ayudándome a mantenerme enfocado y apoyándome en todo momento, ya sea mi madre en su apoyo y contención, en mi padre su fe y perseverancia, en mis hermanos siendo esenciales en contención y, opinión sobre el proyecto y su finalidad.

Contenido

Introducción.....	6
Al Problema	6
Al Proyecto	7
Motivación y Restricciones	7
Conceptos Previos	10
Personas no videntes	10
Programación y sus beneficios	12
Integración de conceptos	15
Especificaciones del proyecto	17
Contexto de aplicación.....	17
Objetivo General	17
Objetivos Específicos	17
Alcance	17
Riesgos.....	18
Metodológicas para la realización	20
Proceso Unificado Racional.....	20
¿Por qué el Proceso Unificado Racional?	20
Gestión del Proyecto	23
Solución.....	24
Metodología en acción.....	24
Alternativas tecnológicas de solución del problema.....	32
Arquitectura de procesamiento y comunicación de la propuesta tecnológica	33
Ingeniería del proyecto	33
Matriz de responsabilidades	34
Asignación de Recursos y Tiempo.....	35
Análisis Económico-Financiero	36
Gestión de Riesgos.....	37
Factibilidad	39
Técnica.....	39
Operativa.....	40
Legal	40
Económica-Financiera	41
Ambiental.....	42
Conclusión y Futuras Líneas de Investigación	42

Bibliografía	44
Anexo I	46
Diagrama de Flujo	46
Modelado del Diseño	46
Diagrama de Clases	46
Diagrama de estado-transición	48
Diagrama de Actividades	58
Diagrama de Componentes	59

Índice de Tablas

TABLA 1 - CARRERAS.....	15
TABLA 2 - TABLA DE RIESGOS	18
TABLA 3 - DIAGRAMA DE CLASE - COMPOSICIÓN FUERTE Y DÉBIL	26
TABLA 4 - DIFERENCIA ENTRE COMPOSICIÓN FUERTE Y DÉBIL	27
TABLA 5 - ELEMENTOS DEL DIAGRAMA DE ESTADOS-TRANSICIONES	28
TABLA 6 – ELEMENTOS BÁSICOS DEL DIAGRAMA DE ACTIVIDADES	29
TABLA 7 - ELEMENTOS NECESARIOS DEL DIAGRAMA DE ACTIVIDADES	31
TABLA 8 - ALTERNATIVAS TECNOLÓGICAS.....	32
TABLA 9 - MATRIZ DE RESPONSABILIDADES.....	34
TABLA 10 - RECURSOS HUMANOS DEL PROYECTO.....	37
TABLA 11 - RECURSOS MATERIALES DEL PROYECTO.....	37
TABLA 12 - RECURSOS DE DESARROLLO.....	37
TABLA 13 - GESTIÓN DE RIESGOS	38

Índice de Imágenes

IMAGEN 1 - LOGO PROGRAMA ORCA	7
IMAGEN 2 - LOGO PROGRAMA NARRADOR DE WINDOWS	7
IMAGEN 3 - PROCESO UNIFICADO RACIONAL.....	22
IMAGEN 4 - ESTRUCTURA DEL DIAGRAMA DE CLASE.....	26
IMAGEN 5 - ESTRUCTURA DEL DIAGRAMA DE CLASE CON VINCULO Y CARDINALIDAD.....	26
IMAGEN 6 - ESTRUCTURA DEL DIAGRAMA DE COMPONENTES	32
IMAGEN 7 - VISIÓN GENERAL DE LOS RECURSOS	35
IMAGEN 8 - VISIÓN GENERAL DE LOS TIEMPOS	36
IMAGEN 9 - DIAGRAMA DE FLUJOS.....	46
IMAGEN 10 - DIAGRAMA DE CLASES.....	47
IMAGEN 11 - DIAGRAMA DE ESTADO-TRANSICIÓN CLASE CAPTURADOR	48
IMAGEN 12 - DIAGRAMA DE ESTADO-TRANSICIÓN CLASE CONCATENADOR	49
IMAGEN 13 - DIAGRAMA DE ESTADO-TRANSICIÓN CLASE ERROR_CATCH.....	50
IMAGEN 14 - DIAGRAMA DE ESTADO-TRANSICIÓN CLASE ERROR	51
IMAGEN 15 - DIAGRAMA DE ESTADO-TRANSICIÓN CLASE ERRORES.....	52
IMAGEN 16 - DIAGRAMA DE ESTADO-TRANSICIÓN CLASE ERROR_FOR	53
IMAGEN 17 - DIAGRAMA DE ESTADO-TRANSICIÓN CLASE ERROR_IF.....	54
IMAGEN 18 - DIAGRAMA DE ESTADO-TRANSICIÓN CLASE ERROR_SWITCH	55
IMAGEN 19 - DIAGRAMA DE ESTADO-TRANSICIÓN CLASE TTS	56
IMAGEN 20 - DIAGRAMA DE ESTADO-TRANSICIÓN CLASE ERROR_WHILE	57
IMAGEN 21 - DIAGRAMA DE ACTIVIDADES	58
IMAGEN 22 - DIAGRAMA DE COMPONENTES	59

Introducción

Al Problema

En la actualidad, en el mundo existen diversas complicaciones que afectan a las distintas personas en la labor de sus actividades o, en el peor de los casos, directamente no pueden realizar tales tareas. Las discapacidades son falta o limitación de alguna facultad física o mental que imposibilita o dificulta el desarrollo normal de la actividad de una persona, entre ellas se contemplan los sordos, minusválidos, el autismo, entre otras. En este proyecto, nos centraremos solamente en la discapacidad visual de las personas y como ayudarlas en un ámbito donde lo visual es importante.

Para empezar, vamos a mencionar a las personas a quienes vamos a ayudar, son todas las personas ciegas donde la visión es menor de 20/200 o 0.1 en el mejor ojo y con la mejor corrección, denominadas “ciegos legales” y que quieran estudiar una carrera universitaria donde la Programación es necesaria.

En la actualidad, las personas ciegas realizan tareas que antes les resultaban difíciles o incluso imposibles, como ser la lectura, para la cual se pudo dar apoyo e inclusión por el lenguaje Braille, otro ejemplo es en el uso de computadoras, donde para estas personas se crearon programas para que se pueda leer las pantallas y así poder interactuar con el ordenador, por ejemplo el software gratuito Orca o el narrador de Windows que están instadas en todos los nuevos Sistemas Operativos de la compañía.

Orca



Imagen 1 - Logo Programa Orca

Fuente: <http://portale.siva.it/en-GB/databases/products/detail/id-19185>

Narrador de Windows



Imagen 2 - Logo Programa Narrador de Windows

Fuente: <https://www.informatica-hoy.com.ar/trucos-consejos-computadora/Narrador-Windows-lee.php>

Aun así, con el apoyo que se les otorga para el manejo de máquinas personales, siguen siendo excluidos en muchos ámbitos dentro del mismo, como ser en la navegación web (los lectores de pantallas no pueden acceder a ellos de forma correcta) y en la programación. En lo que respecta a la programación, esta fue creciendo a pasos agigantados, pero volviéndose dependiente de lo visual y, por consecuencia, muy difícil para las personas que no pueden ayudarse con las ventajas que brindan los mismos.

Al Proyecto

Para atacar el problema antes planteado se tiene que pensar, como para los distintos IDE (Entorno de Desarrollo Integrado o “*Integrated Development Environment*” en sus siglas en inglés), se desarrollará una solución para que las personas no videntes puedan beneficiarse y no estar en desventajas comparadas con sus compañeros sin ninguna discapacidad visual en el ámbito educativo.

En consecuencia, este proyecto contempla una alternativa de solución para resolver, aunque sea parcialmente, el problema que ocasiona el impedimento de utilizar de manera sencilla las herramientas programática en un IDE determinado, atacando tal problema de una manera directa, trabajando prioritariamente la sintaxis y semántica a través de un plugin (se detalla que es un plugin en el capítulo 2, subcapítulo 2) y en conjunto con los distintos programas de lectura de pantalla, se podrán utilizar para disminuir (lamentablemente no eliminar) la línea de separación entre los videntes y no videntes.

Motivación y Restricciones

La motivación de este proyecto empieza por un grupo selecto de la argentina, donde la Universidad Tecnológica Nacional logró que descubra una necesidad aun no resuelta completamente.

En la Argentina, la cantidad de ciegos, según el Ministerio de Salud¹, en el 2016 es de 10.841 personas, donde 4.597 personas aproximadamente pueden acceder a estudios universitarios, ya que cumplen con la condición de haber completado los estudios secundarios y ser mayor de 18 años. Solamente se necesita un estudiante con estas condiciones para que la enseñanza de Ingeniería informática, en computación, en sistemas, entre otras, deje de ser del todo eficiente, ya que no tienen todas las herramientas, tanto pedagógicas como tecnológicas, para guiarlo de manera eficiente. La Universidad Tecnológica Nacional (UTN) de la provincia de Santa Fe, en la carrera de “Ingeniería en Sistemas de Información” se encontró con esta realidad, teniendo que buscar herramientas para poder enseñar al alumno de manera correcta y satisfactoria.

Las dificultades que encontraron en un principio fue en la parte práctica de la cátedra llamada “Algoritmos y Estructuras de Datos”, donde observaron que la persona ciega tenía la dificultad de ubicarse en el código, localizarlo de manera cómoda y especificar el código desarrollado. Esto los llevó a buscar herramientas de apoyo que les ayudara en la enseñanza y comprensión del estudiante, pero ninguna satisfacía las necesidades de su alumno, respecto a lo cual concluyeron en que “[...] existen muchas herramientas disponibles, la mayoría de ellas apunta a brindar soporte al proceso posterior al aprendizaje inicial”. Por lo tanto, decidieron realizar una propuesta de desarrollo, presentando una propuesta llamada “Propuesta de una Herramienta de Soporte para el Desarrollo de Habilidades de Programación en Estudiantes No Videntes” en el 4to Congreso Nacional de Ingeniería en Informática/Sistemas de Información (CoNaIISI) del año 2016, realizado en la Universidad Católica de Salta.

Asistiendo a la charla que realizó la Ing. María Julia Blas sobre la propuesta y leyendo posteriormente su trabajo, pude observar que todas las problemáticas que planteaban estaban siendo resueltas de forma poco cómoda para el estudiante ciego, donde realizarían las mejoras propuestas una vez que la persona ciega haya realizado el trabajo de programar.

Como una alternativa o como un posible trabajo en conjunto, propongo realizar de forma directa varios puntos de solución, para que la persona no vidente pueda realizar la actividad sin pasos extras y, por otro lado, agregarle la opción de indicarle que el bloque de sentencia que acaba de escribir se encuentra de forma errónea, elemento no contemplado en el trabajo de la UTN. En este sentido me comuniqué con la Ing. Blas, para conversar sobre el proyecto que realizaron y cómo fueron sus resultados, no solo me comentó que el proyecto está creciendo,

¹ Ministerio de Salud y Servicio Nacional de Rehabilitación, Anuario Estadístico Nacional.

además, le gustó mucho la propuesta y como se la realizaría, porque la herramienta con la que cuentan no tienen esas especificaciones con el TTS.

Trabajando en conjunto el plugin y la propuesta de la UTN de Santa Fe, mejorará la enseñanza de los alumnos que tienen que realizar las prácticas de programación, siendo en un principio para niveles de programación bajo y que se estén iniciando en el área.

Las restricciones que se encuentran están sobre la sociedad, que por algún motivo de prejuicios o de demanda, no incluyan a las personas ciegas en el ámbito educativo para la programación; los mismos actores (no videntes), donde pueden sentirse en un ambiente que no le beneficia en su totalidad y su discapacidad le frustra en la realización de la tarea; y en las instituciones educativas que tienen que brindar el mayor apoyo y contención a estos alumnos, para que puedan desarrollarse profesionalmente de forma correcta y sin ningún impedimento.

Por otro lado, con la herramienta que se utiliza para realizar el complemento puede presentar restricciones propias, siendo la más significativa la imposibilidad de agregar archivos ajenos al entorno de desarrollo, como ser archivos de fotos, audios, etc.

Conceptos Previos

Personas no videntes

Para el proyecto a realizar tenemos que definir 3 conceptos para comprender en su totalidad el contenido del mismo, para esto se comenzara a mencionar que es una persona ciega. Una persona ciega según la ley y las distintas formas que puede provocar la incapacidad de forma resumida.

Una persona ciega es según la Organización Mundial de la Salud aquella visión menor de 20/400 o 0.05, considerando siempre el mejor ojo y con la mejor corrección.

Una persona ciega según lo legal es aquella que, según la Organización Mundial de la Salud, la visión es menor de 20/200 o 0.1 en el mejor ojo y con la mejor corrección.

La incapacidad puede deberse a distintas causas, las cuales son:

- Accidentes
- Cataratas: La catarata es la pérdida de la transparencia del cristalino, estructura situada en el interior del ojo, por detrás de la pupila, que cumple funciones de una lente, enfocando las imágenes en la retina, además de filtrar y refractar los rayos luminosos.² Esto es tratable en su mayoría por cirugías y se puede afectar a los 2 ojos o a uno en particular
- Glaucoma: Es una de las enfermedades que con más frecuencia produce importantes déficits en la función visual, situándose entre las principales causas de ceguera en todo el mundo.

Se caracteriza por una elevación de la presión intraocular hasta un nivel que produce un daño irreversible en las fibras del nervio óptico.

Las fibras del nervio óptico, se dañan cuando la presión intraocular se eleva por encima de un nivel, que es variable de unos individuos a otros. Si esta presión se mantiene durante mucho tiempo o alcanza cifras muy altas, estas fibras se pueden dañar de forma irreparable y la pérdida de visión se hace irreversible. Cuando la totalidad de las fibras del nervio óptico se han dañado, se pierde por completo la capacidad de transmitir imágenes al cerebro, encontrándonos ante la ceguera total.³

Se tiene 3 tipos de Glaucoma:

- Congénito

² CILSA. Discapacidad Visual

³ CILSA. Discapacidad Visual

- Crónico de ángulo abierto
- Agudo o de ángulo cerrado
- **Retinosis Pigmentaria:** La Retinosis Pigmentaria es una enfermedad ocular que produce una grave disminución de la capacidad visual y que en muchos casos conduce a la ceguera.
Aunque se nace con la enfermedad, esta es raro que se manifieste antes de la adolescencia, y prácticamente siempre lo hace de forma insidiosa de tal modo que el enfermo no es consciente de su enfermedad hasta que ésta se encuentra en fases muy avanzadas.
- **Retinopatía Diabética:** La proliferación y afectación de los vasos sanguíneos del interior del ojo por la Diabetes Mellitus produce la Retinopatía diabética.

Una vez establecido que es una persona ciega legal y la forma en que pueden llegar a padecer la enfermedad, hablaremos un poco de cómo se integran al ámbito educativo de forma resumida. Para esto se tomó a la escuela de educación especial “Corina Lona” como base para poder saber cómo son los pasos a través de los cuales se va integrando a las personas ciegas⁴:

Estimulación Temprana: Se atienden a niños de 0 a 3 años aproximadamente, donde se les brinda apoyo al niño y asesoramiento familiar para su desarrollo integral.

“Un niño que es atendido tempranamente tendrá más y mejores oportunidades de desarrollar sus potencialidades”.

Sala Maternal: Se brinda servicio a niños de entre 3 y 4 años de edad, con un abordaje pedagógico-funcional tendiente a estimular su desarrollo afectivo, cognitivo, social y psicomotriz, dentro de un ámbito familiar e institucional.

Nivel Inicial: Se asiste a niños de pre-jardín y jardín de infantes, de 4 a 5 años aproximadamente. Pretende acrecentar, ejercitar, estimular las potencialidades y capacidades de los niños, teniendo en cuenta las áreas conductuales: Intelectual, Psicomotriz, Social, Afectiva, Lenguaje-Comunicación, Sensor-perceptiva, como así también, responder a las individualidades a partir de un programa ecológico-funcional para niños con patologías asociadas.

Preparatorio: Es el nivel intermedio, entre Jardín de Infantes y primer año de la E.G.B., propio de toda escuela para niños con necesidades educativas visuales. Tiene por objetivo

⁴ Escuela de Educación Especial N°7040 "Corina Lona"

brindar las herramientas necesarias para la lector-escritura braille y el uso del ábaco a partir de un programa de aprestamiento.

