

## UNIVERSIDAD DE TARAPACÁ



## FACULTAD DE INGENIERÍA



Departamento de Ingeniería en Computación e Informática

### **Proyecto:** **Desarrollo de aplicación de reconocimiento y reproducción de texto en víveres para gente no vidente**

**Autor(es):** Christian Cáceres

**Ernesto García**

**Daniel Ramírez**

**Asignatura:** Proyecto 2

**Profesor(es):** Diego Alberto Aracena Pizarro

ARICA, 08 de Enero 2021

## Historial de Cambios

Dueño del documento: Grupo "Ingenieros Anónimos"

Fecha	Versión	Descripción	Autor(es)
06/10/2020	1.0	Versión preliminar del formato	Christian Cáceres Ernesto García Daniel Ramírez
20/10/2020	1.1	Modificación del informe plan de proyecto	Christian Cáceres Ernesto García Daniel Ramírez
27/10/2020	1.2	Revisión del informe plan de proyecto	Christian Cáceres Ernesto García Daniel Ramírez
29/10/2020	1.3	Modificación del informe plan de proyecto	Christian Cáceres Ernesto García Daniel Ramírez
02/11/2020	1.4	Versión final del informe plan de proyecto	Christian Cáceres Ernesto García Daniel Ramírez
18/12/2020	1.5	Modificación del informe de avance	Christian Cáceres Ernesto García Daniel Ramírez
19/12/2020	1.6	Modificación del informe de avance	Christian Cáceres Ernesto García Daniel Ramírez
22/12/2020	1.7	Versión final del informe de avance	Christian Cáceres Ernesto García
03/01/2021	1.8	Modificación del informe final	Ernesto García Daniel Ramírez
06/01/2021	1.9	Modificación del informe final	Ernesto García Daniel Ramírez

## Tabla de Contenidos

Historial de Cambios .....	2
1. Panorama General.....	5
1.1. Resumen del proyecto: .....	5
1.1.1. Introducción: .....	5
1.1.2. Escenario:.....	5
1.1.3. Propósito: .....	5
1.1.4. Alcance:.....	6
1.1.5. Objetivo General: .....	6
1.1.6. Objetivos Específicos:.....	6
1.1.7. Suposiciones: .....	6
1.1.8. Restricciones: .....	6
1.1.9. Entregables del Proyecto: .....	6
2. Organización del Proyecto .....	7
2.1. Personal y entidades internas .....	7
2.2. Roles y responsabilidades .....	7
2.3. Mecanismos de comunicación .....	7
3. Planificación de los procesos de gestión.....	8
3.1. Planificación inicial del proyecto.....	8
3.1.1. Planificación de estimaciones .....	8
3.1.2. Planificación de Recursos Humanos .....	8
3.2. Lista de actividades .....	8
3.2.1. Actividades de trabajo.....	8
3.2.2. Carta Gantt .....	10
3.2.3. Asignación de tiempo.....	10
3.3. Planificación de la gestión de riesgos .....	11
4. Planificación de procesos técnicos .....	12
4.1. Modelo de Proceso .....	12
4.1.1. Diagrama de Caso de Uso General .....	12
4.1.2. Diagrama de Caso de Uso De Sistema.....	12
4.1.3. Diagrama de Secuencia .....	14
4.1.3.1. Inicia Aplicación .....	14
4.1.3.2. Capturar Video .....	14
4.1.3.3. Reconocer Texto .....	15
4.1.3.4. Transformar texto en audio .....	15

4.1.3.5. Reproducir Audio.....	15
4.1.4. Modelo de Clases .....	16
4.1.5. Descripción de la Arquitectura .....	17
4.1.6. Diseño de la Interfaz de Usuario .....	18
4.1.7. Especificación de requerimientos.....	19
4.2. Herramientas y técnicas.....	19
5. Planificación de procesos de soporte.....	20
5.1. Planificación de la documentación .....	20
6. Plan de Integración .....	20
7. Modelo de Implementación .....	20
8. Módulos Implementados .....	21
9. Pruebas .....	22
10. Conclusiones.....	23
11. Referencias .....	24

## 1. Panorama General

### 1.1. Resumen del proyecto:

#### 1.1.1. Introducción:

En la actualidad las personas invidentes o con problemas de vista pueden recurrir a la tecnología para facilitar su día a día. Es por eso que el presente informe tiene como fin mostrar la planificación del proyecto "Cámara lectora para gente invidente", el cual tiene pensado hacer una aplicación móvil para ayudar a las personas con problemas de visión a leer etiquetas en los productos que consumen.

