PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

SPEECH TIMER

ISRAEL MAXIMILIANO OGAS VEGA

ISRAEL ALEJANDRO JASMA CÁCERES

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

DICIEMBRE, 2018

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

SPEECH TIMER

ISRAEL MAXIMILIANO OGAS VEGA

ISRAEL ALEJANDRO JASMA CÁCERES

Profesor Guía: **Iván Mercado Bermúdez**

Profesor Co-referente: **Ismael Figueroa Palet**

Carrera: **Ingeniería de Ejecución en Informática**

DICIEMBRE, 2018

Índice

Resumen ..... **iii**
Abstract ..... **iii**
Lista de Figuras ..... **iv**
Lista de Tablas ..... **v**
1 Introducción **1**
2 Objetivos **2**
2.1 Objetivo General **2**
2.2 Objetivos Específicos **2**
3 Situación en Estudio **3**
3.1 Contexto Organizacional **3**
3.2 Descripción de la situación actual **3**
3.3 Identificación de los problemas detectados **4**
3.4 Estado del arte del tema **4**
4 Propuesta Solución **5**
4.1 Antecedentes generales de la solución **5**
4.2 Modelo de Proceso de desarrollo **5**
4.3 Arquitectura **6**
4.3.1 Física 6
4.3.2 Lógica 7
4.3.2.1 Cliente 7
4.3.2.1 Servidor 7
4.4 Herramientas de Desarrollo **7**
5 Gestión del Proyecto **9**
5.1 Recursos **9**
5.1.1 RRHH 9
5.1.2 Recursos Físicos 9
5.2 Gestión del Riesgo **9**
5.2.1 Matriz de Probabilidad e Impacto 10
5.2.2 Riesgos del Proyecto 10
5.2.2.1 Riesgo del Proyecto 10
5.2.2.2 Riesgo Tecnológico 11
5.2.2.3 Riesgo del Negocio 11

5.3 Plan de trabajo 12
5.3.1 Marco General del proyecto 12
5.3.2 Análisis y Diseño del proyecto 12
5.3.3 Desarrollo y Codificación de la aplicación 12
5.3.4 Pruebas y Resultados 12
6 Desarrollo de proyecto 13
6.1 Identificación de Requerimientos 13
6.1.1 Requerimientos funcionales 13
6.1.2 Requerimientos no funcionales 15
6.2 Análisis de requerimientos 16
6.3 Diseño 16
6.3.1 Comandos Cliente-Servidor 16
6.3.2 Interfaz 17
6.3.2.1 Mockups de Interfaz del Moderador 18
6.3.2.2 Mockups de Interfaz del Expositor 20
6.3.2.3 Interfaz del prototipo funcional del moderador y expositor 22
7 Resultados obtenidos 23
8 Conclusiones 24
9 Referencias bibliográficas 25
Anexos 26

Resumen

Todos los años en la PUCV se realizan actividades extra programáticas, entre ellas está el Claustro Pleno, evento por el cual cobra sentido la realización del presente proyecto. Para la ejecución de este, es necesario contar una gran preparación que va desde la escenografía hasta los organizadores que deben controlar las entradas y salidas de los expositores, así como su tiempo en el escenario exponiendo su discurso, para esto último se usa un software propio. El actual software está quedando obsoleto tecnológicamente por el diseño de cómo fue creado, para ello se debe realizar una nueva versión del software mejorando una gran cantidad de inconvenientes que hay debido a la obsolescencia de la tecnología usada hace algunos años atrás.

Palabras-claves: Temporizador de voz, Android, aplicación móvil, discurso, Claustro Pleno.

Abstract

Every year in the PUCV extra programmatic activities are carried out, among them is the Plenary Cloister, an event for which the realization of the present project makes sense. For the execution of this one, it is necessary to count on a great preparation that goes from the scenography to the organizers that must control the entrances and exits of the exhibitors, as well as their time in the scene exposing their speech, for this last one an own software is used. The current software is becoming technologically obsolete due to the design of how it was created. To do this, a new version of the software must be made, improving a large number of inconveniences due to the obsolescence of the technology used some years ago.

Keywords: Speech Timer, Android, Mobile application, speech, Full Cloister.

Lista de Figuras

Figura 1 Diagrama del modelo de prototipo en el proceso de desarrollo 6
Figura 2 Diagrama de Casos de Uso del software 16
Figura 3 Pantalla de Inicio – Home (Moderador) 18
Figura 4 Configurar tiempo de exposición 18
Figura 5 Pantalla de mensajes predefinidos al expositor 19
Figura 6 Pantalla de ajustes 19
Figura 7 Pantalla del cronómetro en funcionamiento (Moderador) 20
Figura 8 Pantalla de inicio-home (Expositor) 20
Figura 9 Pantalla del cronómetro en funcionamiento (Expositor) 21
Figura 10 Pantalla del cronómetro con avisos 21
Figura 11 Pantalla de Inicio (prototipo funcional) 22
Figura 12 Pantalla del cronometro (prototipo funcional) 22
Figura 13 Pantalla mensaje de aviso al pausar 26
Figura 14 Pantalla del menú general 26
Figura 15 Pantalla mensaje de aviso al finalizar tiempo 27
Figura 16 Cuadro de dialogo con alertas predefinidas 27

Lista de Tablas

Tabla 1 Métricas para evaluar probabilidad de riesgo e impacto 10
Tabla 2 Métrica base de evaluación de riesgo 10
Tabla 3 Riesgos del proyecto y plan de mitigación 11
Tabla 4 Riesgos tecnológicos y plan de mitigación 11
Tabla 5 Riesgos del negocio y plan de mitigación 11
Tabla 6 Requerimiento funcional alerta de tiempo expositor 13
Tabla 7 Requerimiento funcional color en pantalla para el expositor 13
Tabla 8 Requerimiento funcional término de exposición 13
Tabla 9 Requerimiento funcional controlar el tiempo 14
Tabla 10 Requerimiento funcional definir tiempo para el expositor 14
Tabla 11 Requerimiento funcional definir alertas para el expositor 14
Tabla 12 Requerimiento funcional mensajes para el expositor 14
Tabla 13 Requerimiento funcional intermedio al expositor (pausa) 14
Tabla 14 Requerimientos no funcionales 15
Tabla 15 Comandos de la aplicación cliente-servidor 17

1 Introducción

En este momento se debe estar exponiendo o realizando discursos [1] en múltiples lugares del mundo, esto conlleva mucho trabajo; desde preparar la escenografía hasta el tiempo que debe exponer cada participante. Ahora, enfocándonos en el tiempo, éste es muy importante debido a que para dar una conferencia, exponer, etc. debe estar medido y controlado por múltiples factores que le puedan afectar de manera directa o indirecta. Hoy en día existen innumerables aplicaciones para Android, iOS y Escritorio que regulan el tiempo, ya sea para uso personal [2] o bien que alguien lo regule para un grupo de expositores [3]. Dicho esto, en el Salón de Honor de la Casa Central todos los años se realiza el Claustro Pleno [4], instancia donde se reúnen invitados especiales y representantes de los distintos estamentos que conforman la comunidad universitaria. Los participantes de esta importante instancia, exponen sus apreciaciones, consultas acerca de la cuenta rendida y la situación de la Casa de Estudios frente a los asistentes. Actualmente la PUCV posee un software para medir el tiempo de cada expositor, el que fue desarrollado hace algunos años. Al ser así, hoy este software a pesar de que funcione de manera correcta, éste llegó a ser una carga, porque requiere tecnología muy espaciosa la que debe ser siempre trasladada, además de tener que usar ciertas configuraciones a priori para dicho funcionamiento.