Educación Primaria: Ofrece los contenidos socialmente significativos establecidos por la Ley Federal de Educación, a partir de programas con adaptaciones curriculares poco significativas a muy significativas, respondiendo a las necesidades educativas especiales de cada niño.

NEC (Necesidades Educativas Complejas): Ofrece atención integral a niños-jóvenes con Necesidades Educativas Complejas asociadas a la visual, desde edad temprana hasta la transición a la vida adulta. Se brinda asesoramiento y acompañamiento continuo a las familias.

Integración Escolar: Se integra a niños y jóvenes a la escuela común desde el 1º de Educación Primaria hasta el Nivel Secundario. El servicio está a cargo de docentes itinerantes en escuelas de capital e interior de la provincia y viene de la mano con la siguiente sección.

Apoyo a la Integración: Todos los alumnos integrados asisten a la Escuela "Corina Lona", en contra-turno, a recibir apoyo pedagógico y en áreas específicas y complementarias a la educación del sujeto con necesidades educativas visuales.

Programación y sus beneficios

Para continuar con el proyecto es necesario definir por aparte que es la programación y una pequeña historia de la misma.

La programación es un proceso por el cual se le indica a la máquina que debe hacer con sus componentes para realizar una tarea determinada, ya sea de forma física o de forma virtual.

Este se separa en dos conjuntos distintos:

- **Lenguaje de Programación:** Es un lenguaje que comparte elementos con el lenguaje que utilizamos todos los días, ya que, tiene reglas, símbolos y además esta formada por sintaxis y semántica válida y coherente. Estos lenguajes fueron evolucionando para que la programación sea una actividad cómoda y nada frustrante. A través del tiempo, los lenguajes de programación fueron cambiando a distintas metodologías y estructuras para que la programación sea más fácil y cercana a la realidad.

Por ejemplo, en un principio se programaban en 0 y 1 en el sistema binario, que por lo general se interpreta como antónimos lógicos, por ejemplo, abierto o cerrado, con potencia o sin potencia, entre otros; esto ocasionaba que no se pueda programar cosas muy grandes y su supervisión era imposible (para futuros mantenimientos, cambios,

etc.), por la complejidad que este conlleva, los ingenieros fueron creando distintos lenguajes, un intermedio entre el lenguaje cotidiano y el lenguaje máquina.

- **Algoritmo:** son una serie de normas o leyes específicas que hace posible la ejecución de actividades, cumpliendo una serie de pasos continuos que no le originen dudas a la persona que realice dicha actividad. Los algoritmos se pueden expresar de diversas formas: lenguaje natural, lenguaje de programación, pseudocódigo y diagramas de flujo.⁵

La programación contiene mucha disciplina, pero con ello se incorporan muchos beneficios para las personas que puedan manejarla. Para ejemplificar, se nombrarán los 3 beneficios más importantes que conlleva aprender a programar:⁶

1. Pensamiento Computacional

El pensamiento computacional es la capacidad de comunicar sus pensamientos de una manera estructurada y lógica. Este tipo de proceso es como seguir instrucciones paso a paso de manera ordenada, como las que están codificadas en una computadora. También incorpora el proceso de pensar y luego resolver problemas de una manera metódica que podría ser replicada por una máquina. Los ingenieros de software, programadores de computadoras y especialistas en logística utilizan este método de pensamiento para resolver problemas. El pensamiento computacional es una combinación de matemáticas avanzadas, desarrollo de algoritmos y lógica. El desarrollo del pensamiento computacional implica considerar un problema y dividirlo en pasos de acción simple. Cada uno de estos pasos se maneja de la manera más eficiente posible. También hay un elemento de abstracción en el pensamiento computacional, que le permite al sujeto pasar de una solución específica a un problema específico y generalizarlo a otras situaciones. El resultado es un cambio de paradigma al pensar cómo funciona el mundo.

2. Creatividad y Fluidez en el pensamiento

Cuando éramos niños teníamos una mente creativa y fluida que nos permitía pensar de una manera más "fuera de la caja". Las formas casi infinitas de codificar y resolver problemas pueden inspirar a la persona a crecer y cultivar esta parte de la vida un poco olvidada.

⁵ Conceptodefinicion.de, Definición de Algoritmo.

⁶ Elearning Industry. 4 Benefits Of Learning Programming At A Young Age

3. Oportunidades de Trabajo

En un futuro no saber cómo codificar será comparable a no saber leer. La mayoría de los trabajos requieren el conocimiento de habilidades básicas de TI, mientras que incluso los trabajos de venta al por menor y comida rápida requieren el uso de tecnología y computadoras. Los especialistas en codificación están bien pagados y son muy codiciados en el mercado actual, y las oportunidades para estos empleados calificados se ampliarán en el futuro.

4. Apoyo para las TIC's

“En líneas generales podríamos decir que las nuevas tecnologías de la información y comunicación son las que giran en torno a tres medios básicos: la informática, la microelectrónica y las telecomunicaciones; pero giran, no sólo de forma aislada, sino lo que es más significativo de manera interactiva e interconexionadas, lo que permite conseguir nuevas realidades comunicativas.” (Cabero, 1998: 198)

Las Tecnologías de Información y Comunicación (TIC) son todas las tecnologías que administran la información en cualquier formato en que se encuentren (audio, texto, video, etc.). Con la enseñanza de programación para las personas ciegas les ayudaría a entender mejor las tecnologías de administración de datos y de información, siendo útil para cualquier herramienta que utilicen funciones y algoritmos, como por ejemplo las planillas de cálculo.

Este proyecto utiliza una funcionalidad informática, distinta a un programa, denominada Plug-in o complemento, lo cual es “Un complemento de software es un complemento para un programa que le agrega funcionalidad. Por ejemplo, un complemento de Photoshop (como Eye Candy) puede agregar filtros adicionales que puede usar para manipular imágenes. Un complemento del navegador (como Macromedia Flash o Apple QuickTime) le permite reproducir ciertos archivos multimedia en su navegador web. Los complementos VST agregan efectos para grabación de audio y programas de secuenciación como Cubase y Logic Audio.

La mayoría de los programas de gráficos y audio actuales admiten complementos ya que son una forma conveniente de ampliar las capacidades del programa. Aunque algunos complementos pueden enviarse con el programa, la mayoría son desarrollados por terceros y se venden por separado. Dado que las empresas que fabrican complementos de navegador a

menudo compiten por un estándar (como Flash y QuickTime), estos complementos generalmente están disponibles como descargas gratuitas desde Internet.”⁷

Como se definió, un plug-in trabaja sobre un entorno o programa ya realizado, es un agregado de una funcionalidad. Para el proyecto será un plug-in para el entorno Eclipse, que es un entorno IDE para la programación en distintas plataformas, ya sea web, para escritorio y móviles, y también soportado para distintos lenguajes de programación, aunque trabaja de forma específica con Java, el lenguaje al cual está siendo tratado en este proyecto.

Para finalizar, se pondrá una tabla que indica algunas carreras que contemplan la enseñanza de la programación en Argentina, para saber dónde se puede aplicar este proyecto.⁸

Tabla 1 - Carreras

<u>Carreras</u>	
Ingeniería en Sistemas Informática Computación	Armado y Reparación de PC
Licenciatura en Sistemas	Project Management
Administración Informática	Programación Web
Administración de Base de Datos	Administrador de Sistemas
Innovación Tecnología	Administrador de Redes
Analista de Sistemas	Redes
Administración de Sistemas	Infraestructura de IT
Administración de Redes	

Integración de conceptos

La irrupción de la computadora en la vida cotidiana de las personas ha provocado mejoras sustanciales en su desempeño laboral o educativo. También ha repercutido notablemente en las comunicaciones y la recreación.

Desde muy pequeños los niños que tienen acceso a una PC en su casa, mientras juegan aprenden a manejar una herramienta que día a día será más útil. Por su parte el correo electrónico e Internet se han convertido en cosas indispensables para cualquier estudiante, trabajador de una oficina o profesional.

⁷ Christensson, P. (2006). Plug-in Definition. Retrieved 2018

⁸ Guía del Estudiante. Dónde estudiar Programación en Argentina

Las personas con discapacidad visual, desde hace 15 años aproximadamente, cuentan con la posibilidad de manejar en forma autónoma una computadora personal.

A lo largo de estos años los dispositivos y software han ido evolucionando y en la actualidad permiten que un discapacitado visual opere una computadora de la misma manera que una persona con vista.

Enviar e-mails, navegar en Internet o chatear eran tareas impensables 8 años atrás.

Pero, en pocos años, estas tareas también pasaron a formar parte de la vida cotidiana de aquellas personas con discapacidad visual con capacidad para disponer de una computadora.

Es variado el espectro de posibilidades existentes tanto en lo que se refiere a programas como a dispositivos que permiten adaptar la computadora para ser utilizada por una persona ciega o con baja visión.

Sin embargo, a la hora de comprar uno de estos productos resulta imprescindible determinar cuál es la necesidad del usuario y que herramienta podrá ofrecerle las mejores respuestas. Muchas veces ocurre que una persona compra lo que más llama su atención o el último adelanto de una determinada empresa, pero esto no satisface sus necesidades puntuales.

Existen dos formas a través de las cuales, una persona ciega puede acceder al contenido de la pantalla de una computadora.

1. Una a través de un dispositivo que se conecta a la PC, reproduce en Braille el contenido de una línea de la pantalla y cuenta con una serie de botones para desplazarse por la misma.
2. La otra opción es el empleo de un sintetizador de voz, externo o a través de un software que reproduce mediante voz sintetizada el contenido de la pantalla utilizando la placa de sonido del equipo.

La diferencia entre el dispositivo externo y la síntesis por software radica en el precio, los externos son más costosos, por lo que, actualmente la síntesis por software es la más utilizada. Su uso no es complejo y sólo requiere un poco de entrenamiento, ganas de experimentar, un poco de asesoramiento y sacarse la idea de que manejar una computadora es tan complejo⁹.

⁹ Biblioteca Argentina para Ciegos (BAC), Informática

Especificaciones del proyecto

Contexto de aplicación

En este capítulo nos centraremos más en el proyecto, el cual tiene la finalidad de realizar un plug-in o complemento en un entorno de desarrollo IDE, más precisamente en el entorno Eclipse, para ayudar a las personas ciegas legalmente (el que tiene una visión del mejor ojo, con la mejor corrección de 20/200) para realizar la tarea de la programación, y de este modo poder ayudarlos a integrarse a carreras universitarias que tienen esta técnica en algunas de sus materias.

El complemento será realizado por el equipo de desarrollo de la Universidad Católica de Salta, impulsado por un proyecto social para las personas ciegas, realizándose en aproximadamente en 3 meses y teniendo un presupuesto aproximado de USD 28.942. Este presupuesto será financiado por las universidades de la República Argentina que quiera colaborar y obtener el producto, a un precio básico de USD 1 por máquina.

Por último, los recursos asociados para la realización del proyecto son una computadora con un procesador A8 AMD, 4 Gb de RAM, con un sistema operativo Windows 7 como mínimo y el entorno de desarrollo Eclipse. Además, también deben contemplarse el papel y el tóner, ambos siendo orgánicos y gastos varios para mantener un entorno laboral estable y cómodo, incluyendo las necesidades del personal.

Objetivo General

Proveer de una herramienta confiable y eficaz, y, además, que trabaje en conjunto con los lectores de pantallas actuales en el mercado; ayudando así a las personas con visualización restringida o nula, donde puedan aprovechar los beneficios que provee el ejercicio de la programación y ser instruidas con facilidad por los profesionales en el ámbito.

Objetivos Específicos

- Realizar un componente de software de manera que puedan realizar la tarea programática aquellas personas con una visión menor o nula.
- Promover la inclusión de las personas no videntes a carreras universitarias, cuyas materias estén asociadas a la programación visual
- Realizar un componente de software para que, los profesores del área informática, tengan un medio para la enseñanza más homogénea dentro del ámbito.

Alcance

- El complemento se utilizará para el ámbito universitario

- El complemento está pensado en primera instancia para estudiantes ingresantes
- El complemento se desarrollará para la plataforma de Eclipse
- El proyecto será presentado con un prototipo
- Se orientará por TTS la parte sintáctica
- Los errores semánticos se trabajarán solo con los lectores de pantalla
- Se entregarán los planos de desarrollo
- El complemento trabajará solo bajo programación estructurada
- No se reconoce la traducción del manual y planos al braille

Riesgos

Los riesgos asociados al proyecto están descriptos en la siguiente tabla, donde se les colocó un número que lo identifique, el tipo de riesgo, la probabilidad de que ocurra el mismo (yendo del 1 al 5, de lo más improbable a lo seguro), el impacto que tiene sobre el proyecto (yendo del 1 al 5 de lo más leve al caso de que se lo tome como catástrofe) y el riesgo que presenta en total, siendo una multiplicación entre los 2 anteriores, donde del 1 al 5 es inexistente, del 6 al 10 es bajo, del 11 al 15 es medio, del 16 al 20 es alto y del 21 al 25 es total.

Tabla 2 - Tabla de Riesgos

Nro.	Tipo de Riesgo	Probabilidad (p)	Impacto (i)	Riesgo (p*i)
1	Inflación	4 – Muy Probable	2 – Soportable	8 – Bajo
2	Cambio del Gobierno Educativo	2 – Poco Probable	2 - Soportable	4 – Inexistente
3	Renuncia o despido del personal de trabajo	2 – Poco Probable	4 – Grande	8 – Bajo
4	Incompatibilidad	1 - Improbable	5 - Catástrofe	5 – Inexistente

Ahora se describirá cada riesgo y porque se eligió sus valores:

1. Inflación: El proyecto a tener un presupuesto inicial en dólares, si la moneda argentina se desvalúa, el proyecto sería más difícil realizarlo, y con el transcurso del tiempo, la desvaluación es cada vez más evidente, por lo tanto, tiene una probabilidad alta. Pero como el tiempo del proyecto es corto, la desvaluación no impactara agresivamente al

proyecto, haciendo que la misma impacte lo suficiente para que sea soportable por el equipo de trabajo. Por lo tanto, al tener un impacto bajo, pero una probabilidad alta, se considera que el riesgo es bajo, porque, aunque esto ocurra, el proyecto terminara antes de que el cambio sea superior a lo esperado.

2. Cambio del Gobierno Educativo: En el transcurso del proyecto, el ministro de educación, como sus empleados, pueden ser cambiados por un cambio de gobierno en la Argentina. El proyecto a tener una vida útil baja, la probabilidad de que esto ocurra justo durante la ejecución del proyecto es baja, y el impacto tampoco sería grande, ya que el equipo de la Universidad Católica de Salta es una institución privada y autónoma. Esto da como resultado un riesgo inexistente.
3. Renuncia o despido del personal de trabajo: El equipo de trabajo y el esfuerzo del mismo es lo más importante para que un proyecto de cualquier magnitud se realice apropiadamente. La probabilidad de que haya renuncia o despido es baja, existe una probabilidad, pero llega a ser lo suficientemente baja para no preocuparse constantemente. Lo contrario es el impacto, si un empleado es despedido o renuncia por algún motivo, el proyecto se atrasará hasta conseguir su reemplazo, y a este reemplazo enseñarle y ponerlo al corriente del proyecto conllevaría más tiempo. Teniendo una probabilidad baja y un impacto alto, este riesgo es definido como bajo.
4. Incompatibilidad: Este riesgo ocupa un lugar donde el complemento o plug-in sea incompatible con el entorno de desarrollo, logrando así que sea inservible como se muestra en el impacto. Pero como el proyecto está realizado para un entorno determinado, la probabilidad de que esto ocurra es inexistente, por lo tanto, su riesgo es en sí también inexistente.

Metodológicas para la realización

Proceso Unificado Racional

Proceso Unificado (RUP) es un proceso de desarrollo de software: “conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema software”. Es un marco genérico que puede especializarse para una variedad de tipos de sistemas, diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y diferentes tamaños de proyectos. RUP está basado en componentes, ya que el software está formado por ellos y están interconectados a través de interfaces. Por último, la metodología está:

- Dirigida por casos de uso: Un caso de uso es un fragmento de funcionalidad del sistema que proporciona un resultado de valor a un usuario. Los casos de uso modelan los requerimientos funcionales del sistema.
- Centrado en la arquitectura: Conjunto de decisiones significativas acerca de la organización de un sistema software, la selección de los elementos estructurales a partir de los cuales se compone el sistema, las interfaces entre ellos, su comportamiento, sus colaboraciones, y su composición.
- Es iterativo e incremental: Es práctico dividir el esfuerzo de desarrollo de un proyecto de software en partes más pequeñas o, comúnmente dicho, separar el problema en problemas más chicos y manejables. Cada parte del proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos a crecimientos en el producto.

