

---

# Proyecto de Reconocimiento Facial

---



**Primer cuatrimestre 2024 - Universidad Nacional de General Sarmiento - Comisión 02**

**Nombre del proyecto: InnovaSoft**

**Nombre del producto: Cypher Vault**

**Profesores:**

- Ing. Francisco Orozco De La Hoz @ forozco@campus.ungs.edu.ar
- Lic. Leandro Dikenstein @ ldikenstein@campus.ungs.edu.ar

**Equipo de trabajo:**

- Flavio Ybarra - flavio\_712@hotmail.com - DNI: 36322712
- Alejandro Moras
- Javier Galeano - javi\_b\_galeano@hotmail.com - DNI: 41805228
- Melanie Ibarra - meluibarra15@gmail.com - DNI: 44301664
- Ivan Sanchez - ivansncz11@gmail.com - DNI: 42087962
- Fernando Trejo - fernandotrejo125@gmail.com - DNI: 43986607

# Glosario

---

1. [Introducción](#)
  2. [Objetivos del Proyecto](#)
  3. [Objetivos del Documento](#)
  4. [Definición de Roles](#)
  5. [Equipo de Trabajo y Roles](#)
  6. [Metodología](#)
  7. [Enlaces](#)
  8. [Gestión](#)
    1. [Plan de comunicaciones](#)
    2. [Requerimientos](#)
    3. [WBS](#)
    4. [Diccionario](#)
    5. [Calendario](#)
    6. [Estimaciones iniciales](#)
    7. [Riesgos](#)
    8. [Entregables](#)
    9. [Administración en el Manejo de Bugs](#)
    10. [Administración de Cambios](#)
    11. [Indicadores a Utilizar](#)
    12. [Tecnologías](#)
- 

## Introducción

Bienvenidos al proyecto Cypher Vault de Autenticación Facial en Kotlin. Este documento tiene como objetivo dar a conocer el desarrollo de una aplicación capaz de autenticar a los usuarios a través del reconocimiento facial enfocado para Tablets. La misma está diseñada para funcionar sin conexión a internet ofreciendo un servicio de almacenamiento de imágenes seguro y encriptado para el usuario.

## Objetivos del Proyecto

1. Desarrollar una aplicación de reconocimiento facial.
2. Implementar la aplicación en Android usando Kotlin y Android Studio.
3. Utilizar OpenCV y TensorFlow Lite para el reconocimiento facial.
4. Almacenar en el dispositivo imágenes cifradas/encriptadas.
5. Facilitar el uso de la aplicación para cualquier tipo de usuario.
6. Optimizar la aplicación para un uso eficiente de los recursos del dispositivo.

## Objetivos del Documento

Este documento tiene como objetivo explicar cuáles son los pasos a seguir en el ciclo de vida del desarrollo de este software, es decir, se detallarán los requerimientos funcionales, no funcionales, armado de la WBS (funcionalidades del proyecto), definición de roles, estimaciones de implementación y diagrama de arquitectura. Más adelante se detalla mejor el objetivo de cada uno.

## Definición de Roles

- **Líder del Proyecto:** Es el individuo responsable de administrar el proyecto.
- **Scrum Master:** Supervisa el progreso del proyecto y se asegura de que se cumplan los plazos.
- **Desarrolladores:** Encargados de la codificación y la implementación de la aplicación.
- **Tester:** Responsable de probar la aplicación en diferentes escenarios y reportar errores.