Hoy se ha dado la problemática de querer cambiar el software para que ya no posea los inconvenientes que presenta actualmente, pero aun así dado las innumerables aplicaciones que puede haber, no se ha encontrado uno que se adapte a lo que realmente se necesita. Entonces, para ello se le ha propuesto a unos estudiantes de la escuela de informática perteneciente a la PUCV desarrollar un software a la medida, que se pueda adaptar a las diversas plataformas que existen actualmente, reduciendo las herramientas necesarias que se necesitan para su correcto funcionamiento, con un buen grado de usabilidad y, principalmente, sea realmente útil para el usuario objetivo.

2 Objetivos

A continuación, se describen los objetivos, tanto principales como específicos del proyecto, evidenciando la finalidad y enfoque a la solución que se propone.

2.1 Objetivo General

Desarrollar una aplicación móvil basada en la arquitectura cliente-servidor, que reemplace el software actual que se utiliza en la convocatoria del Claustro Pleno de la PUCV [3], accediendo a un servidor web a través de internet mediante un navegador. Este debe tener mejoras sustanciales con respecto al actual y brindar las herramientas necesarias al usuario encargado de controlar los tiempos para las exposiciones que se realicen en este acto importante. Además, debe ser flexible y fácil de usar para el usuario expositor.

2.2 Objetivos Específicos

Para la realización del objetivo general, es necesario especificar cada objetivo concreto y visible de la solución implementada en el proyecto. Estos son:

- Seleccionar el ambiente, herramientas y metodología que mejor se adapte para el desarrollo de la aplicación para cronometrar discursos.
- Desarrollar la aplicación que controle el tiempo de discurso para los expositores utilizando una metodología de desarrollo de software que mejor se adapte al problema.
- Elaborar las pruebas del software en un ambiente y condiciones similares en las que se utilizará en el Claustro Pleno.

3 Situación en Estudio

A continuación, se presentan los detalles de dominio y descripción de los problemas relevantes observados para el desarrollo del sistema, así como el equipo desarrollador de éste.

3.1 Contexto Organizacional

PUCV es una universidad tradicional, privada de chile, la institución a través de sus nueve facultades situados principalmente en la ciudad de Valparaíso siendo Casa Central la principal sede ubicado en dicho lugar, y sus alrededores tales como Viña del mar, Placilla Peñuelas y Quillota. También abarca una amplia área del saber que van desde las artes y las humanidades hasta la ingeniería, pasando por las ciencias básicas, la educación y los estudios tecnológicos. Esta entidad educativa realiza diversas actividades extra curriculares, dando un ejemplo sería el Claustro Pleno, actividad que se realiza todos los años. Esta es una convocatoria en la que participan diversos expositores de distintas áreas de la comunidad universitaria, reuniéndose y presentando diversos temas los que gozan de un interés ya sea para ellos mismo como para los oyentes. Adicionalmente, esta actividad es una exposición organizada y estructurada de carácter formal realizada una vez al año. Dado esto, se requiere de una gran preparación tanto por la infraestructura (adornos, asientos, micrófonos, etc.) hasta el control de todos los tiempos entre cada expositor, siendo aquí donde se centra el presente proyecto que pretende actualizar un software que utiliza la PUCV para dicha actividad.

El cliente del proyecto es un miembro de una de las escuelas que posee la PUCV, siendo esta la Escuela de Ingeniería en Informática. Iván Mercado es el jefe de docencia, además de tomar un rol activo en el diseño de la aplicación. Él posee títulos académicos en el área de la informática.

3.2 Descripción de la situación actual

La PUCV realiza todos los años la convocatoria del Claustro Pleno, convocada por el Consejo Superior de la PUCV y el rector para anunciar la cuenta anual y expresar su parecer respecto a asuntos de carácter universitario. Esta instancia reúne a académicos, alumnos y funcionarios, ya sean pertenecientes a dicha institución o invitados, los cuales pueden ser partícipes de esto para formular apreciaciones, realizar consultas acerca de la cuenta rendida o la situación de la Casa de Estudios.

Para controlar el tiempo que expone cada participante, se usar un software propio, con el fin de regular los tiempos que tiene cada expositor frente a la ceremonia que se realiza en el Salón de Honor de la Casa Central, ubicado en Avenida Brasil 2950, Valparaíso, Chile, perteneciente a la PUCV.

3.3 Identificación de los problemas detectados

El software que actualmente posee la PUCV (desarrollado por la escuela de ingeniería en informática hace algunos años) presenta algunos problemas de portabilidad, dado a su diseño y la manera en que fue desarrollado. Este necesita de un equipo portátil, ya sea una laptop, un router inalámbrico y una tablet o similar para su funcionamiento en la ceremonia del Claustro Pleno.

Otros de los problemas detectados fue el siguiente; la herramienta necesita de una red local para su funcionamiento (router) y no tiene otra alternativa de conexión como podría ser bluetooth en caso de que fallara dicha conexión. Adicionalmente, esta no emite ningún sonido al momento de cambiar de color el cronómetro, lo cual es esencial para que el expositor no deba estar pendiente en todo momento sobre el tiempo que le va quedando a medida que realiza su discurso.

3.4 Estado del arte del tema

En la actualidad existen varias aplicaciones que se encargan de cronometrar diferentes tipos de eventos, exposiciones, charlas, sucesos importantes, actividades diarias, etc. Entre esta variedad de usos que se le da a este tipo de software, se encuentran los de uso personal y los de uso grupal (ej. Situémonos en el contexto de una charla, donde alguien externo configura el cronómetro por ti mientras tú la estás utilizando de manera indirecta verificando el tiempo que llevas exponiendo).

De las herramientas existentes se pueden encontrar alternativas para todo tipo de sistemas operativos, ya sean IOS, Android, Windows Phone, entre otros. Además de poder mencionar alguna como “Speech Timer”, “Timer4TM”, “Time Manager”; sin mencionar que algunas son gratuitas y de paga.

El software que posee actualmente, es un sistema propio de la PUCV, el que fue desarrollado por la escuela de ingeniería en informática de dicha institución. A pesar de ser una herramienta propia es necesario actualizarla por los problemas y/o motivos mencionados anteriormente.

4 Propuesta Solución

Se especifican las metodologías y herramientas de desarrollo para generar una solución a los problemas identificados en el punto anterior.