Además, la metodología permite una gestión del proyecto a través de su metodología, para una adaptación total junto con este. Más adelante se comentará de qué se trata la gestión del proyecto de la metodología del proceso unificado racional.

¿Por qué el Proceso Unificado Racional?

A la problemática planteada y el software propuesto, es necesario una metodología eficaz en el desarrollo del software. Se utilizará la metodología RUP (Proceso Unificado Racional) el cual cumple con ciertas especificaciones útiles y convenientes para el desarrollo del proyecto.

Primero es un desarrollo iterativo, utilizado para la construcción de productos cuyos ciclos de vida está compuesto por un conjunto de iteraciones, las cuales tiene como objetivo entregar versiones del software¹⁰. Esto nos da la pauta de que no hay un problema tan grande que no se pueda resolver, lo cual ayuda a este proyecto en muchas formas, primero viendo las diferencia entre la sociedad y lo técnico, pasando después a una subdivisión más manejables y adaptables.

Al trabajar por iteraciones, estas son compuestas por cuatros distintas fases, utilizadas generalmente en el desarrollo de los sistemas, estas partes son:

- Inicio o inserción: Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones. (Gil, 2004)
- Elaboración: Los casos de uso seleccionados para desarrollarse en esta fase permiten definir la arquitectura del sistema, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar del problema y comienza la ejecución del plan de manejo de riesgos, según las prioridades definidas en él. (Gil, 2004)
- Construcción: El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar el cambio de los artefactos construidos, ejecutar el plan de administración de recursos y mejoras en el proceso de desarrollo para el proyecto. (Gil, 2004)
- Transición: El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados, capacitar a los usuarios y proveer el soporte técnico necesario. (Gil, 2004)

¹⁰ Robín Alberto Castro Gil. Estructura básica del proceso unificado de desarrollo de software. Página 31

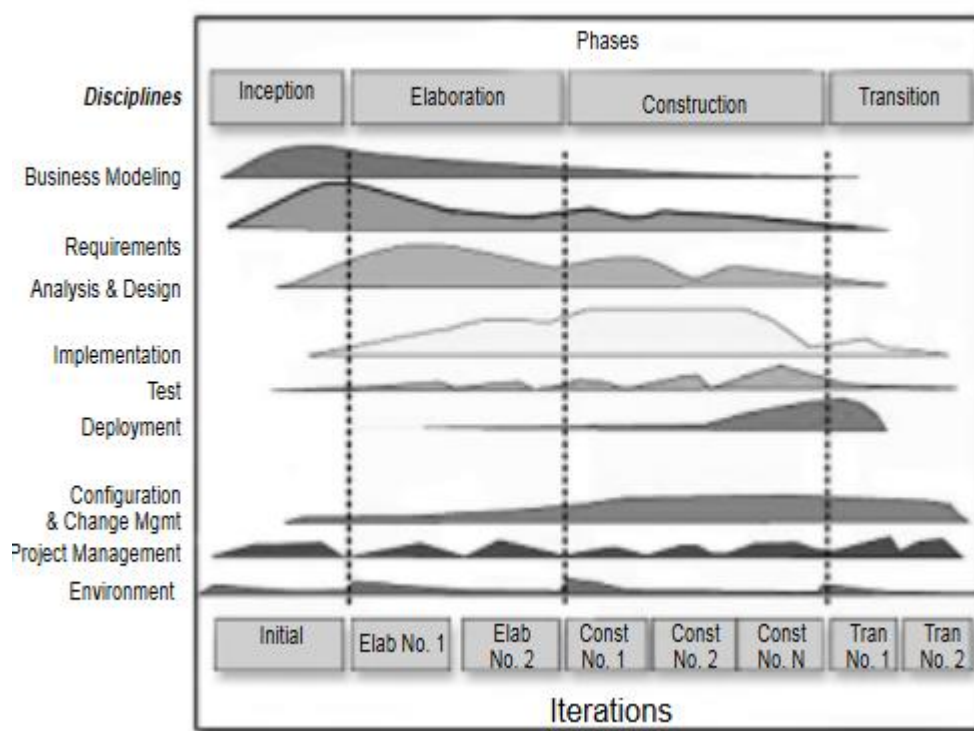


Imagen 3 - Proceso Unificado Racional

Fuente: Robín Alberto Castro Gil. Estructura básica del proceso unificado de desarrollo de software.

En segundo lugar, está orientado al manejo del riesgo, todo proyecto tiene riesgos o impedimentos que retrasan el mismo. Para esto hay que tener unos controles eficientes de ellos y planear como evitarlos, en el caso que sucedan, como salir victoriosos de estos. El proyecto, por lo que contiene material social, contiene una cantidad de riesgos considerables y por lo general variables. Se necesita tener un equipo capacitado, pero aún más, un método que ayude a la salida exitosa de los riesgos.

En tercer lugar, está orientado al cliente, en este caso a la sociedad no vidente que tengan el interés de programar, pero aquí decae un poco la metodología, generalmente hay un cliente que nos hace saber sobre sus necesidades. La cantidad de gente no vidente es entre el 1% y 4% de la población de los países latinoamericanos y un porcentaje aproximado del 1.45% en argentina¹¹, dándonos un total aproximado del 636 miles de personas con visibilidad nula. Por lo cual se obtendrá un grupo selecto que ayudará al proyecto y al programa y se analizará un cuestionario para tener una idea más globalizada para incluir a la gran mayoría.

¹¹ Visión 2020. Cifra de ceguera en Latinoamérica

Por último, pero no menos importante, es un desarrollo evolutivo, por lo que se va desarrollando de a poco y sin dejar de investigar. Esto es sustancial para que el cliente ayude al desarrollo.

El Proceso Unificado de Racional trabaja en conjunto con distintas herramientas para desarrollar los planos previstos para realizar el objetivo completo del proyecto, estos están categorizados y especificados en el Lenguaje Unificado de Modelado (UML), donde serán aplicados en las fases de Inicio y Elaboración en el presente proyecto. En este proyecto se realizará para la fase de inicio la herramienta denominada diagrama de flujo, donde englobará todos los requisitos y pasos que tiene que realizar la persona ciega para usar el plugin y aprovechar sus ventajas. Para la fase de Elaboración se eligió un subconjunto de las herramientas del UML para la realización del diseño del componente, las cuales vienen a ser las siguientes:

- Diagrama de Clases
- Diagrama de estado-transición
- Diagrama de Actividades
- Diagrama de Componentes

Todas estas herramientas serán descriptas en el capítulo de “**¡Error! No se encuentra el origen e la referencia.**”

Gestión del Proyecto

La Gestión del proyecto nos permite lograr un balance al gestionar objetivos, riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.

Los objetivos de este flujo de trabajo son:

- Proveer un marco de trabajo para la gestión de proyectos de software intensivos.
- Proveer guías prácticas realizar planeación, contratar personal, ejecutar y monitorear el proyecto.
- Proveer un marco de trabajo para gestionar riesgos.

La planeación de un proyecto posee dos niveles de abstracción: un plan para las fases y un plan para cada iteración.

Solución

Metodología en acción

Las funcionalidades que buscamos para que el proyecto sea realizado con éxito, es la realización de un plugin funcional que trabaje dentro del entorno eclipse, donde el propósito es ayudar a la persona ciega a guiarse dentro del código, más precisamente dentro de los bloques de las sentencias (en este caso en los bloques que empiezan con una llave abierta “{“ y terminan en una llave cerrada “}”) , además de incluirle ayudas por medios auditivos para saber si las sentencias están escritas correctamente en el ámbito sintáctico.

Para esto es necesario utilizar las herramientas descriptas en el capítulo anterior, siguiendo la metodología elegida, donde se plantearán los planos para el desarrollo de la aplicación para realizar el software de manera correcta y que cumpla con la meta de solucionar el problema.

Los planos realizados se encuentran en el Anexo I

Las herramientas a utilizar en la metodología del proceso unificado racional son las denominadas UML (Unified Modeling Language) o lenguaje de modelado unificado, por el cual se presentan las siguientes herramientas dentro del lenguaje unificado, distribuidos en dos distintos modelos:

- Modelado de Requisitos: En este modelo se definiría todas las características necesarias que tiene que tener el producto, para cubrir todas las necesidades del proyecto. Todas las características que tiene que tener el producto realizado en este proyecto están descriptas en su objetivo y alcances. La única herramienta nueva y que aporta información útil al proyecto es un diagrama de flujo.

Un diagrama de flujo es una representación gráfica de un flujo de un algoritmo o de una secuencia de acciones rutinarias. Se basan en la utilización de diversos símbolos para representar operaciones específicas. Se les llama diagramas de flujo porque los símbolos utilizados se conectan por medio de flechas para indicar la secuencia de la operación

Dentro de las ventajas de realizar Diagramas de Flujo podemos encontrar:

- Favorecen la comprensión del proceso/procedimiento al mostrarlo como un dibujo. Esto por cuanto el cerebro humano reconoce más fácilmente los dibujos que la escritura en prosa. Un buen diagrama de flujo reemplaza varias páginas de texto.

- Permiten identificar los problemas y las oportunidades de mejora del proceso. Se identifican los pasos redundantes, los conflictos de autoridad, las responsabilidades, los cuellos de botella, y los puntos de decisión.
- Muestran las interfaces cliente - proveedor y las transacciones que en ellas se realizan, facilitando a los empleados el análisis de las mismas.
- Son una excelente herramienta para capacitar a los nuevos empleados y también a los que desarrollan la tarea, cuando se realizan mejoras en el proceso¹².

En el modelado de requisitos para del proceso unificado se suele utilizar una herramienta conocida como “Lista de Casos de Usos” y “Especificaciones de los casos de usos”.

Los casos de usos son acciones que realizan unos determinados actores (en este caso serían las personas ciegas) sobre el programa y como interactúa el programa con este actor, pero al ser un proyecto donde todo lo realiza el componente, no solicitándole nada a la persona ciega más allá de una tecla, se desestima la herramienta por el motivo que su realización no será relevante para el desarrollo del mismo, siendo más eficiente un diagrama de flujo.

- Modelado del diseño: Con todo lo anterior tenemos la suficiente información y hechos para poder utilizar las herramientas propias del diseño, útiles para poder realizar un producto eficiente.

Para este modelado se utilizarán las siguientes herramientas:

- Diagrama de Clase: Para conocer esta herramienta es necesario definir que es una clase y un objeto. Una clase es conjunto de objetos similares, es decir, con la misma estructura y comportamiento, y constituidos por los mismos atributos y métodos¹³. Un objeto es una entidad identificable del mundo real. Puede tener una existencia física (un caballo, un libro) o no tenerla (un texto de ley).¹⁴

El Diagrama de Clase contiene los atributos, métodos y asociaciones de los objetos.

Las clases se representa de la siguiente forma:

¹² Mideplan, Guía para la Elaboración de Diagramas de Flujo

¹³ DEBRAUWER y VAN DER HEYDE, 2016.

¹⁴ DEBRAUWER y VAN DER HEYDE, 2016.

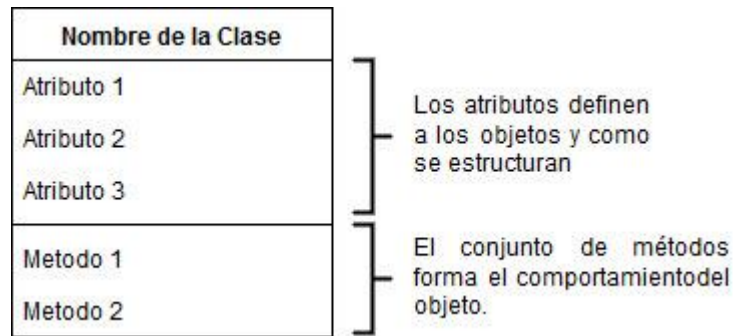


Imagen 4 - Estructura del Diagrama de Clase

Las clases tienen asociaciones entre ellas, estos son vínculos que los une, por ejemplo, entre un objeto y su propietario, contiene un vínculo de pertenencia. En el diagrama de clase se representan de la siguiente forma:

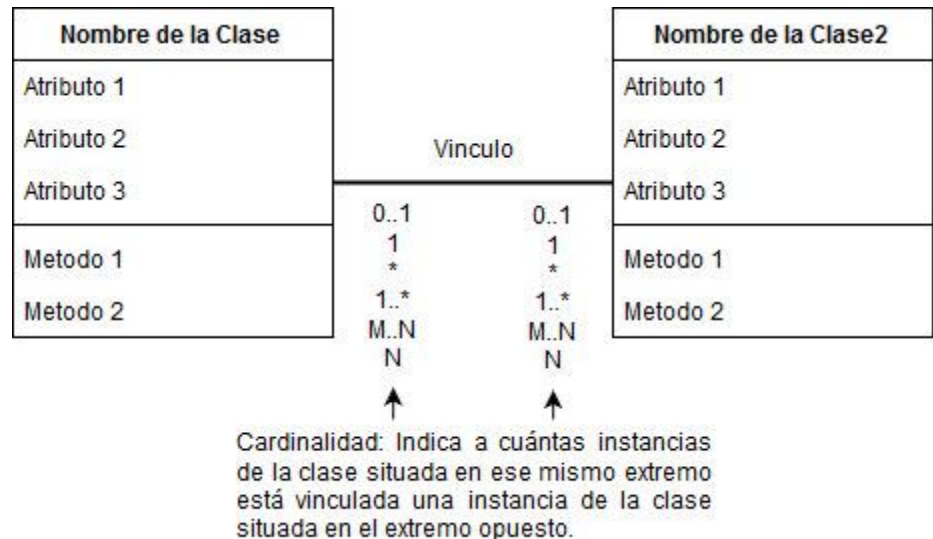


Imagen 5 - Estructura del Diagrama de Clase con vinculo y cardinalidad

Además, dentro del proyecto se debe definir que es un objeto compuesto, el cual es un objeto que puede estar siendo complementado por otros objetos. Esto en el diagrama de clase se lo separa en composición débil o fuerte. El grafico para representarlo es el siguiente:

Tabla 3 - Diagrama de Clase - Composición Fuerte y Débil

Composición Fuerte	Composición Débil
--------------------	-------------------

La diferencia entre ambas son las siguientes:

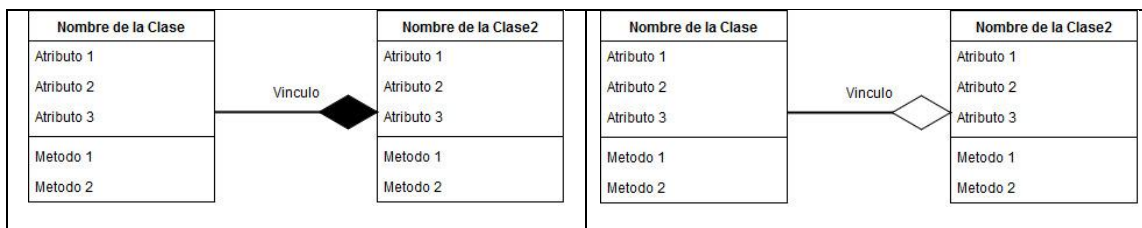


Tabla 4 - Diferencia entre Composición Fuerte y Débil

	Composición Fuerte	Composición Débil
Representación	Rombo Negro	Rombo Transparente
Varias asociaciones comparten los componentes	No	Sí
Destrucción de los componentes al destruir el compuesto	Sí	No
Cardinalidad a nivel del compuesto	0..1 o 1	Cualquiera

- Modelado del Ciclo de Vida de los Objeto: El ciclo de vida de un objeto representa las diferentes etapas o estados por los que pasa para llegar, dentro del sistema, a realizar un objetivo. Un estado corresponde a un momento de actividad o de inactividad del objeto. La inactividad se produce cuando el objeto espera a que otros objetos concluyan una actividad.