## Equipo de Trabajo y Roles

| Nombre          | Rol Primario       | Rol Secundario                  |
|-----------------|--------------------|---------------------------------|
| Profesor        | Líder del Proyecto | Sponsor                         |
| Flavio Ybarra   | Scrum Master       | Tester                          |
| Alejandro Moras | Desarrollador      | UX/UI                           |
| Fernando Trejo  | Desarrollador      | UX/UI                           |
| Javier Galeano  | Desarrollador      | UX/UI                           |
| Ivan Sanchez    | Tester             | Capacitador y Prueba de Usuario |
| Melanie Ibarra  | Tester             | Scrum Master                    |

## Metodología

En este proyecto, implementaremos una combinación de metodologías ágiles y Waterfall, también conocida como "Wagile" o "Agilefall". Este enfoque nos permitirá aprovechar lo mejor de ambos métodos para adaptarnos a las necesidades específicas de nuestro equipo de seis personas. A continuación les presentaremos un resumen de cómo lo haremos:

- **Comprender las metodologías:** Todo el equipo debe entender Agile y Waterfall.
- **Identificar las fases del proyecto:** Dividiremos el proyecto en fases claramente definidas.
- **Aplicar Waterfall en las fases iniciales:** Usaremos Waterfall para la planificación, análisis de requerimientos y diseño.
- **Implementar Agile en las fases de desarrollo:** Aplicaremos Agile para las fases de desarrollo y pruebas.
- **Facilitar la comunicación y colaboración:** Fomentaremos la comunicación abierta y la colaboración durante todo el proceso.
- **Realizar retrospectivas periódicas:** Programaremos reuniones regulares de retrospectiva al final de cada fase o sprint.
- **Ser flexible y adaptativo:** Mantendremos una mentalidad flexible y adaptativa a medida que evolucione el proyecto.

Nuestro enfoque Agile se enfocará en Scrum, el cuál se basa en entregar funcionalidades de forma incremental, en períodos de dos semanas. Dentro de las mismas se realizan reuniones diarias del equipo para planificación, control y revisión del trabajo realizado hasta el momento.

Con este enfoque, nuestro equipo podrá gestionar eficazmente el proyecto, adaptarse a los cambios y entregar valor de manera constante y oportuna.

## Enlaces

- Repositorio: se decidió utilizar Github para que todos los miembros del equipo puedan acceder y trabajar con mayor comodidad. [Repositorio Github](#)
- WBS: Se decidió utilizar Miro que es una plataforma de colaboración digital para realizar la WBS. Por motivos de seguridad no se compartirá el link pero la misma se mostrará en la documentación.
- Herramientas a utilizar: Android Studio, OpenCV, TensorFlow Lite, Visual Studio.
- Comunicación de equipo: WhatsApp y Discord.
- User Stories: Trello
- Comunicación con el Líder del Proyecto: Telegram o Mail.
- Diagrama de arquitectura: draw.io

## Gestión

### Misión/Visión del Negocio

Nuestra visión: Aspirar en que sea una aplicación cómoda y fácil de usar dentro de los estándares de seguridad para así brindar tranquilidad y seguridad al usuario. Nuestra misión: Es crear una aplicación de almacenamiento de imágenes privadas las cuáles son encriptadas en el dispositivo, donde el usuario se registra e ingresa a través del reconocimiento facial mediante la utilización de la cámara frontal del dispositivo (tablet). Posteriormente el ingreso del usuario se realizará comparando la foto tomada con las imágenes guardadas en el dispositivo, estas imágenes están encriptadas y cifradas.

### Alcance

- Aplicación para dispositivos Android (Tablets)
- Registro por reconocimiento facial
- Login por reconocimiento fácil
- Almacenamiento de imágenes en el dispositivo

*Como aún hay poca información sobre la implementación de la app no podemos confirmar las funcionalidades que quedan por fuera del alcance*

### Plan de Comunicaciones

Para facilitar la comunicación utilizamos la plataforma WhatsApp y Discord, ya que nos proporcionan un canal de comunicación instantánea y versátil, que permite una interacción ágil entre los miembros del equipo. Esto nos permite compartir actualizaciones rápidas, discutir ideas y mantenernos conectados en tiempo real. Para gestionar las tareas y el progreso del proyecto, se utilizará Trello. Nos permitió crear un flujo de trabajo estructurado y asignar tareas, asegurando que cada miembro del equipo esté al tanto de sus responsabilidades y plazos.