En este informe se mostrará el escenario considerado para el proyecto, su propósito, alcance y objetivos que fueron considerados llevar a cabo el proyecto. Además, se observa la distribución de roles entre el personal del proyecto, junto con la planificación de los procesos de gestión del mismo.

#### 1.1.2. Escenario:



Figura 1: Escenario del proyecto

En la Figura 1 se puede ver el Escenario propuesto para el proyecto, explicado a continuación.

**Problema:** "Las personas con problemas de vista, tienen dificultades para leer, ya sea los víveres que tiene en la cocina o el texto de una revista"

**Solución:** "Con esta aplicación, la persona puede apuntar con la cámara del celular al objeto o texto, el celular reconoce el texto y luego lo reproduce"

#### 1.1.3. Propósito:

En la actualidad la tecnología de asistencia a videntes está aumentando gracias a todos los avances hechos en las últimas décadas, una persona invidente puede tener una vida y trabajo normales gracias a las nuevas tecnologías. Es por esto que en el presente proyecto se ha determinado desarrollar un software que pueda reconocer el texto frente a la cámara frontal (de un smartphone) y reproducirlo en voz alta,

ayudando a los invidentes (o personas con problemas a la vista) a estar más conectados con el mundo.

#### 1.1.4. Alcance:

La aplicación utilizará la cámara frontal del móvil para localizar texto y luego reproducirlo en forma de audio. Será desarrollado en el lenguaje de programación "Python", versión 3.6.8.

#### 1.1.5. Objetivo General:

Desarrollar una aplicación que reconozca texto y lo reproduzca en voz alta.

#### 1.1.6. Objetivos Específicos:

1. Estudiar y definir el problema.
2. Proponer la solución al problema y sus alternativas.
3. Desarrollar la aplicación de acuerdo a la solución seleccionada.
4. Realizar pruebas a la aplicación.
5. Entregar la aplicación finalizada.

#### 1.1.7. Suposiciones:

1. Se asume que las personas invidentes que utilicen la aplicación tienen un celular con una cámara frontal funcional.
2. Se asume que las personas no tienen problemas auditivos.
3. Se presume que el medio por el cual se distribuirá la aplicación será por Google Play.

#### 1.1.8. Restricciones:

1. El proyecto debe ser realizado en el plazo determinado por el usuario.
2. La programación del software será realizada en el lenguaje de programación Python.
3. La aplicación debe poder ser usada por una persona no vidente.

#### 1.1.9. Entregables del Proyecto:

1. Bitácoras semanales.
2. Informe de Plan de Proyecto
3. Presentación del Escenario.
4. Informe de avance.
5. Informe final.
6. Presentación final.
7. Manual de usuario.
8. Wiki del proyecto.
9. Producto final.

## 2. Organización del Proyecto

### 2.1. Personal y entidades internas

Jefe de proyecto, Programadores, Diseñadores, Secretario.

### 2.2. Roles y responsabilidades

Jefe de proyecto: Se encarga de la coordinación del trabajo del grupo y de la comunicación con el usuario. El responsable es: Daniel Ramírez Díaz.

- Responsable de programar y desarrollar el software en Python que implemente las funciones de leer y reproducir texto para la aplicación.
- Responsable de diseñar la interfaz visual de usuario para la aplicación móvil.
- Responsable de la documentación del proyecto, de esta forma, realizan las bitácoras del proyecto y formulan los informes requeridos.

### 2.3. Mecanismos de comunicación

El grupo se comunica principalmente por la aplicación Discord, la cual se trata de un chat de voz y texto gratuito con una interfaz muy moderna e intuitiva, en la cual se pueden notificar con mensajes de texto o canales de voz. Alternativamente el grupo se comunicará por WhatsApp. Para realizar los informes y bitácoras se utilizará Microsoft Word a través de OneDrive, que permite a los miembros del grupo trabajar simultáneamente en los documentos.