Un evento es un hecho que activa el cambio de estado. UML define cinco tipos de eventos:

- AnyReceivedEvent: corresponde a un evento cualquiera. Sin embargo, solo provoca un cambio de estado si ningún otro evento con un tipo diferente a AnyReceivedEvent provoca al mismo tiempo un cambio de estado. La sintaxis de este evento es la palabra clave all.
- CallEvent: un evento CallEvent se produce cuando se invoca al método correspondiente.
- ChangeEvent: un evento ChangeEvent se produce cuando se produce un cambio en el sistema. Este cambio se describe mediante una expresión lógica que es verdadera en el momento en que se produce el evento.
- TimeEvent: un evento TimeEvent se especifica mediante una instrucción que indica un tiempo absoluto o relativo. Si el tiempo es relativo, la

instrucción adopta la sintaxis after expresión. Si el tiempo es absoluto, la instrucción adopta la sintaxis at expresión. En ambos casos, cuando se alcanza el momento indicado, se produce el evento.

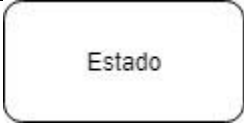
- **SignalEvent:** un evento SignalEvent se produce cuando se recibe una señal.

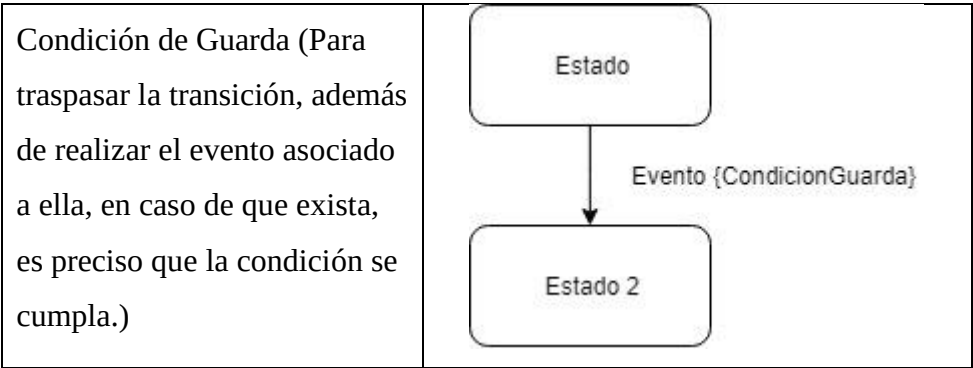
Una transición es un vínculo orientado entre dos estados que expresa que el objeto puede pasar del estado inicial de la transición al estado de destino. Esto generalmente está asociado a eventos y, pero esto no es así si la transición es automática.

Para modelar el ciclo de vida de un objeto, se utiliza los diagramas de estados-transiciones, donde se describen los estados, las transiciones que los vinculan y los eventos que provocan el traspaso de las transiciones.

Sus elementos se diagraman de la siguiente forma:

Tabla 5 - Elementos del Diagrama de Estados-Transiciones

Elementos	Diagramas
Estado	
Estado Inicial	
Estado Final	
Estado de Terminación	
Transición	

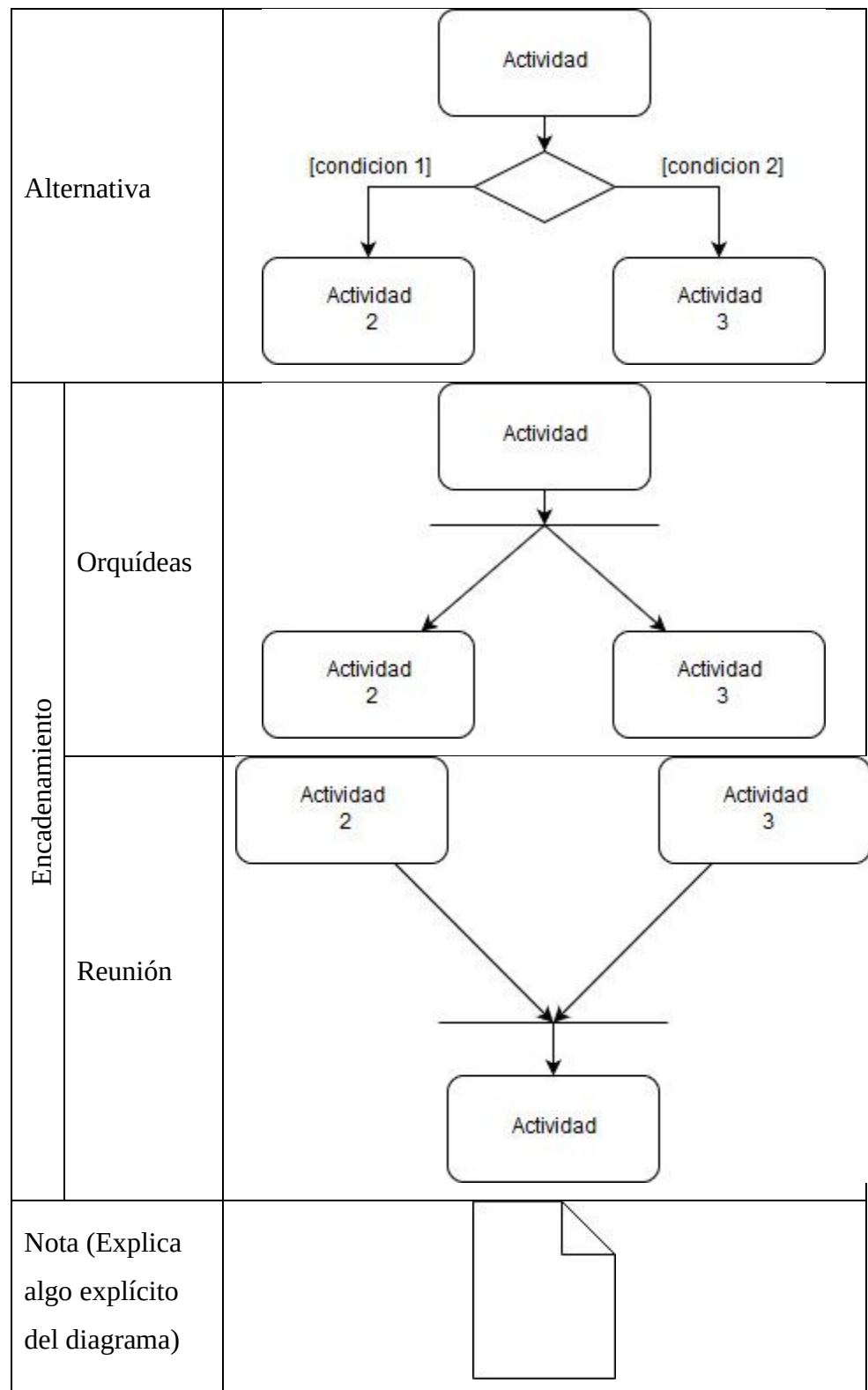


- **Modelado de Actividades:** El modelo de actividad se trata de una forma específica del diagrama de estados-transiciones en la que cada estado se asocia a una actividad y todas las transiciones son automáticas. Posteriormente se amplió para describir las actividades de varios objetos y de esta forma, se pueden representar los encadenamientos entre las actividades de diferentes objetos, cosa que no es posible con el diagrama de estados-transiciones.

Los elementos del diagramado son:

Tabla 6 – Elementos básicos del Diagrama de Actividades

Elementos	Diagramas
Estado	
Estado Inicial	
Estado Final	
Encadenamiento o de Actividades	



A diferencia del diagrama de estados-transiciones, el diagrama de actividades puede representar las actividades realizadas por varios objetos con sus encadenamientos, lo cual es lo que se requiere para el proyecto. Para ello el diagrama se divide en

particiones o calles. A cada calle corresponde el objeto responsable de la realización de todas las actividades contenidas en esa calle o partición.

Además de los anteriores elementos, se agregan:

Tabla 7 - Elementos Necesarios del Diagrama de Actividades

Elementos	Diagramas
Calle o Flujo	
Flujo de Objetos:	
Recepción y emisión de señales: En un diagrama de actividades es posible implementar el envío asíncrono de una señal por parte de un objeto así como la recepción de una señal por un objeto	

- Diagrama de Componentes: Como hablamos anteriormente, los componentes son una unidad de software que ofrece una serie de servicios a través de una o varias interfaces.

Este diagrama de observa de la siguiente forma:

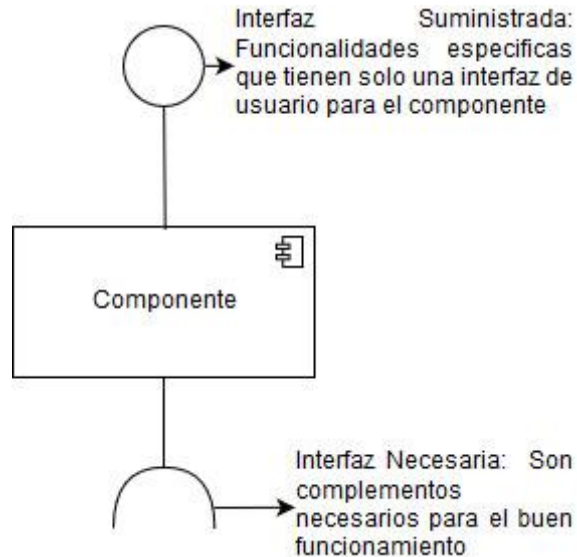


Imagen 6 - Estructura del Diagrama de Componentes

Alternativas tecnológicas de solución del problema

Para este proyecto se planteará las siguientes alternativas, seleccionando una de ellas por razones de comodidad y estadísticas. Las alternativas son:

Tabla 8 - Alternativas Tecnológicas

		Alt. 1	Alt. 2	Alt. 3
Computador	Procesador	AMD A8	Intel i5 4ta generación	AMD A8
	Memoria RAM	4 Gb	8 Gb	8 Gb
	Disco Rígido	500 Gb	1 Tb	1 Tb
Cantidad		1	1	1
Sistema operativo		Windows	Windows	Linux
Entorno de desarrollo		Eclipse	Eclipse	Vim
Entorno de Prueba		Eclipse	Eclipse	Eclipse
TTS		VoiceRSS.org	VoiceRSS.org	VoiceRSS.org

Se elige la alternativa número 1, ya que se utiliza la maquina más económica para su realización, utiliza Windows como sistema operativo, que tiene un soporte para personas no videntes. Se orientará a Java la realización del complemento, para que funciones de manera nativa, ya que es el lenguaje es por excelencia para la enseñanza universitaria. Por ultimo las alternativas de Windows se utilizan el entorno eclipse, ya que es el que más soporte da para proyectos de desarrollo de plugin, además de tener una comunidad grande para el apoyo de la realización del complemento. Por último, el TTS probado es el que provee la página VoiceRSS.org, ya que contiene un entorno en la nube para ser consultado, teniendo un precio ilimitado que es abonado por mes.

Arquitectura de procesamiento y comunicación de la propuesta tecnológica

Esta parte de la solución no está contemplada dentro del proyecto, ya que no hay una arquitectura de procesamiento que se pueda aclarar ni una comunicación entre base de datos y servidores, no logrando un avance significativo en este ámbito.

Ingeniería del proyecto

1.1.1. Procesos técnicos y de negocios

- Inicio o inserción: Se analizará todo lo relevante al problema, desde un planteamiento del mismo hasta la parte de definir las funcionalidades que tendrá cada persona y como se desarrollará.
- Elaboración: Esta sección se encarga de la presentación visual y de las primeras es del proyecto, diagramando sus planos de desarrollo y como deberá funcionar una vez finalizado el proyecto. Esta de una parte esencial, ya que si los planos no se realizan bien puede ocurrir un error grave en la construcción o puede persistir un error en la parte de inicio por el cual se deberá reformular las características y los casos de uso.
- Construcción: El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar el cambio de los artefactos construidos, ejecutar el plan de administración de recursos y mejoras en el proceso de desarrollo para el proyecto. (Gil, 2004)
- Transición: El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados, capacitar a los usuarios y proveer el soporte técnico necesario. (Gil, 2004)

Matriz de responsabilidades

Tabla 9 - Matriz de Responsabilidades

Tareas \ Responsables		Analista / Diseñador	Analista Funcional	Programador 1	Programador 2	Tester
Inicio o Inserción	Definir y acordar el alcance del proyecto	Auxiliar	Responsable			
	Identificar los riesgos	Auxiliar	Responsable			
	Generalizar la arquitectura de software	Auxiliar	Responsable			
	Producir el plan de las fases y el de iteraciones	Responsable	Auxiliar			
	Realizar Diagrama de Flujos del Componente	Responsable	Auxiliar			
Elaboración	Diagrama de Clase	Responsable	Auxiliar			
	Diagramas del Ciclo de Vida de los Objeto	Responsable	Auxiliar			
	Diagrama de Actividades	Responsable	Auxiliar			
	Diagrama de Componentes	Responsable	Auxiliar			
Construcción				Responsable	Auxiliar	
Prueba						Responsable
Transición			Auxiliar	Responsable		

Asignación de Recursos y Tiempo

VISIÓN GENERAL DE LOS RECURSOS



Imagen 7 - Visión General de los Recursos



Imagen 8 - Visión General de los Tiempos

Análisis Económico-Financiero
Recursos Humanos

Tabla 10 - Recursos Humanos del Proyecto

Nombre	Cantidad	Valor Unitario	Costo (U\$D)
Jefe de Proyecto	1	14.948,16	14.948,16
Analista	1	5.175,36	5.175,36
Programador	2	3.581,76	7.163,52
Tester	1	1.307,52	1.307,52

Recursos Materiales

Tabla 11 - Recursos Materiales del Proyecto

Nombre	Costo (U\$D)
Papel A4 (ecológico)	5,73
Tóner (ecológico)	28,66
TTS*	300,00
Otros	12,61

*El TTS en el periodo de desarrollo es gratis, pero al finalizar se necesita pagar por mes una cantidad de 300 dolares

Recursos de Desarrollo

Tabla 12 - Recursos de Desarrollo

Nombre	Costo (U\$D)
Entorno de Desarrollo Eclipse	0
CPU A8, 4Gb de RAM y 500 Disco Duro	0

Total: U\$D 28.942

*Tomando U\$D 1 = \$23,16 pesos argentinos

Gestión de Riesgos

El modo en que se planifica el desarrollo del sistema está influenciado en gran medida por los riesgos que se perciben. Por lo tanto uno de los primeros pasos, al principio de la fase de inicio, es crear una lista de riesgos (véase 0). La administración de estos riesgos es una tarea clave, por lo tanto, se aumentará quien será el responsable de controlar el riesgo y las acciones a tomar para esta persona

Tabla 13 - Gestión de Riesgos

Nro. Del Riego	Riesgo	Responsable	Acciones
1	Inflación	Analista Funcional	1. Evaluar la desviación del presupuesto
			2. Informar al cliente
			3. Negociar con el cliente entre las funcionalidad y un nuevo presupuesto
2	Cambio del Gobierno Educativo	Analista/Diseñador	1. Evaluar si hay nuevas políticas de educación especial
			2. Modificar los requerimientos para moldearse a las nuevas políticas
3	Renuncia o despido del personal de trabajo	Analista Funcional	1. Evaluar el avance del proyecto con la falta del personal
			2. Evaluar el mercado laboral en busca de un suplente
			3. Realizar un proceso de Reclutamiento
			4. Realizar un proceso de Selección
			5. Realizar un proceso de Inducción al proyecto

4	Incompatibilidad	Analista/Diseñador	1. Evaluar una nueva incompatibilidad del software
			2. Modificar o reelaborar los requerimientos y evaluar el alcance
			3. Informar al Analista Funcional sobre la incompatibilidad y la solución encontrada

Factibilidad Técnica

La factibilidad técnica viene dada por 3 cuestiones distintas, primero el hardware que se necesitará para su desarrollo e implementación, en segundo lugar, el software donde se tendrá que visualizar si el entorno virtual es apto y tercero la intelectual, evaluando si es posible hacerlo y si hay personas aptas para realizarlos.

Para el hardware se puede ver que es factible, ya que en las instituciones donde se los quiere implementar es donde ya se enseñen programación, por lo cual cumple los requerimientos mínimos que se necesita para eclipse. Además, se debe ofrecer parlantes o auriculares para que el proyecto se realice totalmente, porque el plug-in y el programa auxiliar que se utiliza para leer la pantalla tiene que usarse con TTS.