4.1 Antecedentes generales de la solución

El sistema a implementar pretende que el flujo de datos sea de la siguiente manera:

- El moderador (persona a cargo de regular el tiempo de cada conferencia/exposición/discurso) deberá crear una 'sala de eventos' ingresando el nombre de esta para poder identificarla y la cantidad de usuarios que podrán entrar a la sala (esto de limitar la cantidad de personas se hace para evitar que otros intenten acceder a la sala para molestar o para que no saturen la conexión). Al crearla, el moderador debe establecer el tiempo que le dará al expositor para realizar su conferencia, indicando la *alerta inicial y alerta secundaria*.
- El moderador, al enterarse que el expositor ya está dentro de la sala, iniciará el cronómetro en cuenta atrás y al mismo tiempo el expositor podrá ver el tiempo avanzar. Entonces, en ese momento esta persona podrá empezar su discurso.
- En cualquier momento el moderador podrá pausar, detener (si se realiza alguna de estas acciones, se dará un margen de tiempo para avisarle al expositor dicha acción) o reiniciar el cronómetro en caso de que ocurra cualquier percance, además podrá enviarle mensajes predefinidos al expositor y mandar alertas sonoras al expositor por medio del dispositivo
- El expositor a medida que se avance el cronómetro, cuando se active la *alerta secundaria,* la interfaz en donde se muestra el tiempo cambiará de color y se escuchará una pequeña alerta sonora.

4.2 Modelo de Proceso de desarrollo

La metodología a utilizar es fundamental ya que indica el modo en que se desarrollará el software la cual es específica para cada producto en particular. Es por esto que, según las consideraciones que se presentarán a continuación, el proceso de desarrollo de software a utilizar será el modelo de prototipo, mejor conocido como modelo de desarrollo evolutivo.

Las ventajas de utilizar el modelo de prototipo, es el hecho de que durante la elaboración del software o aplicación se van construyendo prototipos, los cuales deben ser creados en poco tiempo y con pocos recursos. Todo esto se hace con el fin de obtener el feedback de los stakeholders [5] o usuarios para realizar un siguiente prototipo con los cambios necesarios con la finalidad de llegar al óptimo deseado.

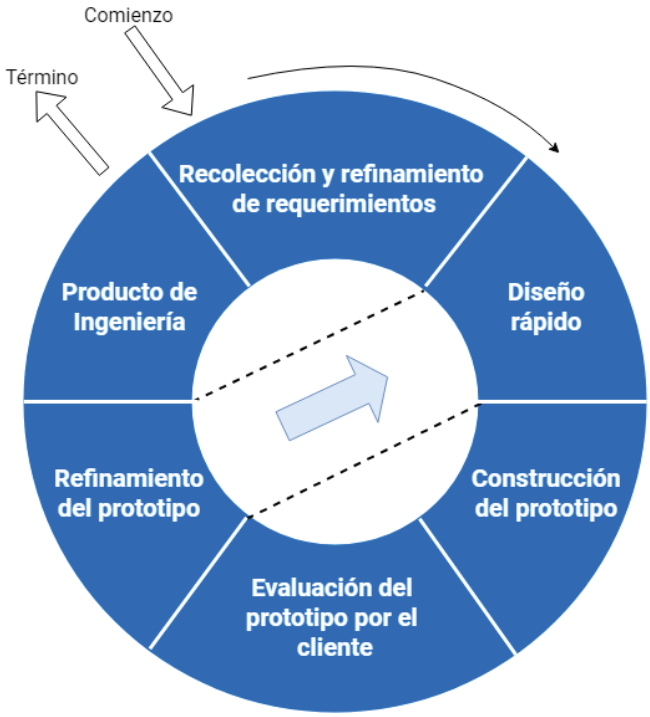

Figura 1 Diagrama del modelo de prototipo en el proceso de desarrollo

Como se puede apreciar en la figura 1 (diagrama replicado desde el sitio de Ecured [6]), podemos observar que después de la recolección y refinamiento de los requerimientos comienza el ciclo del prototipado, en donde se parte con un bosquejo para crear el prototipo, luego se construye el prototipo diseñado para ser evaluado por el cliente y por último se realiza el proceso de refinamiento del prototipo, ya sea desechando el existente y creando uno nuevo o la modificación de elementos puntuales de este.

4.3 Arquitectura

Se definen las capas y niveles del proyecto para su correcta implementación.

4.3.1 Física

La arquitectura física está compuesta de 2 niveles, los cuales son:

- Un dispositivo móvil (Android) que actuará como servidor. Aquí se creará la conexión ya sea por wi-fi o por bluetooth para que el cliente pueda comunicarse con éste y tendrá una interfaz para configurar o utilizar las funciones de moderador en la aplicación. Este servidor estará activo hasta que el usuario cierre la aplicación o decida cerrarlo manualmente por medio de la interfaz.
- Un dispositivo móvil (Android) que actuará como cliente, en el cual se le presentará al usuario (expositor) las pantallas y funcionalidades que están definidas para él,

como por ejemplo ver en tiempo real el cronómetro o recibir mensajes de alerta del moderador. El cliente de estar comunicado en todo momento con el servidor creado en el otro dispositivo

4.3.2 Lógica

La arquitectura lógica se puede dividir en dos partes, por un lado está la parte del cliente y por la otra, la parte del servidor. Tener en cuenta que en este caso no existe una capa de datos ya que la aplicación no requiere del almacenamiento de datos.

4.3.2.1 Cliente

En la arquitectura lógica del cliente tenemos la capa de presentación y negocio. En la capa de presentación se encuentra la interfaz gráfica poder visualizar en tiempo real las operaciones que está realizando el servidor. En la capa de negocio que es donde se hace gestión de la lógica de la aplicación, tenemos la configuración del tiempo, envío de avisos predefinidos del moderador, control del cronómetro y alertas.

4.3.2.1 Servidor

En arquitectura lógica del servidor se presentan las mismas capas del cliente, que son la de presentación y negocio. En la capa de presentación está la interfaz gráfica para configurar correctamente el servidor, por ejemplo establecer la conexión TCP/IP. Por otra parte, en la capa de negocio se encuentra el visualizar en tiempo real el tiempo y recibir alertas del moderador.

4.4 Herramientas de Desarrollo

En cualquier proyecto de sistema software, es necesario contar con herramientas ya desarrolladas. A continuación, se detallan las principales herramientas utilizadas para el proyecto:

- Android Studio, es un entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para android, ya que dado con un estudio previo, se comprobó que cumple con todo lo necesario para la solución del problema presentado.
- GitHub, se utiliza para el control de versiones, ya que cuenta con una versión gratuita, además de ser conocido y utilizado anteriormente por el equipo de trabajo, lo que ahorra recursos de investigación para tecnologías similares.
- Draw.io, software gratuito online para la elaboración de diagramas de flujo, diagramas de proceso, organigramas, UML, etc.
- Adobe Experience Design, utilizado para el modelado de interfaces y prototipado de la solución del problema, herramienta utilizada anteriormente por el grupo de trabajo en otros proyectos.
- Trello, herramienta utilizada para la gestión de proyectos, que hace que la colaboración sea sencilla.
- Visual Studio Code, herramienta para la edición de código fuente desarrollado por Microsoft de tipo Open Source.