### 3. Planificación de los procesos de gestión

#### 3.1. Planificación inicial del proyecto

##### 3.1.1. Planificación de estimaciones

Recurso	Valor	Cantidad
Notebook	\$500.000	3
Smartphone	\$180.000	3
Software de desarrollo Python	De libre acceso	3
Microsoft Office	\$56.000	3
Sueldo del desarrollador	\$938.000	3
Costo Total del proyecto	\$5.022.000	



##### 3.1.2. Planificación de Recursos Humanos

Responsable de programar: Ernesto García.

Responsable de diseñar: Daniel Ramírez.

Responsable de documentar: Christian Cáceres.

#### 3.2. Lista de actividades

##### 3.2.1. Actividades de trabajo

- Crear diseño básico de la aplicación.  
**Descripción:** Se crea el diseño base de la aplicación.  
**Responsable:** Daniel Ramírez.
- Realizar el informe de plan de proyecto.  
**Descripción:** Se realiza el informe de plan de proyecto  
**Responsable:** Christian Cáceres.
- Investigar sobre el software OCR.  
**Descripción:** Se investigan sobre los tipos de software OCR y lo que pueden hacer.  
**Responsable:** Ernesto García.
- Investigar la programación en Python.



**Descripción:** Se investiga la programación en Python y se hacen cursos. ✓✓  
**Responsable:** Daniel Ramírez.

- Diseñar la interfaz de usuario de la aplicación.  
**Descripción:** Se diseña una interfaz más avanzada para la aplicación.  
**Responsable:** Daniel Ramírez.
- Implementar el reconocimiento de texto.  
**Descripción:** Se implementa el reconocimiento de texto en Python.  
**Responsable:** Daniel Ramírez.
- Realizar pruebas de reconocimiento de texto.  
**Descripción:** Se realizan pruebas de reconocimiento de texto.  
**Responsable:** Ernesto García.
- Realizar el informe de avance de proyecto.  
**Descripción:** Se realiza el informe de avance de proyecto.  
**Responsable:** Christian Cáceres
- Implementar la lectura de texto.  
**Descripción:** Se implementa la lectura de texto en Python.  
**Responsable:** Ernesto García.
- Realizar pruebas de lectura de texto.  
**Descripción:** Se realizan pruebas de reconocimiento de texto.  
**Responsable:** Ernesto García.
- Programar la interfaz de la aplicación.  
**Descripción:** Se programa la interfaz de usuario con el software Kivy.  
**Responsable:** Daniel Ramírez.
- Realizar pruebas de funcionalidad.  
**Descripción:** Se realizan pruebas de funcionalidad de la aplicación.  
**Responsable:** Ernesto García.
- Realizar documentación de los códigos.  
**Descripción:** Se realiza la documentación.  
**Responsable:** Christian Cáceres.
- Realizar el informe final de proyecto.  
**Descripción:** Se realiza el informe final del proyecto.  
**Responsable:** Christian Cáceres
- Realizar la presentación final del proyecto.  
**Descripción:** Se realiza la presentación final de proyecto. ✓  
**Responsable:** Daniel Ramírez

**Comentado [u1]:** Secretario Ejecutivo???  
Justificar

### 3.2.2. Carta Gantt

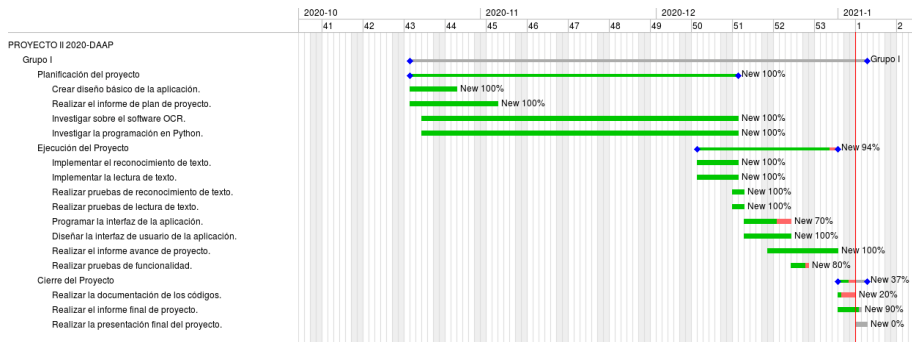


Figura 2: Carta Gantt del proyecto

En la Figura 2 se puede apreciar el estado actual del avance del proyecto según la Carta Gantt.

### 3.2.3. Asignación de tiempo

Planificación del proyecto: 4 semanas.