El software por otro lado puede variar, pero el entorno IDE Eclipse, funciona tanto para Windows como para Linux, siendo apto para cualquier entorno. Además, tiene que tener las máquinas virtuales de java, que son fácilmente instalables siguiendo las indicaciones del IDE.

En lo intelectual no habría inconveniente, el entorno Eclipse cuenta con un sistema de ayudas para la programación de plug-in, y solo se tendría que saber su estructura, lo cual el IDE tiene su propia documentación completa. La programación del componente es en el lenguaje

Java, uno de los lenguajes más conocidos a nivel mundial, siendo fácil encontrar personas con este conocimiento.

En conclusión, la factibilidad Técnica esta solventada en todas sus áreas, siendo posible en su totalidad y teniendo un riesgo aproximadamente nulo.

Operativa

La factibilidad operativa se cumple si en un futuro el producto va a ser utilizado o no. La Argentina tiene en la ley de educación nacional, en el artículo 11, las siguientes palabras “Brindar a las personas con discapacidades, temporales o permanentes, una propuesta pedagógica que les permita el máximo desarrollo de sus posibilidades, la integración y el pleno ejercicio de sus derechos”. Para la inclusión correcta de las personas discapacitadas en el ámbito universitario, es necesario herramientas para que se puedan integrar más fácilmente.

Como vimos anteriormente, en la Argentina hay aproximadamente 636 mil personas ciegas, donde cada una se le rige el derecho a estudiar una carrera universitaria e incentivándolos a carreras científicas, ya que van a tener un respaldo mejor y una ayuda en un área problemática para ellos.

La factibilidad operativa se cumple, porque aquellas personas ciegas que ejerzan sus derechos a recibir educación superior en donde se enseñe programación, van a utilizar el componente como medio de aprendizaje y una forma para poder estar al mismo nivel de sus compañeros universitarios que no tienen el problema de ceguera.

Legal

Una determinación de cualquier infracción o ilegalidad que pudiere resultar del desarrollo del sistema.

El proyecto deberá acatar las siguientes Leyes / Normas:

- CÓDIGO PENAL DE LA NACIÓN ARGENTINA
- LEY 26.388 de Ley de Delitos Informáticos
- LEY 22.362 de Marcas y Designaciones, establece que las marcas y designaciones comerciales que pueden registrarse en la Rep. Argentina.
- LEY 11.723, régimen legal de la Propiedad Intelectual para la Rep. Argentina, con la inclusión expresa de los programas de computación dentro de las obras intelectuales protegidas (luego de las modificaciones de la Ley 25.036).
- Decreto 41.223/1934, reglamenta la Ley 11.723 de Propiedad Intelectual.

- LEY 24.624 de Presupuesto General de la Administración Nacional (1996), modificatoria de la Ley 11.672, que considera con pleno valor probatorio a la documentación de la Administración Pública Nacional archivada en soportes electrónicos.
- LEY 25.856 de Asimilación de la Producción del Software como Actividad Industrial, establece que la actividad de producción de software debe considerarse como una actividad productiva de transformación asimilable a una actividad industrial, a los efectos de la percepción de beneficios impositivos, crediticios y de cualquier otro tipo.
- LEY 25.922 de Promoción de la Industria del Software, crea un Régimen de Promoción de la Industria del Software para poder acceder al fondo fiduciario del FONSOFT.
- LEY 26.692 de Promoción de la Industria del Software, que modifica la Ley 25.922 extendiendo los plazos de vigencia.
- DECRETO 1594/2004, que reglamenta la LEY 25.922 de Promoción de la Industria del Software e instituye el FONSOFT dentro de la Secretaría de Ciencia, Tecnología e Innovación Productiva del Ministerio de Educación, Ciencia y Tecnología de la Nación y a través de la Agencia Nacional de Promoción Científica y Tecnológica.
- DECRETO 1315/2013, que aprueba la Reglamentación de la Ley 25.922 y su modificatoria Ley 26.692, y crea en el ámbito de la Subsecretaría de Industria, dependiente de la Secretaría de Industria del Ministerio de Industria, el Registro Nacional de Productores de Software y Servicios Informáticos.
- Ley N° 25.573 – Educación Superior
- Ley N° 25.730 – Programas y Proyectos a favor de las personas con discapacidad

Analizando el marco legal y regulatorio de la República Argentina, en el contexto del proyecto, nos encontramos aptos para declarar que acatamos de forma absoluta y rigurosa, las dispuestas leyes y regulaciones.

Económica-Financiera

Para evaluar si es económicamente fiable, se nombrará a continuación los costos tangibles y los beneficios intangibles propuestos:

- ❖ Costo de equipamiento y mano de obra: U\$D 28.942
- ✓ Beneficios Intangibles:
 - Inclusión social
 - Incentivo Tecnológico

- Incentivo de Desarrollo
- Incentivo a la Educación
- Profesionales no videntes más capacitados en el área informática
- Mejora en la calidad de enseñanza en las universidades a las carreras que contienen parte de la programación.
- Mejora de la imagen de la universidad

Como se puede observar, todos los beneficios son intangibles, por la naturaleza del proyecto, pero analizando y considerando el apoyo financiero tanto de la sociedad como del mismo Estado de la República Argentina, ya que es la que más se beneficia por su realización, se considera que el costo es aceptable y correcto para la realización del mismo. Además, si todas las universidades plantean solo pagar un 1 dólar por estudiante, siendo esto un gasto mínimo, los gastos de desarrollo quedaran solventados solamente si el 8% de las personas ciegas quieran programar e incluirse en carreras en las que la programación es una materia. Anteriormente, en la sección “¿Por qué el Proceso Unificado Racional?” concretamos que en argentina hay 636 mil personas ciegas, si el 4,55% de esta cantidad quiere unirse a una carrera universitaria que incluya la programación, los gastos del desarrollo quedarían absolutamente solventados.

Ambiental

El impacto ambiental no es excesivo, pero si influyente, por lo cual se planteara usar hojas recicladas para una mejora sustancial en este ámbito y tinta ecológica, haciéndose de esta forma factible totalmente por la parte ambiental.

Conclusión y Futuras Líneas de Investigación

La conclusión del proyecto se desarrolla durante el trabajo, indicando que no importa que tan difícil sea, la integración de los discapacitados en ámbitos que pensamos que no pueden ejercer es limitarlos a ellos y a la sociedad para que no crezca. Es deber de la sociedad hacer que sean incluidas estas personas en todos los ámbitos y que los obstáculos sean disminuidos. La motivación del proyecto viene dada por estas cuestiones, las personas ciegas con el componente descripto lograran notar que pueden elegir carreras que capaz no les convenía por su condición, perdiéndose así a un número importante de científicos en el área informática con capacidad para brillar y mejorar el día a día de otras personas.

Las futuras líneas de investigación que puede seguir con este proyecto será mejorar el producto aquí descrito, porque, aunque cubra una parte de la problemática, esta no está totalmente resuelta. Se propone las siguientes líneas para completar un objetivo aun mayor, que es incluir totalmente a una persona ciega en el ámbito informático:

- Normalizar y estandarizar las enseñanzas a personas ciegas en un ámbito universitario
- Realizar componentes similares al aquí descrito, para que en todos los lenguajes actualmente utilizados tengan soportes para las personas ciegas.
- Mejorar el entorno educativo universitario, para pasar documentación y etiquetado al braille.
- Mejoramiento de la herramienta para adaptarlo en otras metodologías de desarrollo, como ser orientado a objetos y a eventos
- Optimizar la herramienta para que personas ciegas que quieran programar por si solas, por medios externos al ámbito universitario, puedan realizarlo de manera correcta para no depender tanto la presencia de algún docente.
- Generar el plugins para lenguajes de enseñanza secundaria o primaria, como ser Logo, para que se pueda ampliar a la secundaria y primaria, y llegar así a todas las edades y nivel educativo.

Bibliografía

- Visión 2020. Cifras de Ceguera en Latinoamérica.
<https://vision2020la.wordpress.com/2014/07/14/cifras-de-ceguera-en-latinoamerica/>. Página vigente al 27/08/2017
- Robín Alberto Castro Gil, 2004. Estructura básica del proceso unificado de desarrollo de software. Apuntes de catedra de la universidad ICESI.
- A.U.S. Gustavo Torossi. El Proceso Unificado de Desarrollo de Software
<http://dsc.itmorelia.edu.mx/~jcolivares/courses/pm10a/rup.pdf>. Página vigente al 26/08/2017
- DEBRAUWER y VAN DER HEYDE, 2016. UML 2.5 - Iniciación, ejemplos y ejercicios corregidos (4ª edición). ENI Training. ISBN: 9782409003721
- JACOBSON, BOOCH & RUMBAUGH, 2000. El Proceso Unificado de Desarrollo de Software. Addison Wesley. ISBN: 0-201-57169-2
- Consejo Profesional de Agrimensores, Ingenieros y Profesionales a fines. Honorarios de Informática. <http://www.copaipa.org.ar/informatica/>. Página Vigente al 24/09/2017
- Consejo Profesional de Ciencias Informáticas de la Provincia de Córdoba. Tabla de Honorarios. <http://www.cpcipc.org.ar/content/honorarios/>. Página Vigente al 24/09/2017
- Ministerio de Salud. Servicio Nacional de Rehabilitación. Legislación sobre Discapacitada. <https://www.snr.gob.ar/legislacion-sobre-discapacidad/>. Página Vigente al 20/10/17
- CILSA, ONG por la inclusión.
https://www.cilsa.org/paginas_php/esp_sitio/util/discapacidad/visual.php#ceguera.
Página Vigente al 11/04/18
- ConceptoDefinicion, Definición de Algoritmo,
<http://conceptodefinicion.de/algoritmo/>. Página vigente al 17/04/18
- Elearning Industry, 4 Benefits Of Learning Programming At A Young Age,
<https://elearningindustry.com/4-benefits-learning-programming-at-a-young-age-2>.
Página Vigente al 18/04/18
- Guía del Estudiante. Dónde estudiar Programación en Argentina,
<http://www.estudios.com.ar/estudiar-programacion.html?q=Programaci%C3%B3n>. Página Vigente al 18/04/18

- Biblioteca Argentina para Ciegos (BAC), Informática.
<https://www.bac.org.ar/informatica.html>. Página Vigente al 19/04/2018
- Christensson, P. (2006). Plug-in Definition. Retrieved 2018,
<https://techterms.com/definition/plugin>. Página Vigente al 19/04/2018
- Mideplan, Guía para la Elaboración de Diagramas de Flujo, Julio 2009,
<https://documentos.mideplan.go.cr/alfresco/d/d/workspace/SpacesStore/6a88ebe4-da9f-4b6a-b366-425dd6371a97/guia-elaboracion-diagramas-flujo-2009.pdf>.
[Página Vigente al 27/04/2018](#)
- Escuela de Educación Especial N°7040 "Corina Lona".
<http://corinalona7040.blogspot.com.ar/> . Página Vigente al 09/05/2018
- Ministerio de Salud y Servicio Nacional de Rehabilitación, Anuario Estadístico Nacional. <https://www.snr.gob.ar/wp-content/uploads/2017/08/Anuario-2016-FINALRE.pdf>. Página Vigente al 09/05/2018