- Computadores personales, estos tienen la tecnología necesaria para el desarrollo del proyecto, además fue opción del equipo de trabajo utilizar los equipos propio de cada uno.
- Dispositivos Android, una tablet y un smartphone con sistema operativo Android para realizar pruebas reales.

5 Gestión del Proyecto

Se presenta la información desde la gestión para la realización del proyecto.

5.1 Recursos

Para llevar a cabo un proceso se necesitan 3 recursos fundamentales, los cuales son los recursos humanos, físicos y tecnológicos. Se detallaran cada uno a continuación.

5.1.1 RRHH

El equipo de trabajo a cargo del proyecto, está conformado por dos estudiantes de último año de la carrera de Ingeniería de Ejecución en Informática de la Pontificia Universidad Católica de Valparaíso en el marco de la asignatura Proyecto de Título con el fin de optar al título profesional de Ingeniero de Ejecución en Informática, quienes cuentan con diversos conocimientos técnicos y prácticos en las distintas disciplinas que componen el diseño e implementación de un sistema informático.

5.1.2 Recursos Físicos

Se requieren computadores para el desarrollo del proyecto, se trabaja con un ASUS ROG con procesador Intel Core I5 de 7ma generación y 8GB de RAM, un DESKTOP con procesador FX 8350 y 16GB de RAM, un smartphone y/o tablet con sistema operativo android proporcionados por el cliente.

5.1.3 Recursos Tecnológicos

Los recursos tecnológicos requeridos para este proyecto son herramientas tales como:

- GitHub: Control de versiones.
- Android Studio: Herramienta oficial para el desarrollo de aplicaciones móviles android.
- Draw.io: Herramienta de lenguaje de modelado de sistemas
- Trello: Herramienta utilizada para la gestión de proyectos,
- Adobe Experience Design: Herramienta para el modelado de mockup y prototipos
- Además, es fundamental contar con acceso a internet para llevar a cabo el proyecto.

5.2 Gestión del Riesgo

En todo proyecto en desarrollo existen varios elementos que pueden traer problemas en la elaboración de este, es por esto que en esta sección veremos los distintos riesgos observados para el sistema, la relevancia de cada uno y las formas de mitigación propuestas.

Características de los riesgos de un software:

- Incertidumbre: Puede o no ocurrir, hay riesgo del 100% de probabilidad
- Pérdida: Si el riesgo ocurre, hay pérdida

5.2.1 Matriz de Probabilidad e Impacto

Para el correcto análisis se propone una escala para medir la probabilidad del riesgo y el impacto que traerá. Estas escalas se muestran en las siguientes tablas.

Tabla 1 Métricas para evaluar probabilidad de riesgo e impacto

Probabilidad	Valor Numérico	Impacto	Valor Numérico
Muy Probable	0.1	Muy Bajo	0.05
Relativamente Probable	0.3	Bajo	0.10
Probable	0.5	Moderado	0.20
Muy Improbable	0.7	Alto	0.40
Casi certeza	0.9	Muy Alto	0.80

Para el análisis total del riesgo y su tipo se tomará como base la siguiente escala.

Tabla 2 Métrica base de evaluación de riesgo

Tipo de Riesgo	Probabilidad x Impacto
Muy Alto	>0.50
Alto	<0.50
Moderado	<0.30
Bajo	<0.10
Muy Bajo	<0.05

5.2.2 Riesgos del Proyecto

A continuación se muestra el análisis detallado de riesgo de software, según el tipo de riesgo que existe. Además encontrará el plan de mitigación que mostrará las acciones a realizar en caso de que ocurra el riesgo, este plan busca reducir el impacto de dichos eventos si llegaran a suceder durante del desarrollo.

5.2.2.1 Riesgo del Proyecto

Este tipo de riesgo amenaza el plan del proyecto, la planificación temporal y sus costos.

Tabla 3 Riesgos del proyecto y plan de mitigación

Riesgo	Prob.	Imp.	TR	Mitigación
Falta de conocimiento en herramientas tecnológicas: Android Studio, ya que el desarrollo está basado en él y errores en el desarrollo tendrían repercusión en la entrega del producto final	0.9	0.20	Moderado	Estudiar las herramientas elegidas para el desarrollo y buscar referentes avanzados en estos temas en internet para resolver las dudas que se presenten.
La interfaz sea compleja, que no sea de fácil entendimiento para el usuario final(Sin usabilidad)	0.5	0.80	Alto	Realizar un estudio de usabilidad con expertos en el área para que evalúen el software

5.2.2.2 Riesgo Tecnológico

Este tipo de riesgo amenaza la calidad y la planificación temporal del software. La implementación podría volverse difícil o imposible.

Tabla 4 Riesgos tecnológicos y plan de mitigación

Riesgo	Prob.	Imp.	TR	Mitigación
Diferencias entre el modelo diseñado y lo solicitado por el cliente	0.7	0.8	Muy Alto	Establecer un plan de trabajo con reuniones periódicas con cliente para que apruebe los avances y se establezcan las modificaciones y mejoras para la siguiente reunión

5.2.2.3 Riesgo del Negocio

Este tipo de riesgo amenaza la viabilidad del software a desarrollar.

Tabla 5 Riesgos del negocio y plan de mitigación

Riesgo	Prob.	Imp.	TR	Mitigación
Realizar desarrollo que incumpla alguna legislación	0.1	0.80	Bajo	Contratar asesoría legal en relación a los proyectos a realizar. Mantener un estándar ético y una imagen limpia siempre.

5.3 Plan de trabajo

El plan de trabajo tentativo del presente proyecto se dividirá en 4 fases denominadas: *Marco General del proyecto*, *Análisis y Diseño*, *Desarrollo y Codificación*, *Pruebas y resultados*. Se debe tener en cuenta de que la metodología de trabajo que se utilizará durante el desarrollo de la aplicación corresponderá al *modelo de desarrollo evolutivo*, siendo más específicos, el modelo basado en prototipos.

A continuación, se presentará de forma más detallada cada una de las fases mencionadas anteriormente.