Ejecución del proyecto: 5-6 semanas.

Cierre de proyecto: 1-2 semanas.

### 3.3. Planificación de la gestión de riesgos

RIESGOS	PROBABILIDAD DE OCURRENCIA	NIVEL DE IMPACTO	ACCIÓN REMEDIAL
Un integrante del equipo tiene problemas de salud.	10%	1	Repartir trabajo entre los integrantes restantes
La cámara o altavoz del Smartphone se dañan	25%	1	Reponer Smartphone.
La pantalla táctil del Smartphone esté defectuosa	15%	2	Reemplazar pantalla táctil.
La batería del Smartphone deja de funcionar.	30%	3	Cargar el Smartphone
Los programas se pierden debido a un error en el dispositivo de almacenamiento.	10%	1	Respaldar los datos periódicamente

Nivel de impacto:

- 1: Catastrófico
- 2: Crítico
- 3: Marginal
- 4: Despreciable



## 4. Planificación de procesos técnicos

### 4.1. Modelo de Proceso

#### 4.1.1. Diagrama de Caso de Uso General

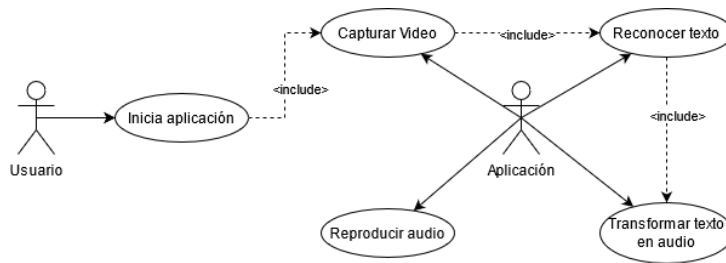


Figura 3: Caso de Uso del proyecto

En la Figura 3 se pueden ver el Caso de Uso del proyecto en el cual interactúan el usuario y la aplicación.

#### 4.1.2. Diagrama de Caso de Uso De Sistema

<b>Nombre:</b>	Iniciar aplicación
<b>Autor:</b>	Ernesto García
<b>Fecha:</b>	22/12/2020
<b>Descripción:</b>	Permite al usuario iniciar la aplicación.
<b>Actor:</b>	Usuario.
<b>Flujo Normal:</b>	
Usuario	Sistema
1.-El usuario presiona el icono de la aplicación	2.-El sistema inicia la aplicación

<b>Nombre:</b>	Capturar video
<b>Autor:</b>	Christian Cáceres
<b>Fecha:</b>	22/12/2020
<b>Descripción:</b>	Permite al usuario capturar un video que contenga texto que se desea leer.
<b>Actor:</b>	Usuario.
<b>Precondiciones:</b>	La aplicación debe tener acceso a la cámara
<b>Flujo Normal:</b>	La aplicación detecta texto
Usuario	Aplicación
4.-El usuario presiona la pantalla	1.-La aplicación captura video 2.-Incluye el C.U.S Reconocer texto 3.-La aplicación manda un mensaje de voz "Se ha encontrado texto" 5.-Incluye C.U.S Reproducir Audio

<b>Flujo Alternativo:</b> La aplicación no detecta texto	
Usuario	Sistema
	2.1.-La aplicación manda un mensaje de voz "No se encuentra texto"
<b>Postcondiciones:</b> Se genera una imagen	

<b>Nombre:</b>	Reconocer texto
<b>Autor:</b>	Ernesto García
<b>Fecha:</b>	22/12/2020
<b>Descripción:</b> Comprueba si se encuentra texto en la imagen	
<b>Actor:</b> Aplicación.	
<b>Precondiciones:</b> La aplicación debe tener acceso a los archivos del smartphone. Debe haber un haber una imagen capturada.	
<b>Flujo Normal:</b> La aplicación detecta texto	
Aplicación	Sistema
1.-La aplicación busca la imagen capturada en el C.U.S Capturar Video 2.-La aplicación es procesada	
<b>Postcondiciones:</b> Se genera un archivo de texto	