Anexo I

Diagrama de Flujo

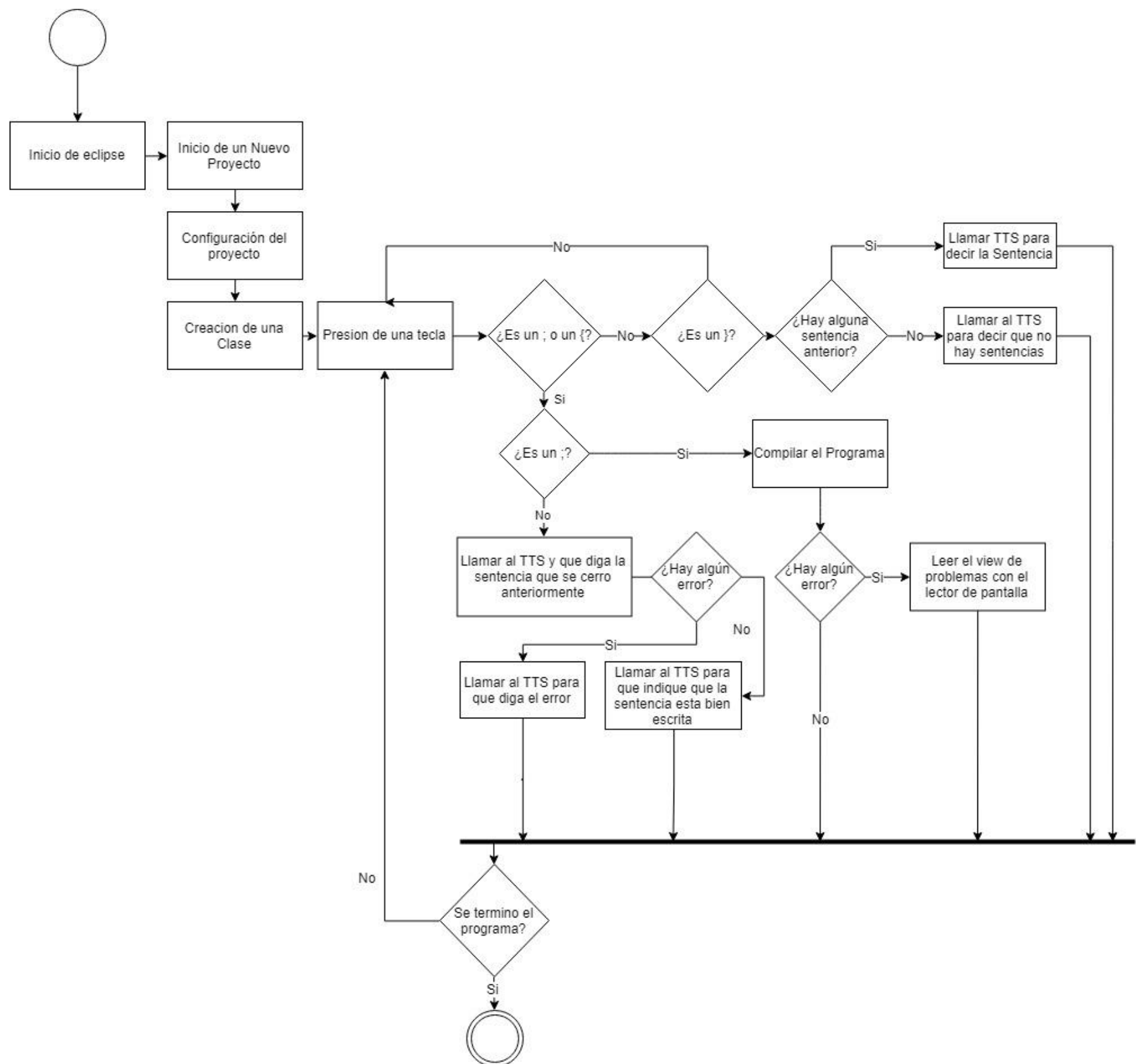


Imagen 9 - Diagrama de Flujos

Modelado del Diseño

Diagrama de Clases

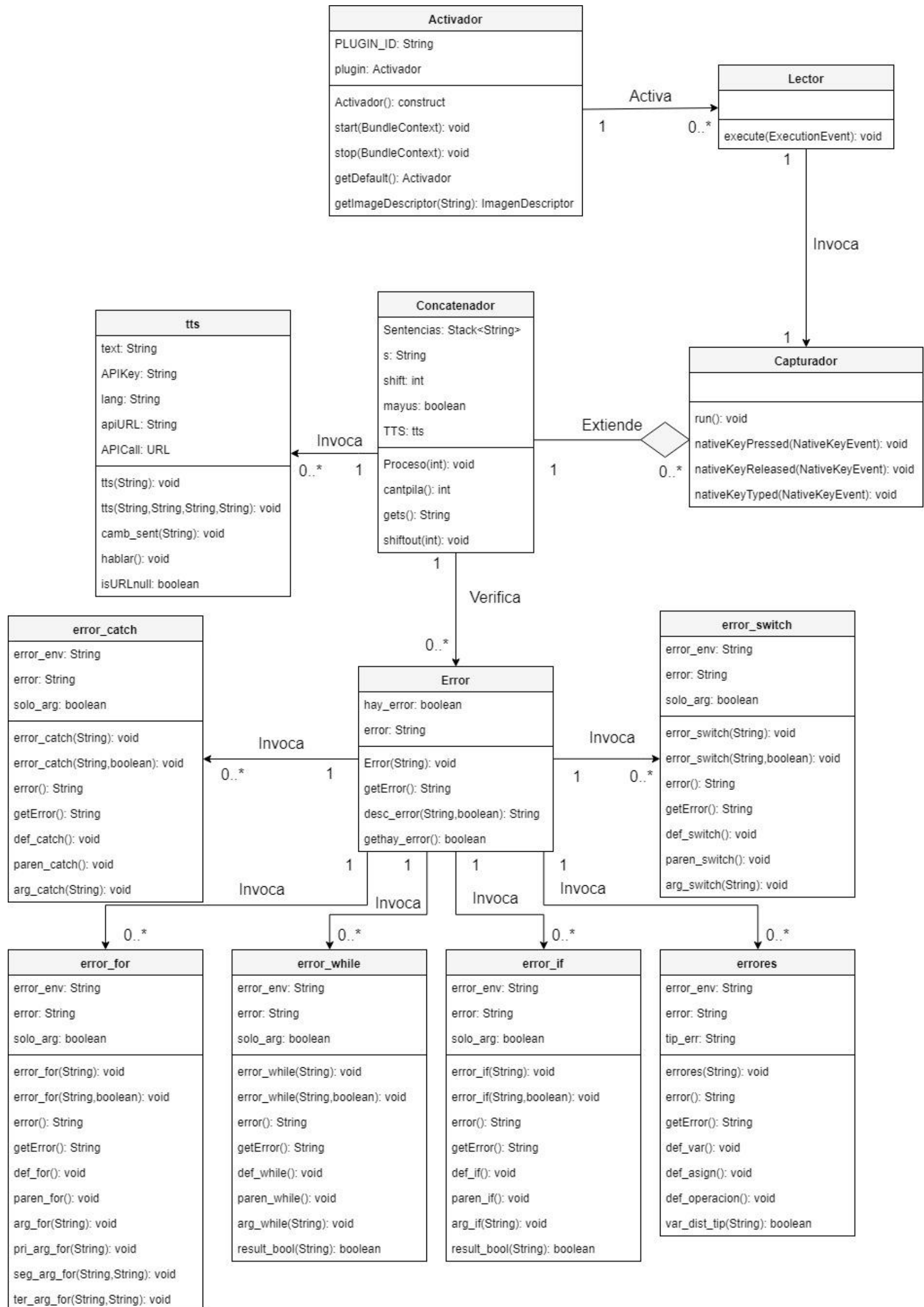


Imagen 10 - Diagrama de Clases

Diagrama de estado-transición

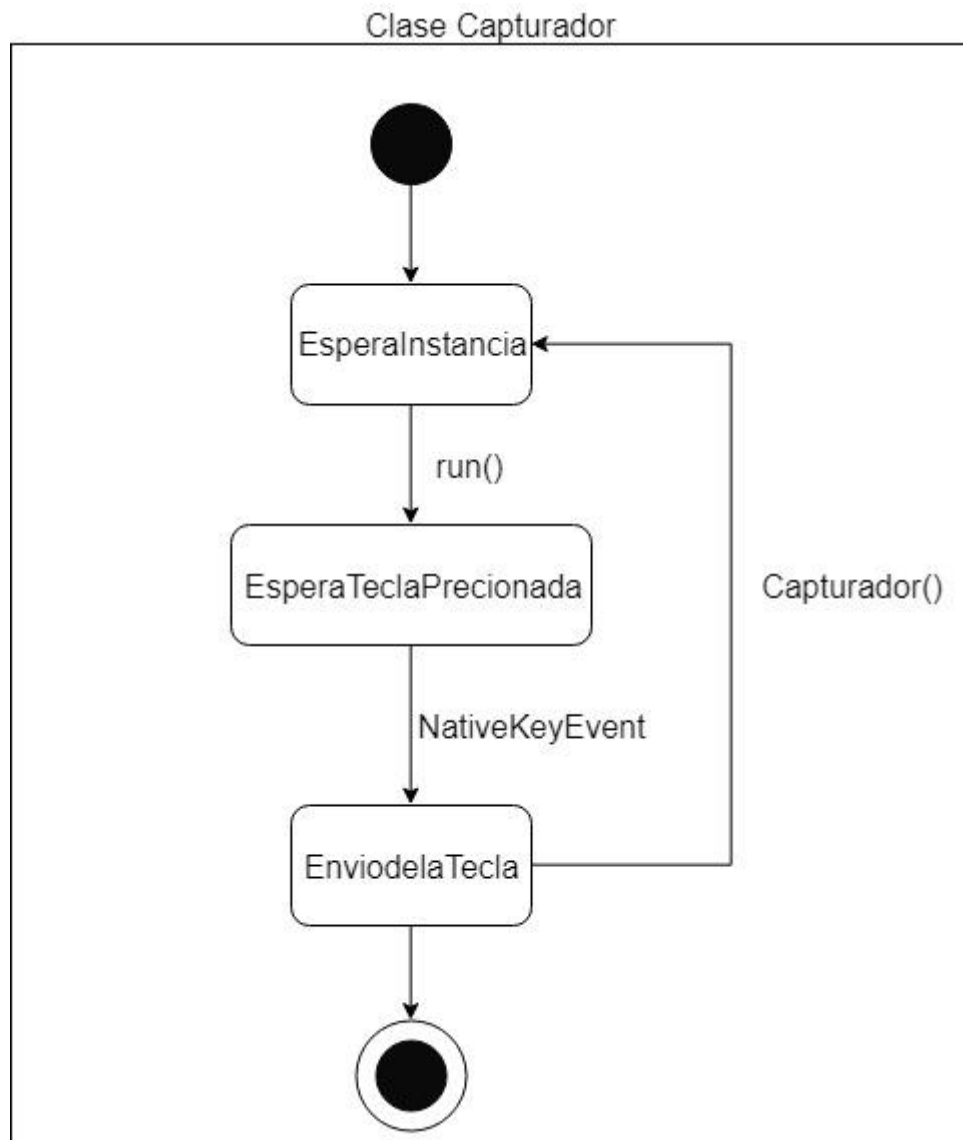
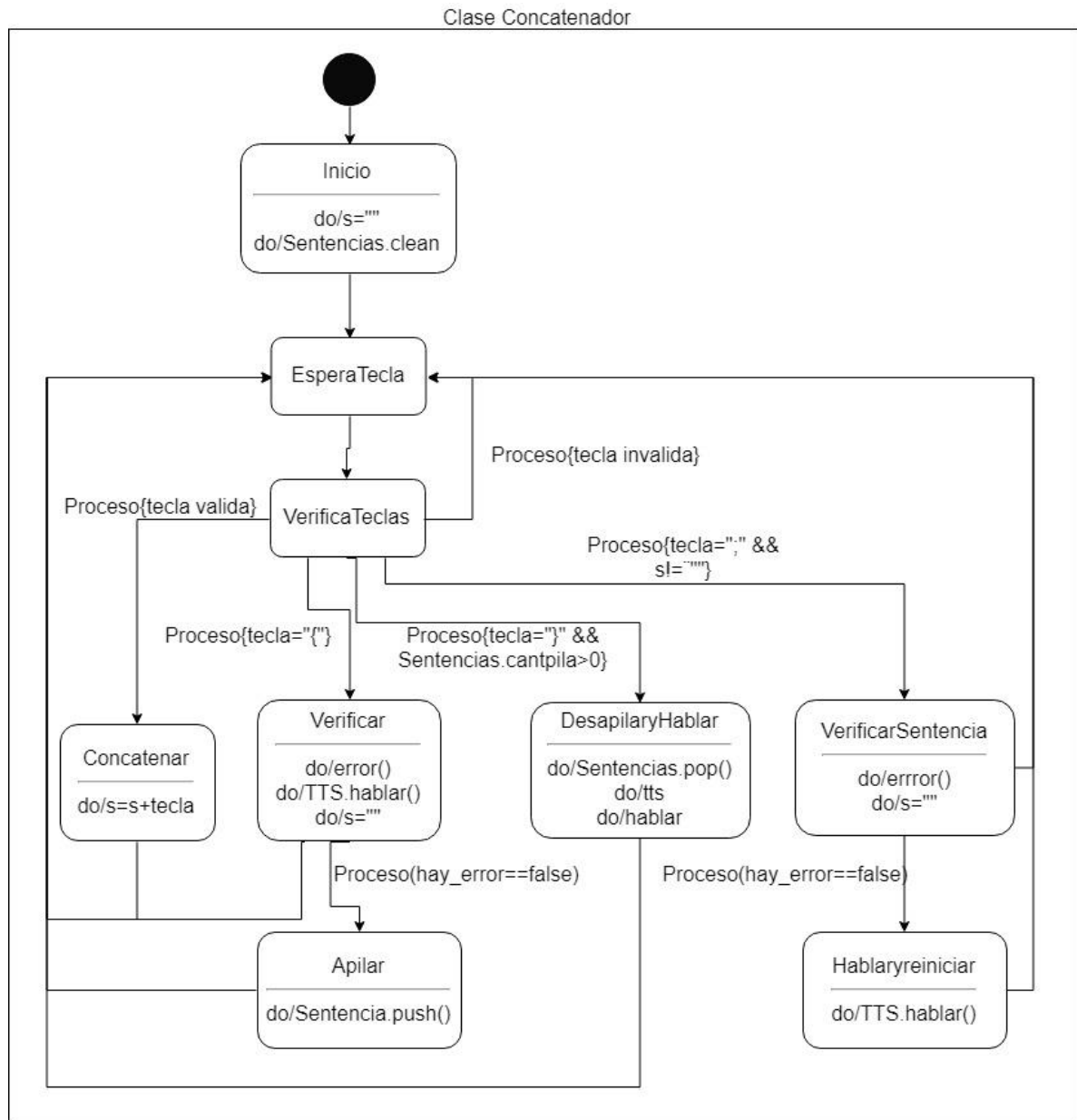


Imagen 11 - Diagrama de Estado-Transición Clase Capturador



Teclas Validas: Todas las letras del abecedario, números y símbolos

Teclas Invalidas: Código UTF-8 para los f1 al f12, bloqueo de mayúsculas, etc.