5.3.1 Marco General del proyecto

Período estimado: (19 Días)

- Definición de la problemática
- Objetivo general y específicos
- Identificación de los problemas detectados
- Diseño preliminar tentativo

5.3.2 Análisis y Diseño del proyecto

Período estimado: (30 Días)

- Captura de requerimientos funcionales y no funcionales
- Confección y validación de prototipos
- Creación de documento de especificación
- Arquitectura física y lógica de la solución
- Herramientas tecnológicas a utilizar

5.3.3 Desarrollo y Codificación de la aplicación

Período estimado: (66 Días)

- Desarrollo de versión funcional de la aplicación
- Pruebas internas de la aplicación
- Confección de actividades
- Corrección de errores y crear validaciones

5.3.4 Pruebas y Resultados

Período estimado: (15 Días)

- Diseño de pruebas
- Pruebas de la aplicación en terreno
- Análisis de los datos y obtención de resultados
- Conclusiones sobre la aplicación propuesta
- Definición de trabajo a futuro

6 Desarrollo de proyecto

A continuación, se presentan las distintas etapas de identificación y organización de todas las actividades y procesos relevantes que intervienen en la búsqueda de una meta u objetivo.

6.1 Identificación de Requerimientos

Comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado.

Los requerimientos obtenidos se separan en requerimientos funcionales y no funcionales.

6.1.1 Requerimientos funcionales

En esta sección se presentaran las principales funcionalidades del software.

Tabla 6 Requerimiento funcional alerta de tiempo expositor

ID	RF-01
Título	Alerta de tiempo para expositor
Descripción	Un usuario expositor, debe poder escuchar un sonido y/o ver mediante una alerta que le queda poco tiempo para exponer, según la configuración de tiempo previa que le hayan asignado

Tabla 7 Requerimiento funcional color en pantalla para el expositor

ID	RF-02
Título	Color en pantalla para el expositor
Descripción	Un usuario expositor, debe poder diferenciar con 3 tipos de colores diferentes(verde, amarillo y rojo) el tiempo transcurrido indicando que este se ha disminuido, siendo previamente configurado según un tiempo estipulado en qué minutos/segundos aparecerán en pantalla

Tabla 8 Requerimiento funcional término de exposición

ID	RF-03
Título	Término de exposición

Descripción	Un usuario expositor, deber poder ver en pantalla con un signo de “Stop” cuando el contador de tiempo para exponer haya llegado a cero, o lo haya detenido manualmente el usuario moderador

Tabla 9 Requerimiento funcional controlar el tiempo

ID	RF-08
Título	Controlar tiempo
Descripción	Un usuario moderador, debe poder iniciar, parar y reiniciar el cronómetro.

Tabla 10 Requerimiento funcional definir tiempo para el expositor

ID	RF-04
Título	Definir tiempo para el expositor
Descripción	Un usuario moderador, debe poder ingresar el tiempo que tendrá el usuario expositor.

Tabla 11 Requerimiento funcional definir alertas para el expositor

ID	RF-05
Título	Definir alertas para el expositor
Descripción	Un usuario moderador, podrá definir el tiempo en que la pantalla cambiará de color (verde, amarillo y rojo), asignándoles menores tiempo según el tiempo total del expositor y los colores según el orden mencionado.

Tabla 12 Requerimiento funcional mensajes para el expositor

ID	RF-06
Título	Mensajes para el expositor
Descripción	El usuario moderador, debe poder enviar mensajes al usuario expositor, los cuales son predefinidos, estos deben informar al expositor

Tabla 13 Requerimiento funcional intermedio al expositor (pausa)

ID	RF-07

Título	Intermedio al expositor (Pausa)
Descripción	Un usuario moderador, debe poder pausar el cronómetro, dándole un aviso previo de al menos 20 segundos al usuario expositor, si se presenta alguna eventualidad que lo amerite.

6.1.2 Requerimientos no funcionales

En esta sección se presentarán las características generales y restricciones de la aplicación.

Tabla 14 Requerimientos no funcionales

ID	**Descripción**
RNF-01	El dispositivo móvil tiene que operar con Android Lollipop (versión 5) o superior
RNF-02	El sistema debe tener un tiempo de respuesta que no sea mayor a los 3 segundos.
RNF-03	El sistema tiene que operar con una conexión wifi y/o bluetooth
RNF-04	La interfaz de la aplicación debe ser sencilla e intuitiva.
RNF-05	La conexión se tiene que restablecer automáticamente si esta se pierde.

6.2 Análisis de requerimientos

Los requerimientos del software deben mostrar todo lo que el sistema debe hacer más todas las restricciones sobre la funcionalidad. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software. Se presenta a continuación el diagrama de casos de uso ligado al sistema desarrollado.