<b>Nombre:</b>	Transformar texto en audio
<b>Autor:</b>	Christian Cáceres
<b>Fecha:</b>	22/12/2020
<b>Descripción:</b> La aplicación transforma el texto encontrado a formato audio.	
<b>Actor:</b> Aplicación.	
<b>Precondiciones:</b> La aplicación debe tener acceso a los archivos del smartphone. Debe existir un archivo de texto.	
<b>Flujo Normal:</b> La aplicación detecta texto	
Aplicación	Sistema
1.- Busca el archivo de texto que se generó el C.U.S Reconocer texto 2.- El texto es procesado en audio.	
<b>Postcondiciones:</b> Se genera un archivo de audio	

<b>Nombre:</b>	Reproducir audio
<b>Autor:</b>	Christian Cáceres
<b>Fecha:</b>	22/12/2020
<b>Descripción:</b> La aplicación reproduce el archivo de audio con el texto que se desea leer.	
<b>Actor:</b> Usuario.	
<b>Precondiciones:</b> Debe existir un archivo de audio con el texto que se pretende leer.	
<b>Flujo Normal:</b>	
Usuario	Aplicación

	1.- La aplicación busca el audio generado en el C.U.S Transformar texto en audio 2.-La aplicación reproduce el audio
<b>Flujo Alternativo:</b> El usuario detiene el audio	
Usuario	Aplicación
2.1- El usuario presiona la pantalla	2.2.- Detiene la reproducción del audio.

### 4.1.3. Diagrama de Secuencia

En las siguientes Figuras 4, 5, 6, 7 y 8 se muestran los diagramas de secuencias de los casos de uso previamente descritos.

#### 4.1.3.1. Inicia Aplicación

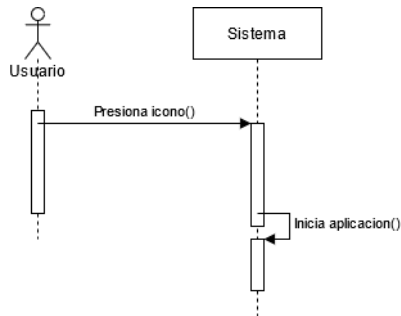


Figura 4: Diagrama para el Caso de Uso "Iniciar Aplicación"

#### 4.1.3.2. Capturar Video

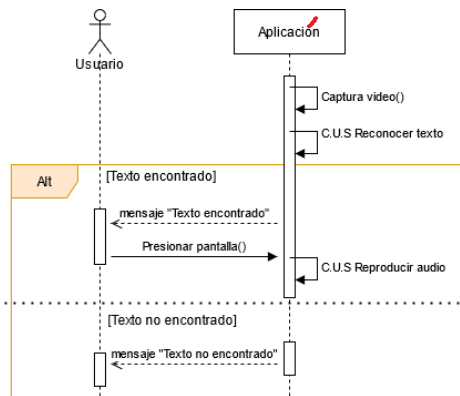


Figura 5: Diagrama para el Caso de Uso "Capturar Video"



### 4.1.3.3. Reconocer Texto

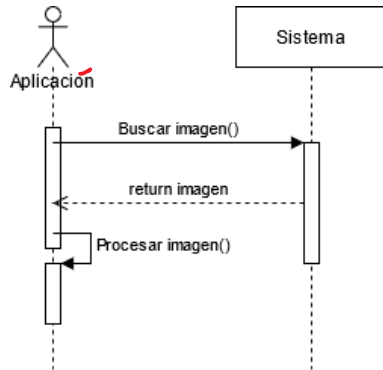


Figura 6: Diagrama para el Caso de Uso "Reconocer Texto"

### 4.1.3.4. Transformar texto en audio

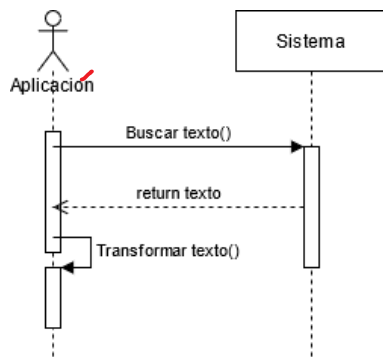


Figura 7: Diagrama para el Caso de Uso "Transformar texto en audio"