Imagen 12 - Diagrama de Estado-Transición Clase Concatenador

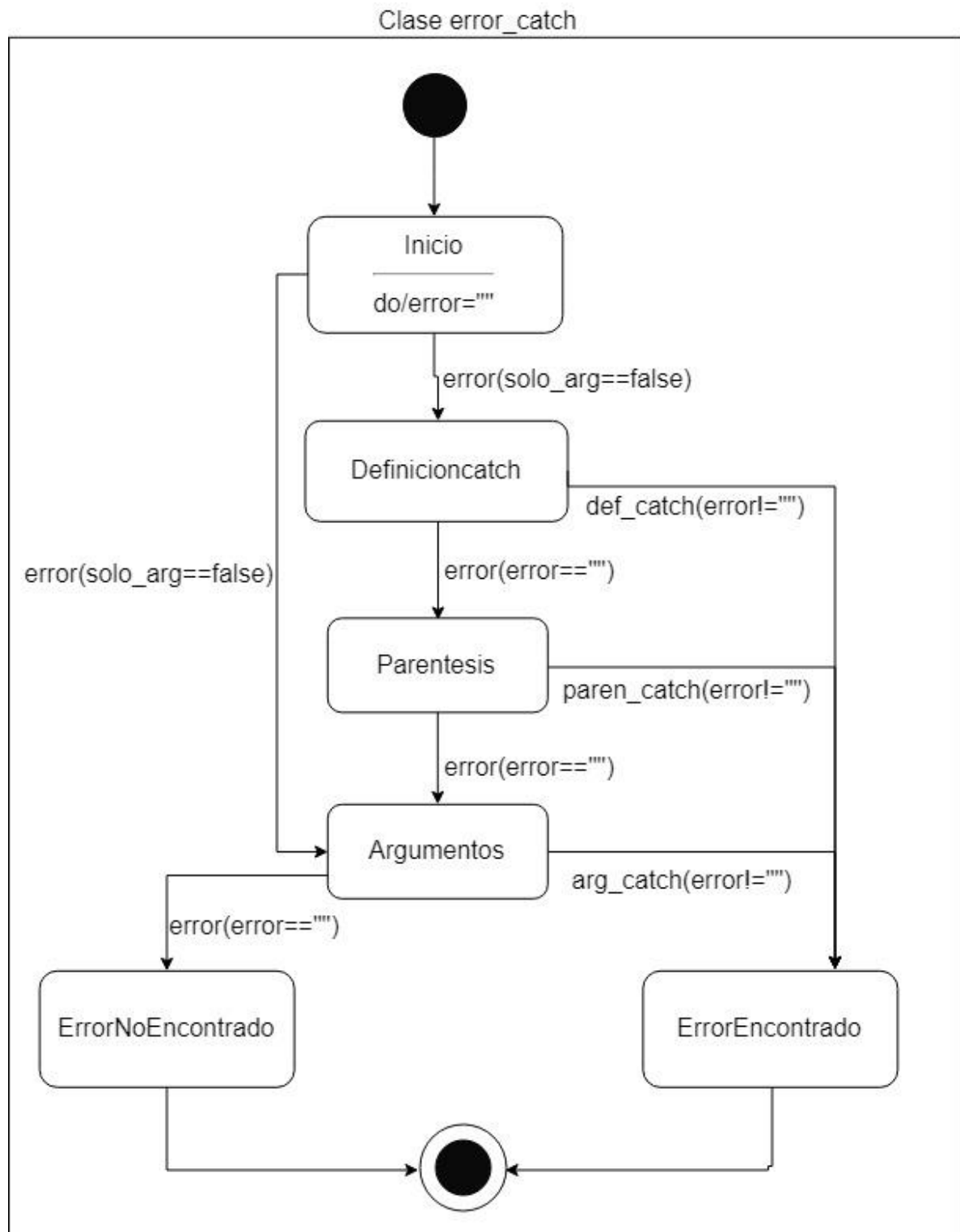


Imagen 13 - Diagrama de Estado-Transición Clase error_catch

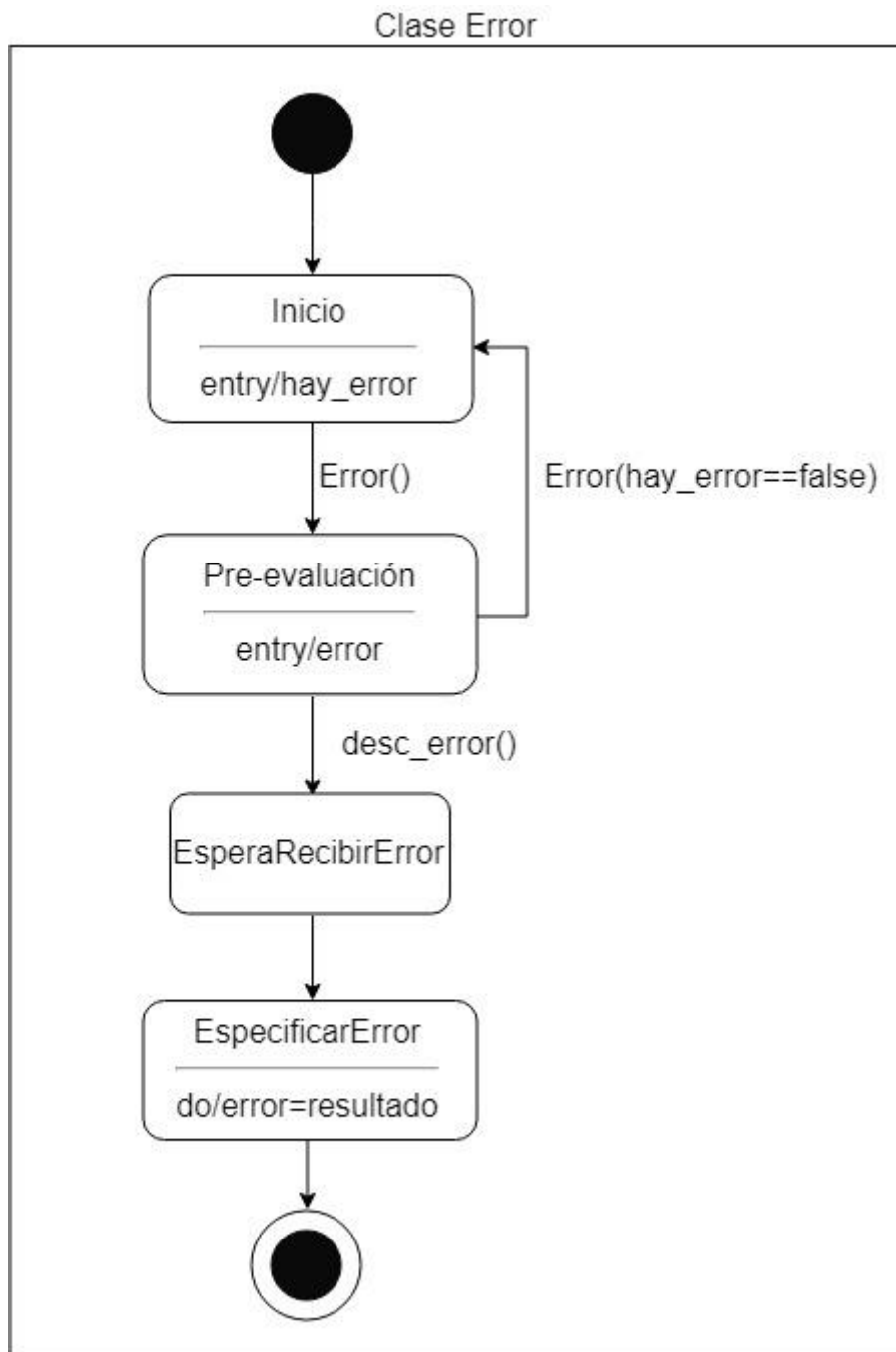


Imagen 14 - Diagrama de Estado-Transición Clase Error

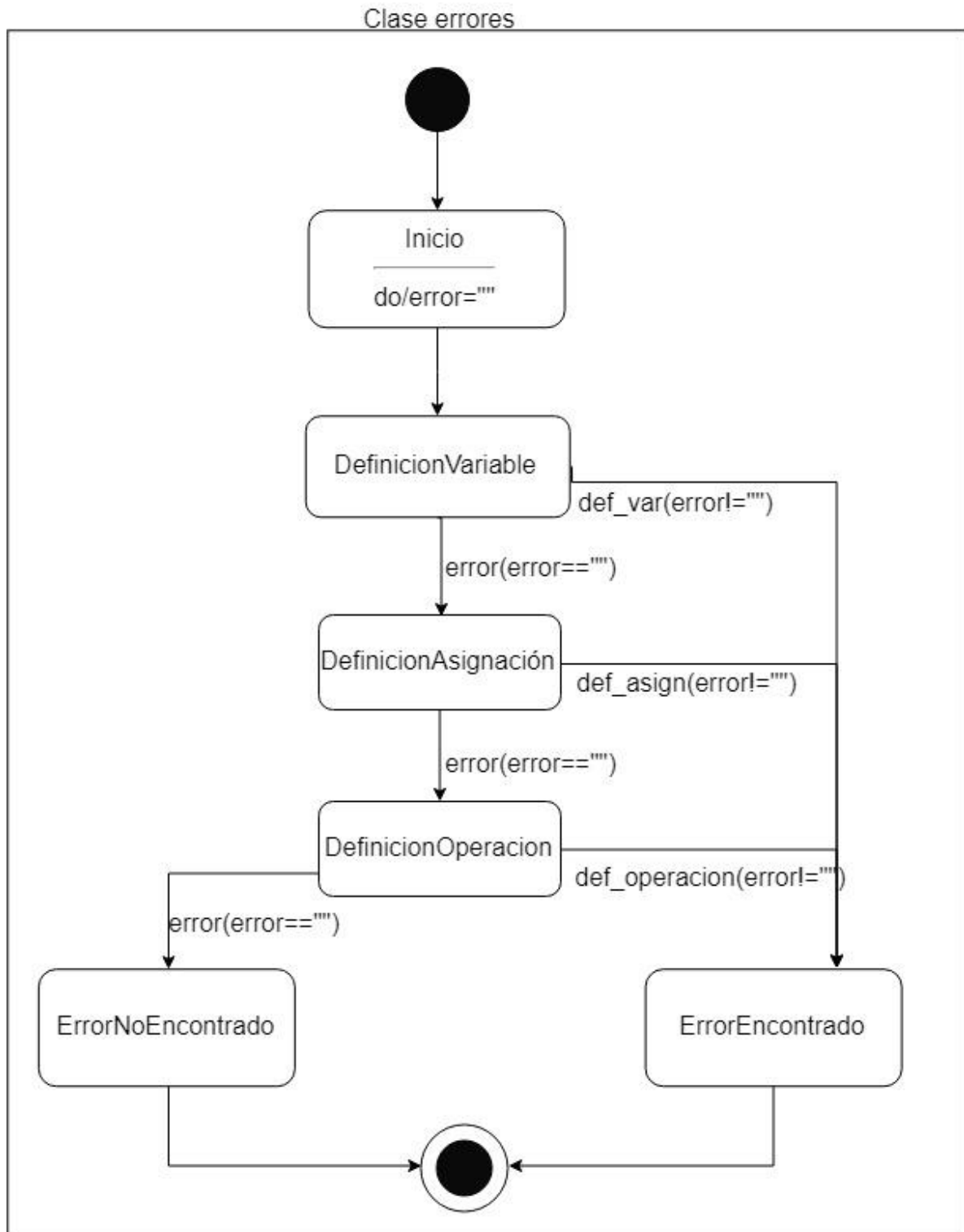


Imagen 15 - Diagrama de Estado-Transición Clase errores

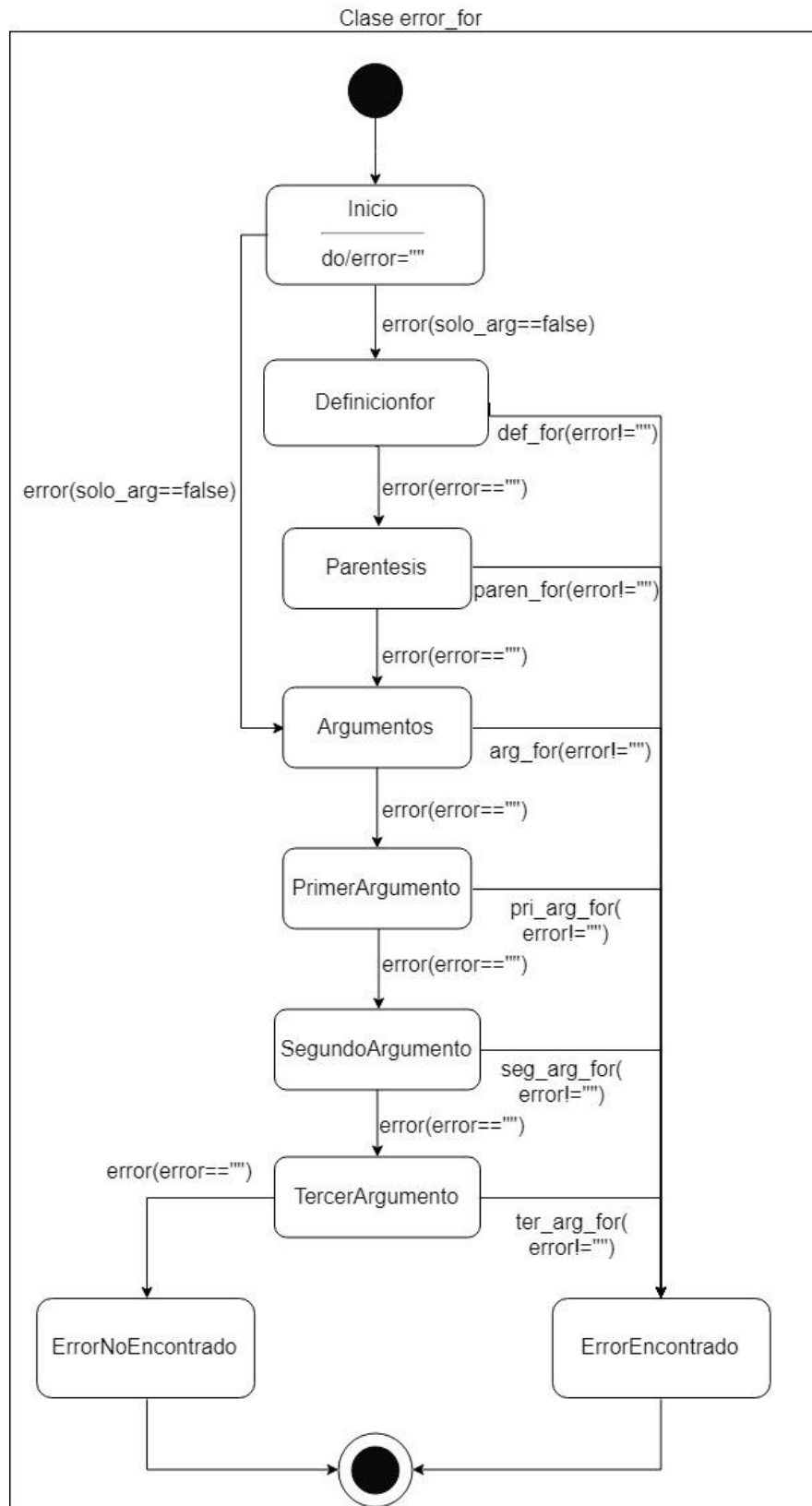


Imagen 16 - Diagrama de Estado-Transición Clase error_for

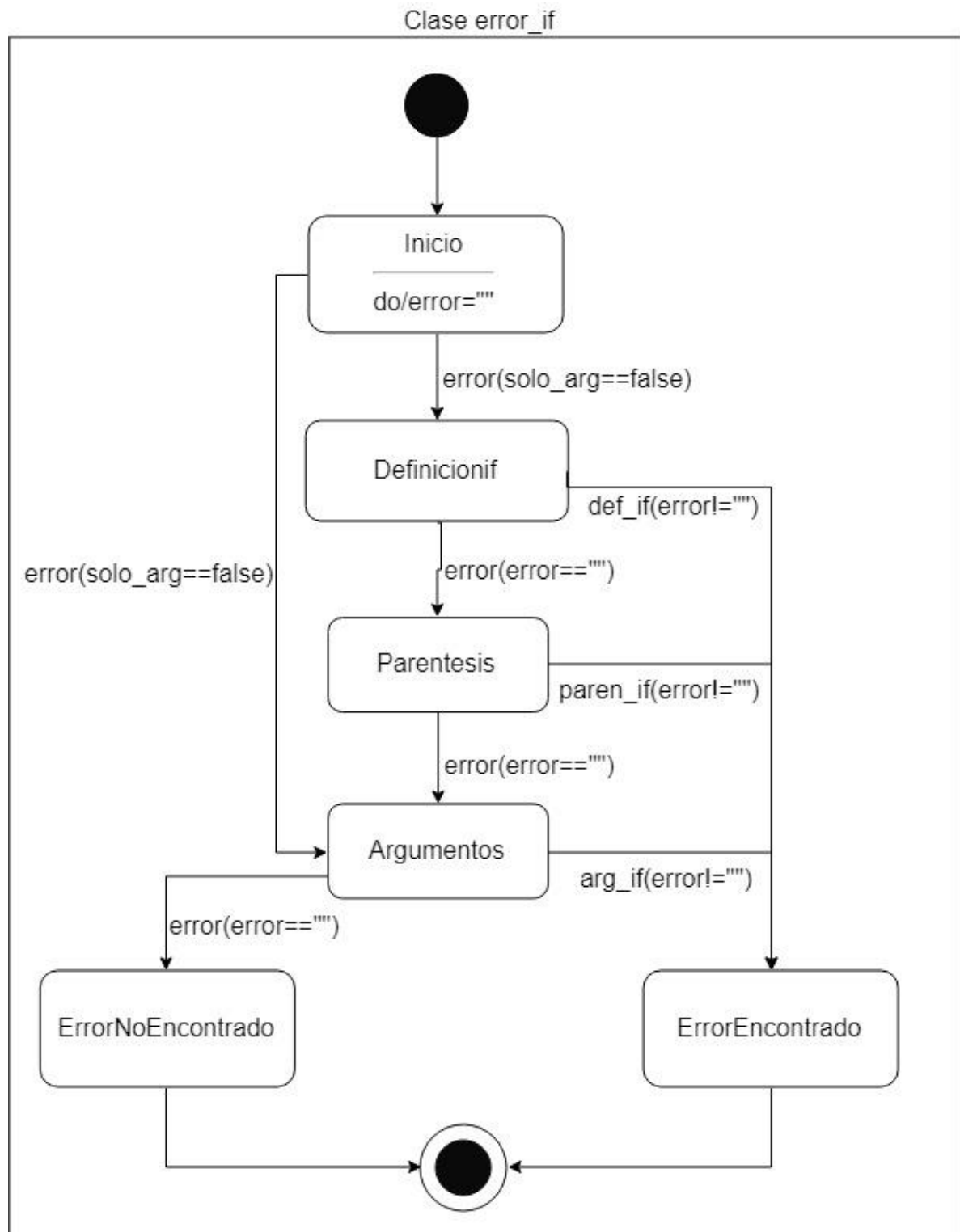


Imagen 17 - Diagrama de Estado-Transición Clase error_if