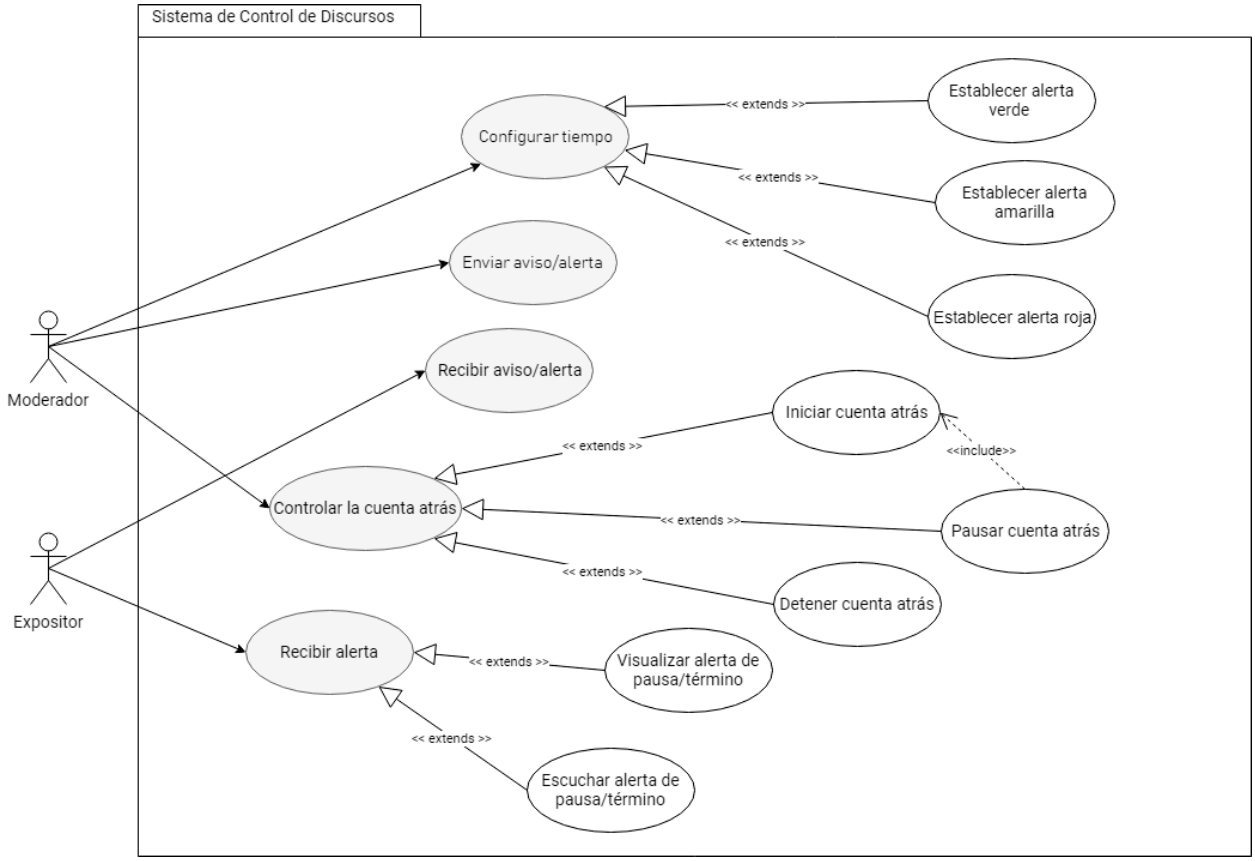

Figura 2 Diagrama de Casos de Uso del software

6.3 Diseño

En este caso, el software que queremos implementar no será necesaria la construcción del diseño de base de datos porque no se requerirá el almacenamiento de datos. Por otra parte si necesita una identificación de los mensajes enviados desde el servidor al cliente por lo que es necesario especificar, además se añadirán los mockups del diseño del software.

6.3.1 Comandos Cliente-Servidor

La arquitectura cliente servidor necesita de la especificación de comandos, los que son utilizados para transportar el mensaje los cuales están en forma de bytes al momento del envío. Los comandos para los mensajes son los siguientes:

Tabla 15 Comandos de la aplicación cliente-servidor

Comando	Explicación
ALERTAS_”TIEMPO INICIAL”_”ALERTA 1”_”ALERTA 2”_”ALERTA 3”_”SONIDO ALERTAS ON/OFF”	La aplicación iniciará el contador cuenta atrás con los valores ingresados en el mensaje. Tenemos lo siguiente: • “TIEMPO INICIAL”: es un número de tipo entero (interpretado en minutos) utilizado para establecer el tiempo inicial del contador. • “ALERTA 1”: es un número de tipo entero (interpretado en minutos) utilizado para establecer la alerta primaria del contador. • ”ALERTA 2”: es un número de tipo entero (interpretado en minutos) utilizado para establecer la alerta secundaria del contador. • ”ALERTA 3”: es un número de tipo entero (interpretado en minutos) utilizado para establecer la alerta final del contador. • ”SONIDO ALERTAS ON/OFF”: es un booleano (TRUE o FALSE) para establecer si se activará o no los sonidos de las alertas. Un ejemplo de este comando puede ser: “ALERTAS_4_3_2_1_TRUE”
INICIAR/PAUSAR	La aplicación pausara o reanudará el conteo en retroceso del tiempo. La acción dependerá si el contador está en pausa o en ejecución
DETENER	La aplicación detendrá el conteo en retroceso del tiempo y cerrará la actividad del contador.
“CUALQUIER PALABRA O FRASE”	La aplicación al escribir cualquier frase o palabra que no sea igual a las mencionadas anteriormente, se interpretará como “un mensaje” que solo podrá ser enviado desde HOST y podrá ser recibido/leído por el CLIENTE.

6.3.2 Interfaz

En esta sección para esta entrega, se mostrará la maqueta del diseño del software, mostrando las pantallas principales las que se dividirán en dos, interfaz del moderador y del expositor.

6.3.2.1 Mockups de Interfaz del Moderador

Por parte de la interfaz del moderador tenemos las funciones de control de la aplicación, en el cual se presenta lo siguiente:

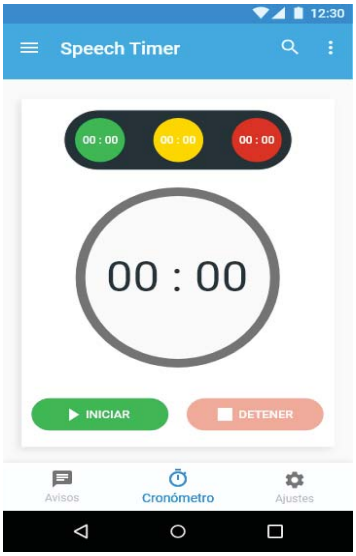

Figura 3 Pantalla de Inicio – Home (Moderador)

En la figura 3 se puede apreciar la pantalla de inicio que aparece cuando el moderador crea una sala. En esta pantalla puede definir las 2 alertas y luego de haberlas definido, puede iniciar el cronómetro.

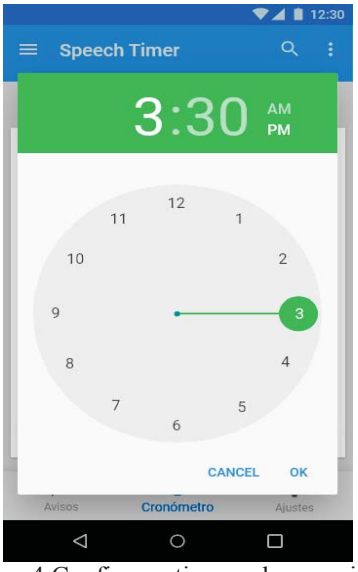

Figura 4 Configurar tiempo de exposición

En la figura 4 se puede ver la pantalla en el cual se puede controlar el tiempo estimado que durará la conferencia. Esta misma funciona tanto para definir la alerta principal y la secundaria.

Figura 5 Pantalla de mensajes predefinidos al expositor

En la figura 5, se representa la pantalla de mensajes predefinidos que puede enviarle el moderador al expositor. Estas pueden surgir por ejemplo cuando el moderador quiere pausar o detener el cronómetro por una situación inesperada, entonces puede avisarle al expositor por este medio lo sucedido.

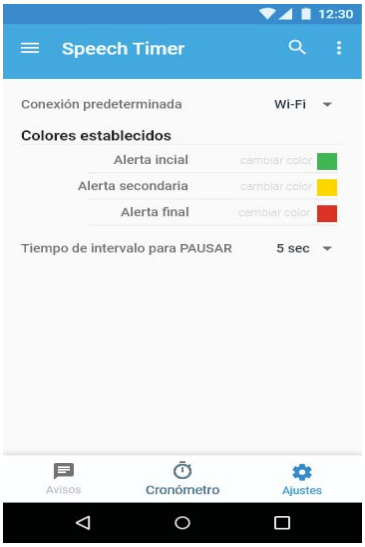

Figura 6 Pantalla de ajustes

En la figura 6, se muestra la pantalla de ajustes de la aplicación, donde el usuario puede cambiar la configuración predeterminada entre wi-fi o bluetooth. También puede cambiar los colores de las alertas y establecer un margen de tiempo para mostrarle un mensaje al expositor de que el cronómetro se pausará.