### Requerimientos

En este apartado se detallarán los requerimientos del sistema, además se hará mención de la nomenclatura a utilizar para la clasificación de dichos requerimientos. Los requerimientos funcionales son aquellos que definen la funcionalidades que va a tener el software. Tales requerimientos se clasifican en estos tres tipos:

**Requerimientos esenciales:** Estos requerimientos hacen que el sistema tenga sentido, es decir, sin esta clases de funcionamientos no se cumplirían el objetivo que necesitan los usuarios.

**Requerimientos importantes:** Son aquellos que, si no están, el software funciona igual pero se limitará el funcionamiento.

**Requerimientos deseables:** Son componentes adicionales que pueden ser agregados al software pero su prioridad es la mínima.

Una vez explicado las clasificación de requerimientos funcionales, se hará a continuación mención de los requerimientos no funcionales:

**Requerimientos No funcionales:** El objetivo de estos requerimientos es explicar las limitaciones o restricciones que el sistema posee. Estos requisitos no tienen ningún impacto en la funcionalidad del software, pero garantizan que el sistema satisfaga las necesidades de los usuarios del sistema.

- **Funcionales:**

1. **Registro:**

1. El sistema debe ser capaz de capturar imágenes de la cámara frontal de la Tablet.
2. Se debe crear una interfaz de login donde el usuario se registre con sus datos (nombre y mail) y su rostro.
3. La interfaz debe tener un boton para capturar la imagen.
4. Tiene que solicitar los permisos necesarios para acceder a la camara
5. Se debera guardar la imagen en una base de datos almacenada en la tablet
6. El sistema debe ser capaz de detectar rostros en las imágenes capturadas.
7. El sistema debe ser capaz de identificar a las personas a partir de sus rostros.
8. Asociar las rostros de las personas a su cuenta de registro

2. **Autenticación**

1. Una vez registrado el usuario debe ser capaz de loguearse a su cuenta atraves de la verifcacion facial
2. El sistema debe ser capaz de autenticar a las personas comparando sus rostros con una base de datos de rostros conocidos almacenada en la Tablet.
3. El sistema debe mostrar un mensaje de "Acceso Permitido" o "Acceso Denegado" en la pantalla de la Tablet en función del resultado de la autenticación.
4. El sistema deberá registrar un log con los datos de ingresos (Hora, ID de persona, etc.)
5. El sistema deberá permitir una alternativa manual de ingreso ante posibles desconexiones (sin Wifi o datos).