### 4.1.3.5. Reproducir Audio

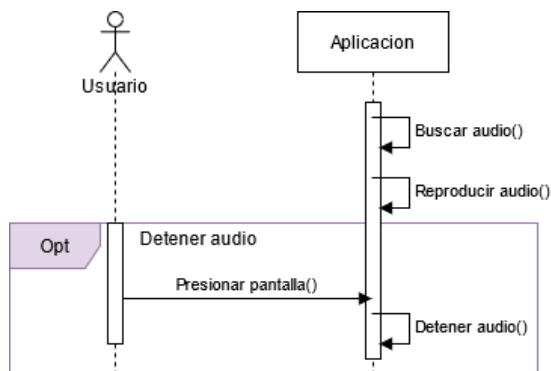


Figura 8: Diagrama para el Caso de Uso "Reproducir Audio"

### 4.1.4. Modelo de Clases

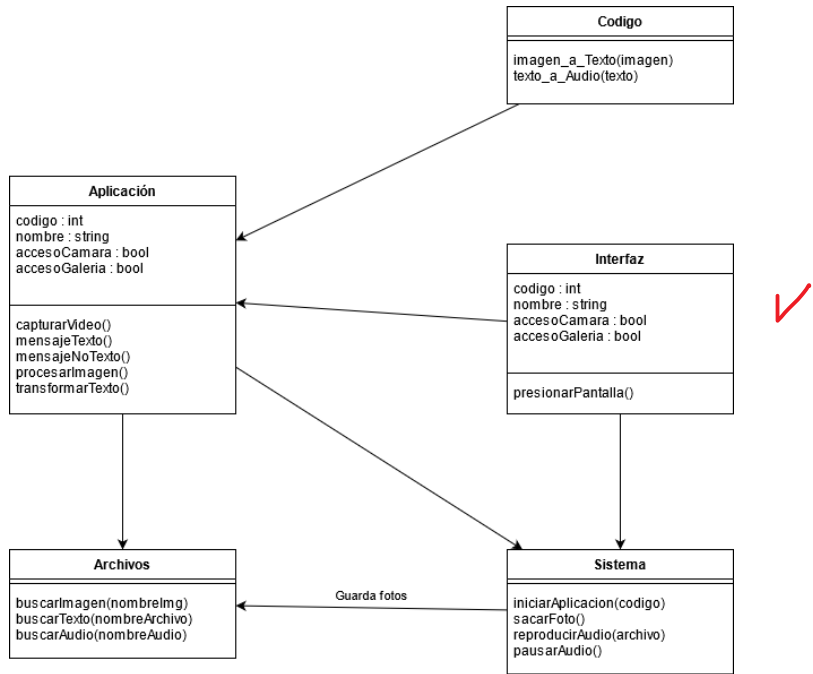


Figura 9: Modelo de Clases

En la Figura 9 se puede apreciar el Modelo de Clases relacionados a los Casos de Uso, el cual muestra las relaciones entre las clases y las funciones que contienen. ✓



#### 4.1.5. Descripción de la Arquitectura



Figura 10: Arquitectura del proyecto

En la Figura 10 se puede ver la Arquitectura del proyecto, en la cual:

1. Dispositivo móvil que utiliza un software con el que el cliente puede reproducir un audio del texto reconocido en una foto.
2. Interfaz de un software con el que se controla la aplicación.
3. Cámara del dispositivo móvil con la que se captura la foto a analizar.
4. Foto tomada por la cámara.
5. Texto extraído de la foto.
6. Dispositivo de reproducción de audio con el que el software reproducirá el texto.

Comentado [u2]: Los colores de la imagen son opacos



#### 4.1.6. Diseño de la Interfaz de Usuario



Figura 11: Simulación de vista inicial

En la Figura 11 se aprecia la interfaz de vista inicial de la aplicación, en esta vista la aplicación enviará un mensaje de voz cuando haya encontrado texto, luego el usuario podrá tocar una vez la pantalla para sacar reproducir el texto.



Figura 12: Simulación de la vista de reproducir audio

En la Figura 12 se aprecia la interfaz de vista al detectar texto de la aplicación, en esta vista el usuario podrá tocar una vez la pantalla para detener la reproducción del audio correspondiente.