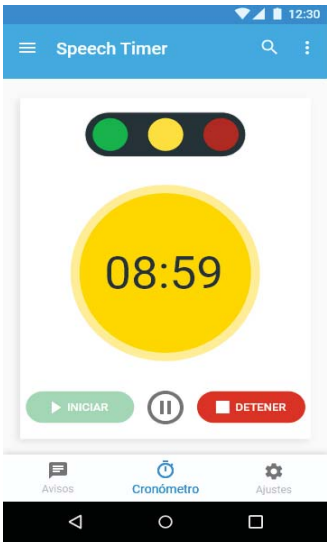

Figura 7 Pantalla del cronómetro en funcionamiento (Moderador)

En la figura 7, podemos ver la pantalla del cronómetro en funcionamiento donde se destaca el cambio de color a medida que pasa el tiempo y en este caso por ejemplo, se haya activado la alerta secundaria que está representada en color amarillo. Por otra parte, se pueden observar los botones para pausar o detener la cuenta atrás.

6.3.2.2 Mockups de Interfaz del Expositor

Por consiguiente tenemos la interfaz del expositor, es muy similar a la del moderador pero de forma limitada de tal forma que solo puede reflejar las acciones hechas por el moderador.

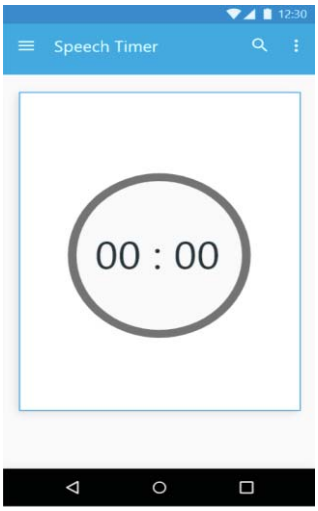

Figura 8 Pantalla de inicio-home (Expositor)

En la figura 8, se puede apreciar la pantalla de inicio que aparece cuando un expositor se une a la sala creada por un moderador. En esta pantalla el expositor queda a la espera de las modificaciones y/o acciones del moderador.

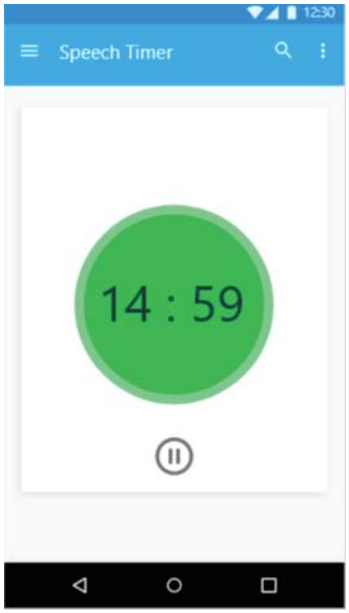

Figura 9 Pantalla del cronómetro en funcionamiento (Expositor)

En la figura 9, podemos ver la pantalla del cronómetro en funcionamiento donde se destaca el primer color que cambiará a medida que pasa el tiempo.

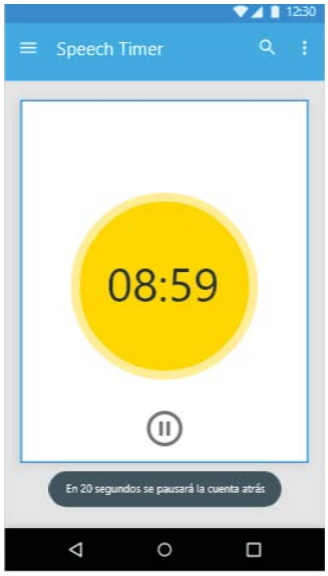

Figura 10 Pantalla del cronómetro con avisos

En la figura 10, podemos ver la pantalla del cronómetro con otro color a medida que el tiempo se redujo, por otro lado podemos ver un mensaje que indica que el tiempo se entra dentro de 20 segundos, acción que fue enviada por el moderador.

6.3.2.3 Interfaz del prototipo funcional del moderador y expositor

Después de todos los prototipos que se hicieron durante la etapa de codificación y testing se llegó al primer prototipo funcional del software, con un interfaz algo cambiada a lo que se tenía pensado anteriormente. Estos cambios fueron producto de las pruebas que se realizaron con usuarios finales o stakeholders, para que el usuario pudiera interactuar con la aplicación de una manera más amigable y no tan engorrosa como fue en un principio.

El diseño de la interfaz tanto del moderador como el expositor se presentará a continuación:

Figura 11 Pantalla de Inicio (prototipo funcional)

En la figura 11 se puede observar la pantalla inicial del prototipo funcional, el cual será el mismo para el moderador y expositor.

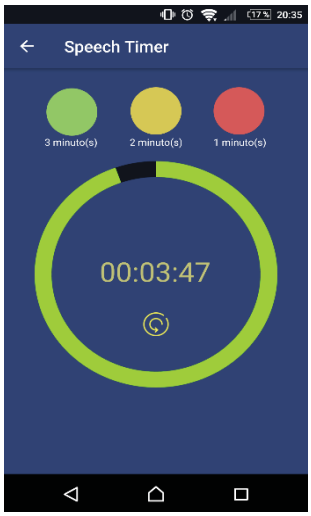

Figura 12 Pantalla del cronometro (prototipo funcional)

En la figura 12 se puede apreciar la pantalla del cronómetro cuenta atrás, que podrá verla el expositor cuando el moderador haya establecido los valores del tiempo inicial, alertas y presione INICIAR.

7 Resultados obtenidos

Recopilando todos los datos, recomendaciones y consejos de los stakeholders o usuarios finales que se pudieron adquirir en la etapa de desarrollo y testeo, se ha llegado a un prototipo funcional estable de la aplicación. Comenzando con la etapa inicial del desarrollo del software (primer prototipo desechable), se buscó validar que la pantalla del contador fuese en lo posible lo más fiel al diseño presentado al cliente, ya que la interfaz presentada era lo más cercano a lo que necesitaba el cliente en un principio. Esta interfaz básicamente constaba de una barra de progreso circular que va cambiando de color a medida que se activan las alertas preestablecidas, en el centro se encuentra el indicador de la cuenta atrás. También se presentaba en forma de "semáforo" las alertas para indicarle al usuario si ya se activó o no. En este prototipo tomamos el consejo de uno de los usuarios, el que señaló lo siguiente: "Podrían haber alertas preestablecidas en la aplicación, así el usuario podría elegir alguna de ellas con solo un toque".

Luego de haber pasado por el primer prototipo desechable se continuó con el siguiente que se enfocó esencialmente en la conexión entre sí de dos dispositivos (smartphones o tablets) por medio de "Wifi Direct" (protocolo indispensable para el desarrollo del software). Esta etapa fue la más complicada y que más cambios tuvo, debido a que surgieron varios problemas en la codificación para establecer la conexión entre los dispositivos. El cambio más radical que se produjo en esta fase, fue el diseño de la interfaz dedicada a la activación del wifi a través de la aplicación, búsqueda de pares y aceptar la conexión para establecer a un CLIENTE y un HOST.