3. **Perfil de usuario**

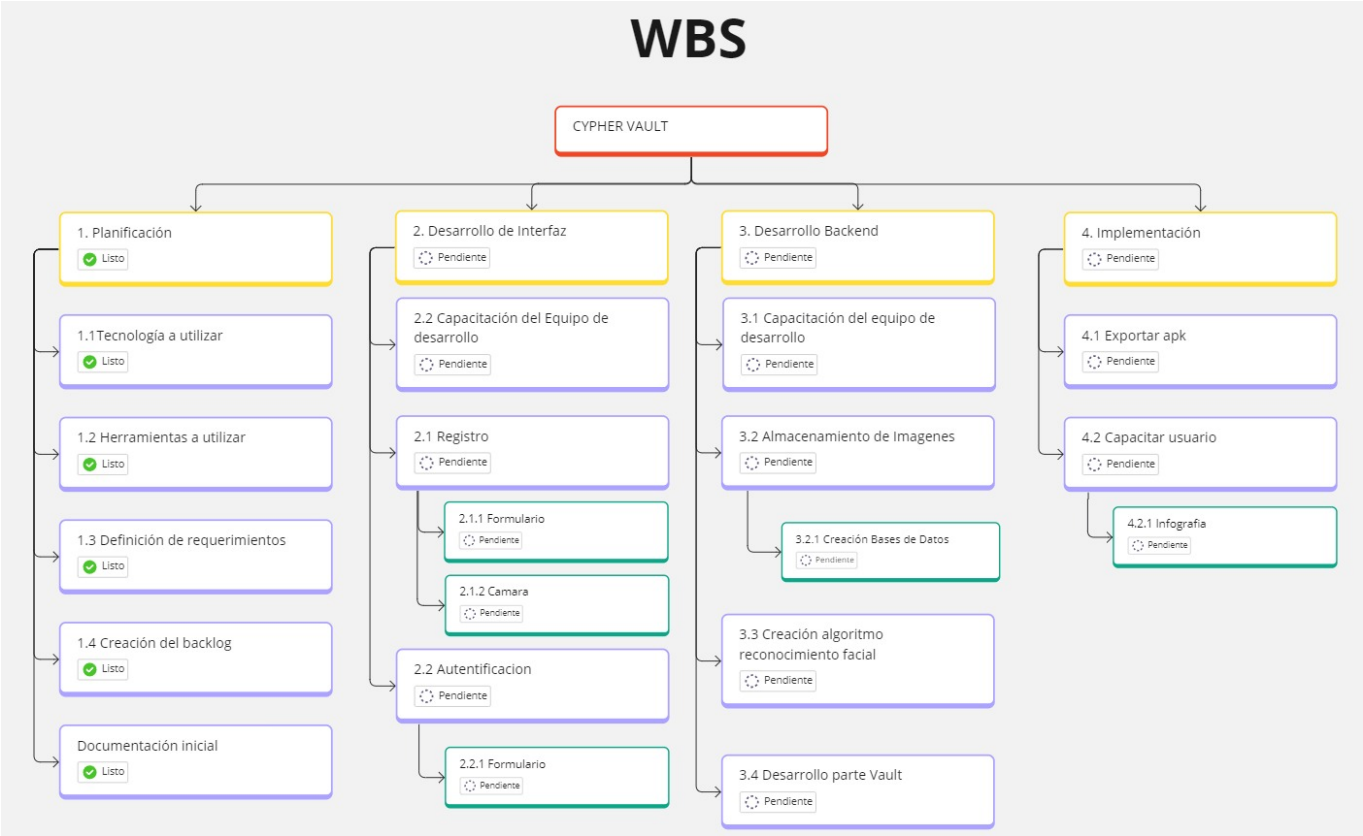
1. El sistema deberá permitir el ALTA/MODIFICACIONES de las personas a autenticar.
2. El usuario podra modificar su informacion personal o registrar otra foto de su rostro

- **No Funcionales:**

1. **Usabilidad**

1. Si el usuario desea entrar y no esta registrado se le debe mostrar un mensaje de "acceso denegado, primero necesitas registrarte"
2. Si el usuario desea ingresar a su cuenta con una foto u otro rostro (no asociado a su cuenta) se le debe mostrar un mensaje de "acceso denegado".
3. Si el usuario pudo ingresar a su cuenta se le debe mostrar un mensaje de "acceso permitido".
2. Rendimiento
1. El sistema debe ser eficiente en el uso de la batería, la memoria y el procesador de la Tablet.

WBS



Diccionario

Los pesos se clasifican en base a:

- 3: Esencial
- 2: Importante
- 1: Deseable

| ID  | NOMBRE                  | DESCRIPCIÓN                                    | TAREA                        | RESPONSABLE    | PESO |
|-----|-------------------------|--|------------------------------|----------------|------|
| 1   | Planificación           | Planificación sobre las tareas que la componen | Planificación, investigación | Todo el equipo | 3    |
| 1.1 | Tecnologías a utilizar  | Investigar sobre las tecnologías a utilizar    | Investigación                | Todo el equipo | 3    |
| 1.2 | Herramientas a utilizar | Investigar sobre las herramientas a utilizar   | Investigación                | Todo el equipo | 3    |