#### 4.1.7. Especificación de requerimientos

Requerimiento Funcional	Descripción
La aplicación debe utilizar la cámara del smartphone	La aplicación debe poder acceder a la cámara del smartphone y debe ser capaz de tomar fotos y guardar archivos.
La aplicación debe poder reconocer texto	La aplicación debe tener la capacidad de reconocer texto a través de la cámara
La aplicación debe poder reproducir audio	La aplicación debe tener la capacidad de reproducir el texto a través del altavoz o audífonos

Requerimiento no Funcional	Descripción
La aplicación debe estar hecha en Python 3.6.8	La aplicación debe ser programada utilizando la versión de Python 3.6.8
La aplicación debe utilizar software OpenCV 4.0	Las imágenes que se capturen con la aplicación deben ser procesadas por el software OpenCV 4.0
La aplicación debe reconocer texto en un tiempo determinado.	La aplicación debe poder reconocer texto en menos de 1 minuto.
La aplicación debe reconocer texto correctamente.	La aplicación debe reconocer el texto que se quiere leer de manera correcta y fidedigna.
La aplicación debe reproducir texto en forma de audio correctamente.	La aplicación debe reproducir el texto que se quiere leer mediante un audio claro y entendible.
La aplicación debe ser instalada en Android.	La aplicación debe ser instalada en un dispositivo móvil Android 5.0.2 o superior.

## 4.2. Herramientas y técnicas

**Herramientas:** Google Docs, Microsoft Office, OneDrive, Visual Studio Code, Pycharm.

**Técnicas:** Prueba y error, Dividir para conquistar, etc.

## 5. Planificación de procesos de soporte

### 5.1. Planificación de la documentación

**Manual de usuario:** Usuario e Instalación. Este documento contendrá las indicaciones para que el usuario pueda manejar la aplicación de manera correcta.

**Wiki del proyecto:** Blog donde se documenta detalladamente la realización del proyecto y su propósito.

**Documentación del código:** Este documento detallara la finalidad de cada función dentro del código que compone la aplicación.

## 6. Plan de Integración

Para integrar cada parte de la aplicación se tuvieron que seguir los siguientes pasos:

1. Una vez terminado el módulo para la aplicación, hacer pruebas necesarias para un correcto funcionamiento.
2. Modificar el módulo para que funcionase en Kivy (para esto fue necesaria una pequeña investigación para reemplazar ciertas partes como la capturadora de video o el envío de mensajes).
3. Una vez hecho esto se agrega el módulo a la aplicación.
4. Se realizan pruebas para el funcionamiento del módulo.

## 7. Modelo de Implementación

Como fue mencionado en el punto 6, se necesitó modificar los módulos para la implementación con Kivy, un ejemplo de esto es en el caso de los módulos para mandar mensajes de voz (text to speech). El cambio necesario fue utilizar las funciones de texto a voz de "Plyer", una API que es independiente de la plataforma que se usa y permite utilizar funciones comunes para smartphones, en el caso de este módulo, el de mandar un texto como mensaje de voz para comunicarse con el Usuario.

## 8. Módulos Implementados

- **Detectar texto en tiempo real:** Este módulo es parecido al anterior, pero con la diferencia de que se utiliza para buscar texto en tiempo real, destacando el texto que encuentra en espacios rojos.

```

entr = 0
while True:
    cntr = cntr + 1
    if (cntr % 20) == 0:
        ret, frame = cap.read()
        hImg, wImg, _ = frame.shape
        boxes = pytesseract.image_to_data(frame)
        for z, b in enumerate(boxes.splitlines()):
            if z != 0:
                b = b.split()
                if len(b) == 12:
                    x, y, w, h = int(b[6]), int(b[7]), int(b[8]), int(b[9])
                    cv2.rectangle(frame, (x, y), (w + x, h + y), (0, 0, 255), 2)

        cv2.imshow("Video", frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

```

Figura 13: Módulo inicial creado en Python

En la figura 13 se puede ver el módulo inicial hecho para capturar el texto en tiempo real, este módulo luego fue pasado a Kivy y se tuvo que hacer cambios para que su funcionamiento fuera correcto, este nuevo código se puede ver en la figura 14.

```

ret, frame = self.capture.read()
if ret:
    hImg, wImg, _ = frame.shape
    boxes = pytesseract.image_to_data(frame)
    for z, b in enumerate(boxes.splitlines()):
        if z != 0:
            b = b.split()
            if len(b) == 12:
                x, y, w, h = int(b[6]), int(b[7]), int(b[8]), int(b[9])
                cv2.rectangle(frame, (x, y), (w + x, h + y), (0, 0, 255), 2)

```

Figura 144: Código modificado del módulo

- **Reconocer texto en una imagen:** En este módulo se utiliza OpenCV para modificar la imagen mediante filtros (de color a gris) y luego Pytesseract para que detecte el texto encontrado en la imagen.
- **Mandar mensajes en voz alta:** En este módulo se utiliza Plyer para mandar un mensaje de texto a voz.