Con las pruebas que realizaron los usuarios, se pudieron encontrar varios detalles que se han ido solucionando hasta llegar al prototipo funcional. Por ejemplo en la pérdida de conexión entre los dispositivos, si se quería restablecer la conexión el usuario no podía ya que no se lograba cerrar correctamente la conexión en caso de que ocurriera esa situación. Otro problema detectado a través de las pruebas fue debido a la pregunta de uno usuario que nos preguntó lo siguiente: "Al realizar la conexión entre los dispositivos, ¿se puede elegir manualmente CLIENTE y HOST? Al depurar la aplicación, el equipo de trabajo se percató que la conexión se establecía al azar, lo que generó un problema puesto que eso no debía ocurrir de aquella manera, por lo que se tuvo que investigar y leer la documentación de Android con el fin de encontrar una manera de poder hacer la conexión manual. A fin de cuentas, se encontró la solución a este detalle y ahora el usuario puede elegir qué dispositivo será el CLIENTE o el HOST.

Finalmente, dentro de todo el periodo de pruebas realizado para validar el funcionamiento de la conexión, se llegó finalmente a un prototipo funcional y estable, donde solo falta realizar algunas validaciones a algunos elementos, por otro lado se logró el funcionamiento de este. Se debe agregar que todas las pruebas se realizaron en un ambiente controlado, con 8 participantes, citados en distintas fechas según el prototipo a evaluar, en el que se les entregó un pequeño manual de cómo funciona el software y que debían hacer para luego tener una retroalimentación que va desde su percepción hasta que no les gusta y/o que cambiarían y porque.

8 Conclusiones

En el presente proyecto, tras una ardua y exhaustiva investigación del equipo de trabajo sobre el tema de las conferencias, exposiciones, discursos, etc., se determinó que existen diversas herramientas que sirven de apoyo para ejercer correctamente estos eventos. Esta investigación sirvió para estudiar el tema y entender de una mejor manera el área en el que se enfocará el software.

Por otra parte se seleccionaron las herramientas, ambiente, y metodología para iniciar a desarrollar la aplicación. Entre esta selección se encuentra Android Studio, cuya herramienta será la principal para codificar, testear e implementar en los dispositivos móviles. Respecto a la metodología, se utilizará el modelo de proceso evolutivo basado en prototipos, ya que es uno de los métodos que mejor se adapta a las necesidades del proyecto, considerando que se puede ir validado el uso de cada versión con el cliente y los usuarios finales a través de prototipos desechables.

En la actualidad, luego de diversos prototipos desechables, el software se encuentra en una versión funcional y estable que cumple con las expectativas del cliente, pero aun así sigue siendo un prototipo, por lo que no es una versión final puesto que carece de algunas características como validaciones, entre otras, que lo convertirían en un software perfectamente funcional.

9 Referencias bibliográficas

- [1] [Karam, 05] Tanius Karam, Estudio del discurso y al análisis del discurso. Disponible via web en http://www.redalyc.org/pdf/687/68720305.pdf. Revisada por última vez el 15 de Agosto.
- [2] [Santamaria, 12] Pedro Santamaría. Timer, Una sencilla pero funcional aplicación para gestionar intervalos de tiempo. Disponible via web en https://www.applesfera.com/aplicaciones-ios-1/timer-una-sencilla-pero-funcional-aplicacion-para-gestionar-intervalos-de-tiempo. Revisada por última vez el 14 de Agosto de 2018.
- [3] [Dobbins, 14] Michael Dobbins, Speech Timer. Disponible vía web en http://www.hellogovernance.com/android-apps/speech-timer. Revisada por última vez el 14 de Agosto de 2018.
- [4] [Moltedo ,05] y [Muga, 05] Claudio Moltedo Castaño y Alfonso Muga Naredo, Reglamento general del Claustro Pleno. Disponible vía web en http://www.ucv.cl/pucv/site/artic/20150410/asocfile/20150410094342/2__reglamento_general_del_claustro_pleno.pdf. Revisada por última vez el 17 de Agosto de 2018.
- [5] Stackeholders, Wikipedia, https://es.wikipedia.org/wiki/Parte_interesada_(empresas). Revisado por última vez el 25 de Septiembre de 2018
- [6] Modelo prototipo, Ecured, https://www.ecured.cu/Modelo_de_prototipos. Revisado por última vez el 20 de Agosto de 2018

Anexos

A. Se añadirán las demás capturas de los mockups en esta sección.

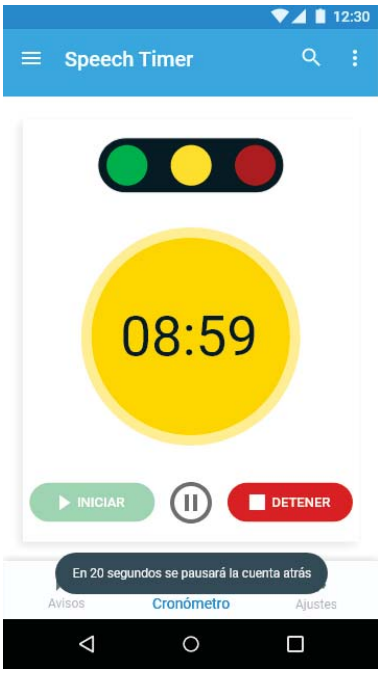

Figura 13 Pantalla mensaje de aviso al pausar

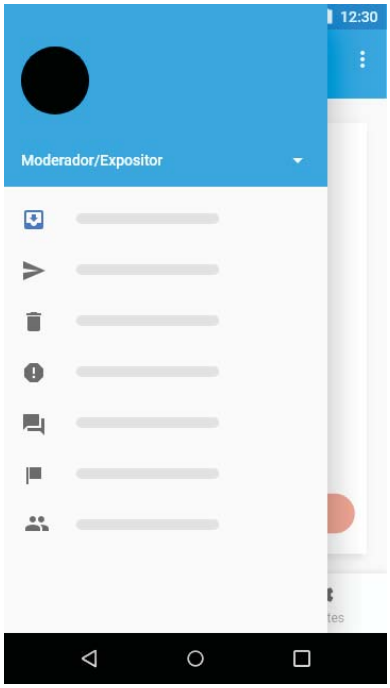

Figura 14 Pantalla del menú general

Figura 15 Pantalla mensaje de aviso al finalizar tiempo

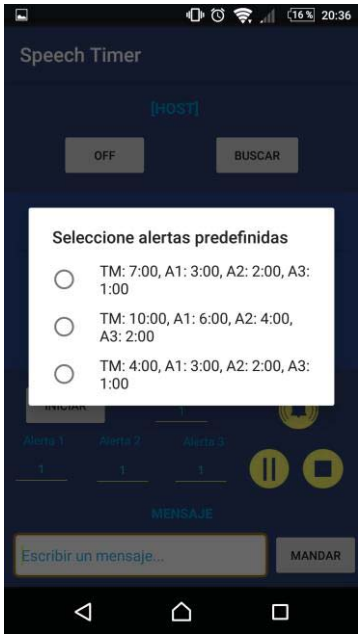

Figura 16 Cuadro de dialogo con alertas predefinidas