| ID    | NOMBRE                                | DESCRIPCIÓN  | TAREA         | RESPONSABLE          | PESO |
|-------|---------------------------------------|--|---------------|----------------------|------|
| 1.3   | Definición de requerimientos          | Definir los requerimientos del proyecto                          | Documentación | Todo el equipo       | 3    |
| 1.4   | Creación del backlog                  | Crear backlog  | Planificación | Scrum Master         | 3    |
| 1.5   | Documentación inicial                 | Crear la documentación   | Planificación | Todo el equipo       | 3    |
| 2     | Desarrollo de Interfaz                | Desarrollar la interfaz  | Desarrollo    | Equipo de desarrollo | 2    |
| 2.1   | Capacitación del Equipo de Desarrollo | Capacitar al equipo con las tecnologías a utilizar               | Capacitación  | Equipo de desarrollo | 2    |
| 2.2   | Registro                              | Crear interfaz registro  | Desarrollo    | Equipo de desarrollo | 2    |
| 2.2.1 | Formulario                            | Creación del formulario  | Desarrollo    | Equipo de desarrollo | 2    |
| 2.2.2 | Cámara                                | Implementar cámara en interfaz                                   | Desarrollo    | Equipo de desarrollo | 2    |
| 2.3   | Autenticación                         | Implementar interfaz de autenticación                            | Desarrollo    | Equipo de desarrollo | 2    |
| 2.3.1 | Formulario                            | Creación de formulario de autenticación                          | Desarrollo    | Equipo de desarrollo | 2    |
| 3     | Desarrollo backend                    | Desarrollar la lógica de la aplicación                           | Desarrollo    | Equipo de desarrollo | 3    |
| 3.1   | Capacitación del equipo de desarrollo | Capacitar al equipo de desarrollo con las tecnologías a utilizar | Capacitación  | Equipo de desarrollo | 2    |
| 3.2   | Almacenamiento de imágenes            | Almacenar las imágenes en el dispositivo del usuario             | Desarrollo    | Equipo de desarrollo | 2    |
| 3.2.1 | Creación de Base de Datos             | Crear base de datos  | Desarrollo    | Equipo de desarrollo | 2    |
| 3.3   | Crear algoritmo reconocimiento facial | Implementación de lógica de reconocimiento facial                | Desarrollo    | Equipo de desarrollo | 3    |
| 3.4   | Desarrollo parte Vault                | Implementar aplicación de galería privada                        | Desarrollo    | Equipo de desarrollo | 3    |

| ID    | NOMBRE            | DESCRIPCIÓN   | TAREA          | RESPONSABLE   | PESO |
|-------|-------------------|---|----------------|---|------|
| 4     | Implementación    | Puesta en servicio de la aplicación                       | Implementación | Capacitador y Prueba de Usuario, Equipo de desarrollo | 2    |
| 4.1   | Exportar apk      | Compilación del proyecto a formato de dispositivo android | Implementación | Equipo de desarrollo                                  | 2    |
| 4.2   | Capacitar usuario | Capacitar a usuario final                                 | Capacitación   | Capacitador y Prueba de Usuario                       | 2    |
| 4.2.1 | Infografía        | Mostrar imagen de uso                                     | Capacitación   | Capacitador y Prueba de Usuario                       | 2    |

Calendario

| Entrega | Fecha  | Tareas           |
|---------|--------|------------------|
| 1       | (19/4) | Sprint 0         |
| 2       | (3/4)  | Reunión Formal 1 |
| 3       | (17/5) | Reunión Formal 2 |
| 4       | (31/5) | Reunión Formal 3 |
| 5       | (14/6) | Reunión Formal 4 |
| 6       | (28/6) | Reunión Formal 5 |

Estimaciones Iniciales

Se entregará un prototipo de la interfaz funcional para el registro, la autenticación y base de datos. Estimamos que el tiempo empleado será:

- **Capacitación del equipo en las tecnologías:** 5hs por cada desarrollador y tester.
- **Desarrollo:** 20hs por cada desarrollador.
- **Testing:** 10hs por tester.