Con formato: Fuente: Negrita

## 9. Pruebas

Se realizaron varios tipos de pruebas en el transcurso del proyecto, pero las más utilizadas fueron:

- **Prueba de lenguaje:** Estas pruebas fueron para familiarizarse con el lenguaje usado durante el proyecto (Python) y las funciones, librerías que se podían importar en este.
- **Prueba de módulo:** Estas pruebas fueron para verificar si el módulo hecho cumplía con la eficiencia y eficacia requeridas.
- **Prueba de interfaz:** Estas pruebas fueron para aprender a utilizar Kivy y como hacer la interfaz para la aplicación.

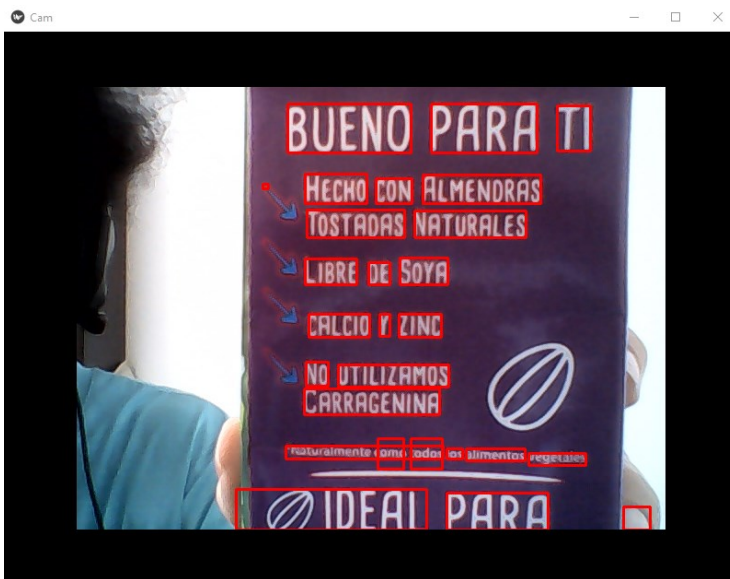


Figura 15: Prueba de captura de texto en tiempo real

En la figura 15 se puede ver una de las pruebas realizadas para la captura de texto en tiempo real, en esta se puede ver como la aplicación detecta el texto y coloca bordes rojos rodeando el texto identificado.

## 10. Conclusiones

Para concluir este informe, se debe mencionar la importancia de optimizar la productividad de las actividades y procesos que transcurren durante el proyecto, para esto es necesario priorizar los puntos a investigar, de modo que la investigación se centre en los aspectos fundamentales a conocer y sea fructífera para el conocimiento del grupo. De igual manera es fundamental utilizar todos los recursos disponibles que se tienen para conseguir información de manera eficaz. ✓

Por otro lado, se debe hacer una buena gestión de los riesgos y de los tiempos del proyecto. Por ejemplo, se puede señalar que ha habido un par de demoras con respecto a las fechas estipuladas en la carta Gantt, además se ha tenido problemas con el desarrollo de la interfaz de aplicación con Kivy y también un integrante del grupo ha tenido problemas del hardware. Por otro lado, el desarrollo de los módulos no ha sido complicado una vez se hizo la investigación necesaria.

Por último, es notable recalcar la flexibilidad que debe tener el grupo con la realización de las actividades del proyecto. Por ejemplo, luego de completar los módulos el grupo se redistribuyó las actividades para implementar la aplicación y concluir los entregables establecidos. ✓

## 11. Referencias

- Apuntes del Curso "Proyecto 2".
- Apuntes del Curso "Taller de Técnicas de Programación".
- Materiales del Curso "Formato Bitacora.pdf".
- Materiales del Curso "Propuesta formato plan de proyecto 2.docx".
- Materiales del Curso "Jit 1 ImageProcessing.pdf".
- Materiales del Curso "Jit 2 Filtros y Rasgos.pdf".
- Materiales del Curso "Entregables.pdf".
- Materiales del Curso "Proyecto II Piloto 2do Semestre 2020v1.0.pdf".