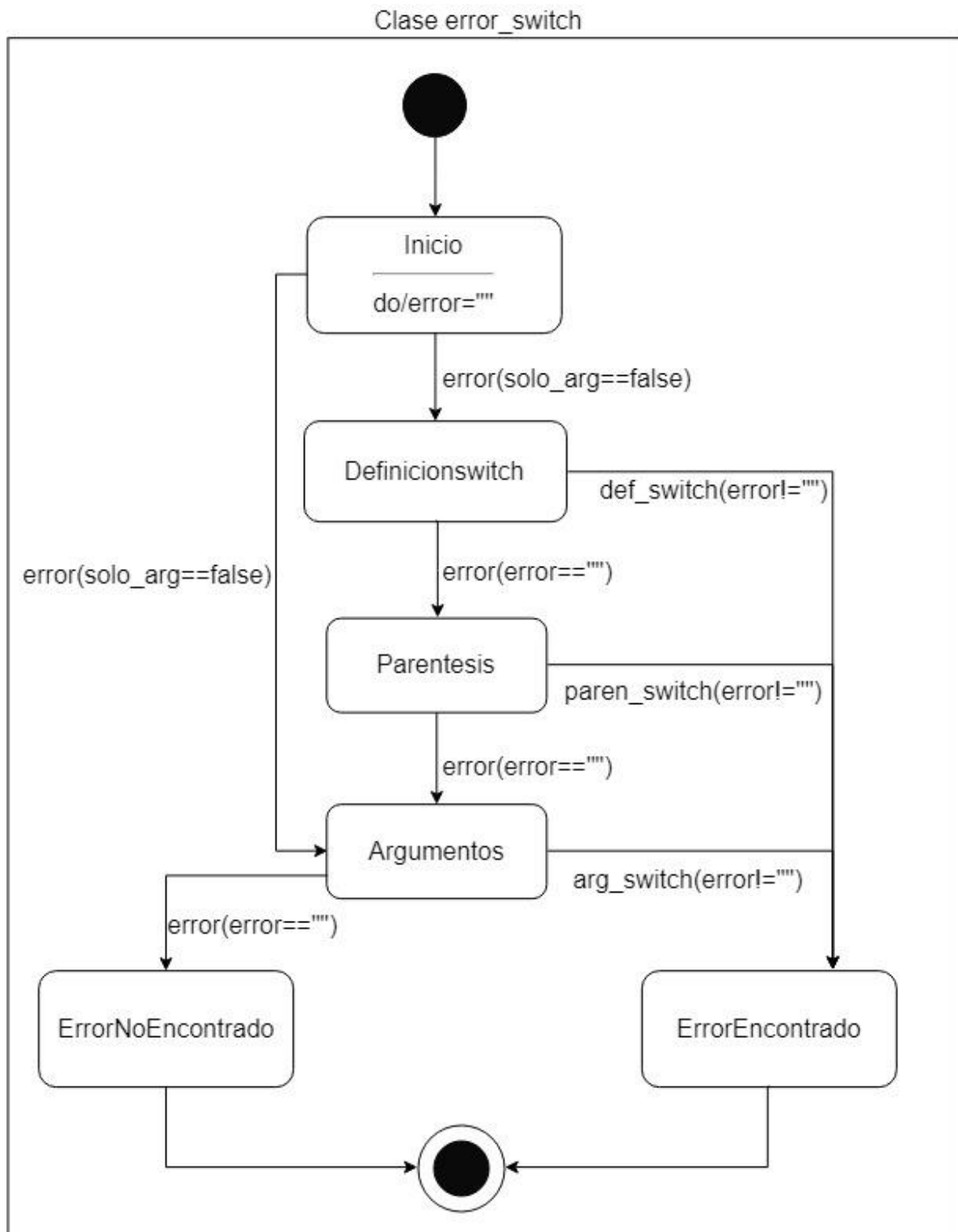


Imagen 18 - Diagrama de Estado-Transición Clase error_switch

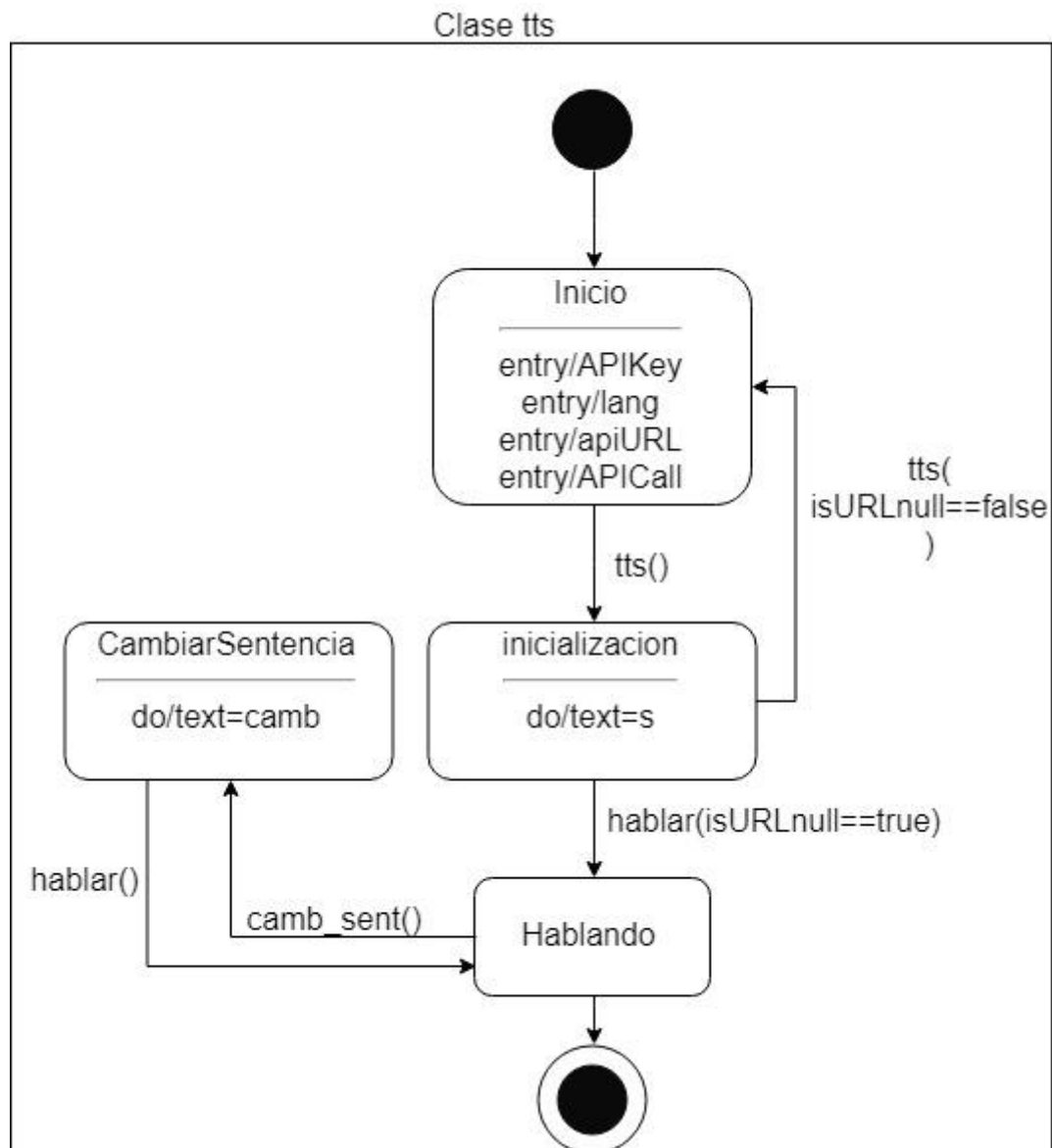


Imagen 19 - Diagrama de Estado-Transición Clase tts

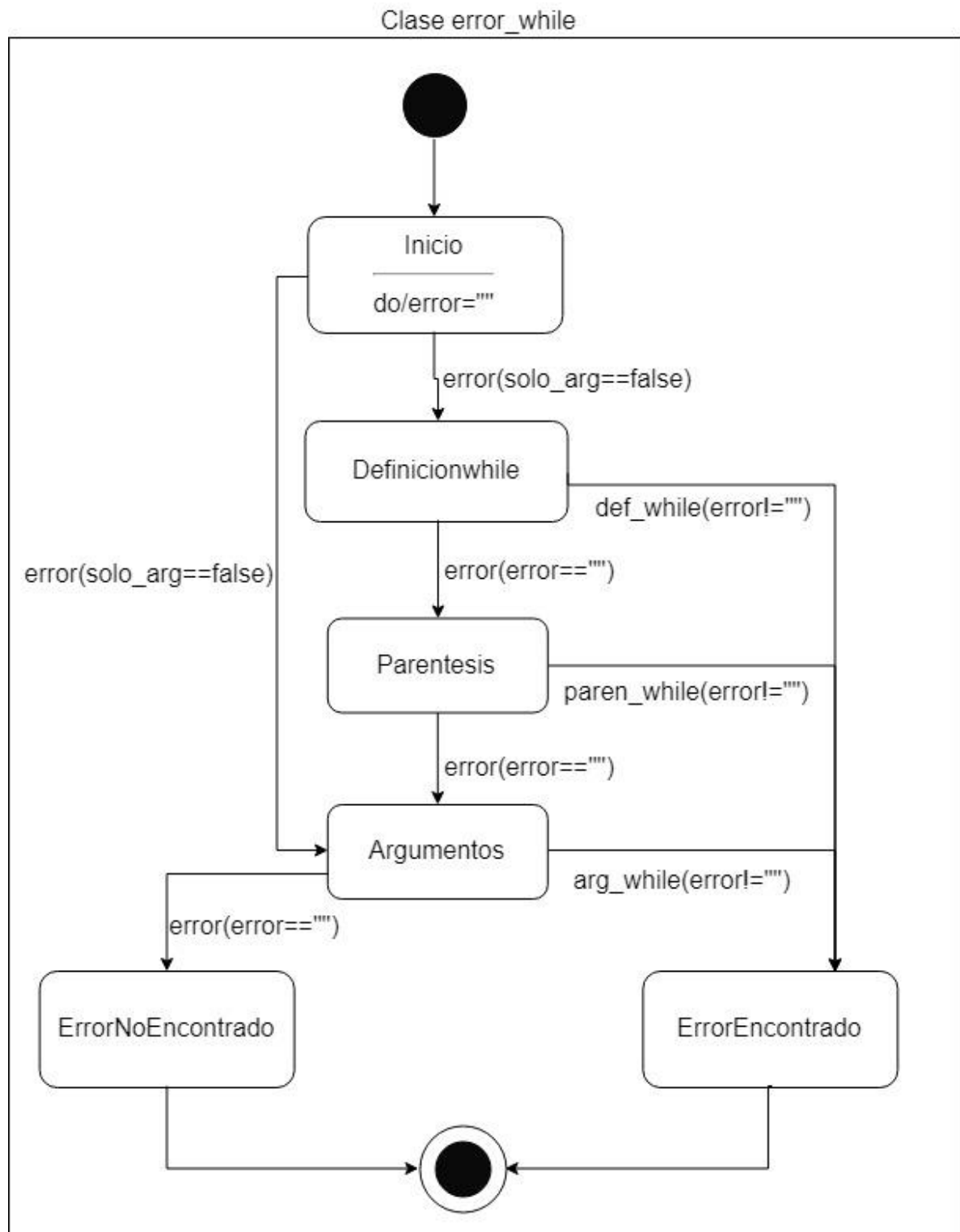


Imagen 20 - Diagrama de Estado-Transición Clase error_while

Diagrama de Actividades

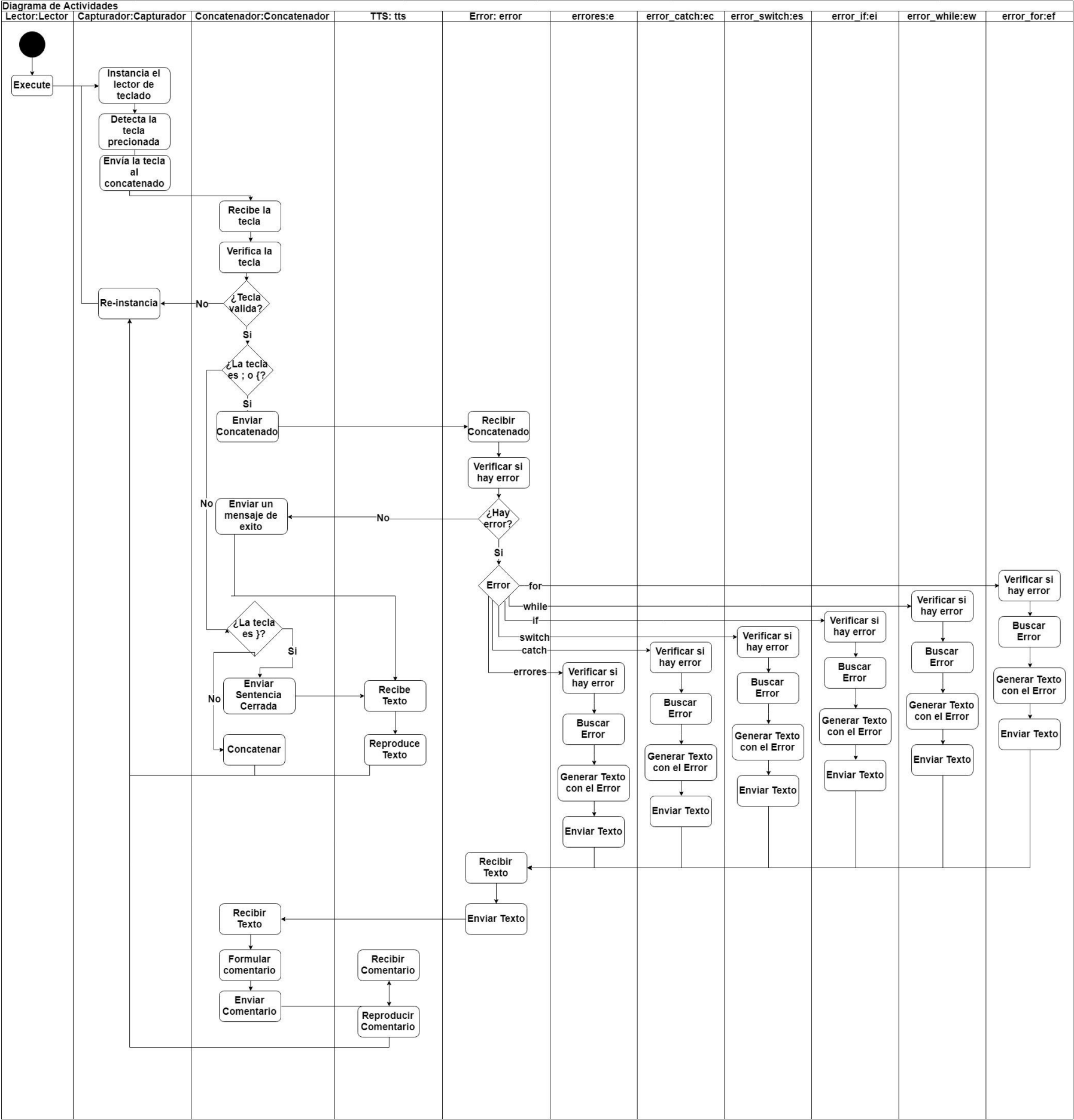


Imagen 21 - Diagrama de Actividades

Diagrama de Componentes

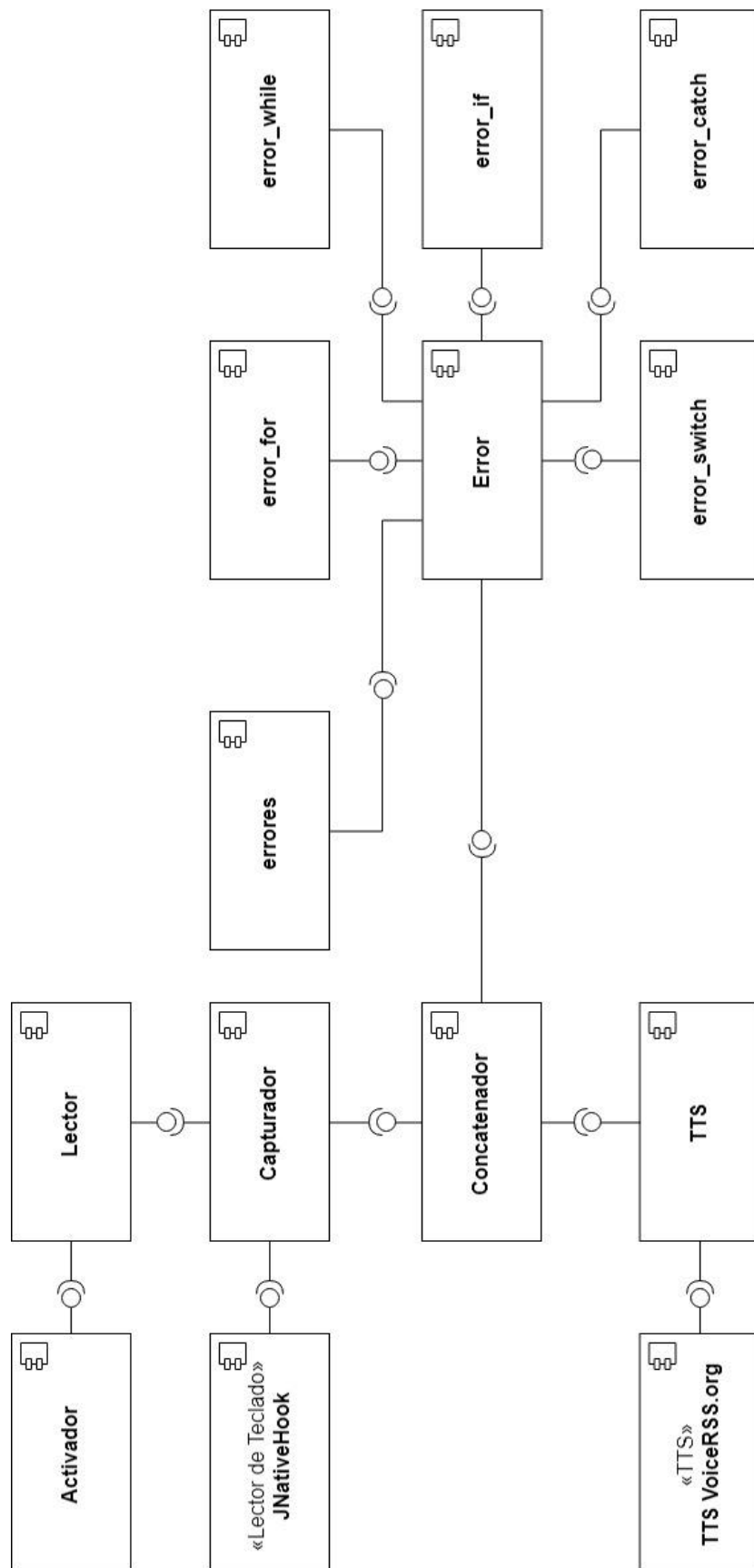


Imagen 22 - Diagrama de Componentes