Riesgos

- Baja de un integrante del equipo.
- Superposición de tareas fuera de la materia.
- Enfermedad de un integrante.
- Dificultad en la comprensión de nuevas tecnologías.
- Impedimento con las herramientas de trabajo.
- Planificación inadecuada por falta de experiencia.
- Conflicto en el equipo en cuanto a comunicación.
- Cambios excesivos de los requerimientos.



- Retrasos en el desarrollo.

## Entregables

Definimos los hitos que ocurrirán en las diferentes fechas del proyecto. El primer hito se enfocará en la presentación formal del proyecto al cliente. En esta se explicará el plan de gestión que tendremos para administrar el proyecto. En los hitos restantes se presentará al cliente los avances en el producto.

1. Presentación del proyecto el día 19/09
2. Reunión formal 1 el día 19/04
3. Reunión formal 2 el día 03/05
4. Reunión formal 3 el día 17/05
5. Reunión formal 4 el día 31/05
6. Reunión formal 5 el día 14/06
7. Presentación final el día 28/06

## Administración en el Manejo de Bugs

Ejecutar una gestión eficaz de errores y pruebas es un componente esencial en un sistema de software. Estas tareas son vitales para asegurar que el sistema opere de forma fiable, eficiente y satisfaga las necesidades de los usuarios.

- Haremos un seguimiento de los errores en Trello, proporcionando en la tarjeta una explicación del fallo, el entorno en el que se produce, y datos útiles para su posible corrección. Los categorizaremos en tres niveles de acuerdo a su severidad: bajo, medio o alto. Esto nos permitirá determinar cuáles son las dificultades más urgentes y cuáles son de menor prioridad.

## Administración de Cambios

La administración eficiente de cambios es crucial para el triunfo de cualquier proyecto. Los cambios pueden aparecer por varias causas, como nuevas demandas del cliente, hallazgos durante el desarrollo, o variaciones en las condiciones del mercado. En este informe, se explica cómo planeamos registrar, validar y supervisar las modificaciones en nuestro proyecto, así como quién es el encargado de cada una de estas tareas.

Para registrar los cambios, emplearemos diversas herramientas y procedimientos:

- Trello: como se mencionó previamente, será nuestra plataforma principal para el seguimiento y administración de modificaciones.
- Recolección de peticiones de cambios: se anexarán todas las peticiones de cambios que incluirán detalles exhaustivos sobre cada petición de modificación. Esto abarcará una descripción de la modificación, su razón, el impacto anticipado en el proyecto y cualquier documentación de apoyo pertinente.
- Informe de Progreso: se mantendrá un informe de progreso actualizado que contendrá un registro de todas los cambios aprobados y pendientes. Esto ofrecerá una visión general del estado actual del proyecto en relación con las modificaciones.

## Indicadores a Utilizar

- Funcionalidad Completa
- Nivel de Calidad

- Evolución de la Prueba
- Burndown Chart

## Tecnologías

- Android Studio: Es un entorno de desarrollo integrado gratuito diseñado específicamente para el desarrollo de aplicaciones Android.
- Kotlin: Kotlin Es un lenguaje de programación de código abierto para aplicaciones Android.
- OpenCV: Es una biblioteca que proporciona una amplia gama de funciones y algoritmos para el procesamiento de imágenes y vídeo
- TensorFlow Lite: TensorFlow Lite es un marco de trabajo ligero desarrollado por Google que permite ejecutar modelos de aprendizaje automático en dispositivos móviles e integrados.
- SQLite: SQLite es un sistema de gestión de bases de datos relacional (RDBMS) ligero, rápido, autónomo y de código abierto.