**EBI  
git:** [**https://github.com/BrianGbrielMarinoBagnato/EBI-Municipio-2025.git**](https://github.com/BrianGbrielMarinoBagnato/EBI-Municipio-2025.git)

EBI(Escáner Biométrico Inteligente) se trata de un sistema de seguridad que permite detectar intrusos y mandar alertas en tiempo real. Primeramente, se carga en una base de datos el nombre (opcional), DNI (opcional) y luego se carga la foto del intruso. Cuando se carga la foto, automáticamente la aplicación se coloca en modo operando/detectando. El modo Operando implica en una cámara nuevamente habilitada lista para capturar rostros. Se van a detectar todos los rostros que pasen por la cámara o mejor dicho, que la cámara detecte. Si el rostro detectado coincide con alguno registrado ya en la base de datos intrusos, suena una alarma por 5 segundos, y manda una alerta por Gmail, SMS o WhatsApp a un número o la policía directamente

# **Integrantes - Roles**

**Aclaración: Los involucrados en la interfaz gráfica hay que ver si es posible vincularlo con las librerías openCV. Me refiero si es necesario que tengan conocimiento del tema ese también. Solamente por ahora vayamos HITO por HITO y vamos viendo**

[**Juan Pablo LUCIO**](mailto:jplucio@itel.edu.ar): **Administrativo -** Se encarga del papeleo, a pasar a limpio los informes y a recolectar datos para el informe y carpeta de campo (la carpeta de campo no es obligatoria pero suma puntos a la larga)

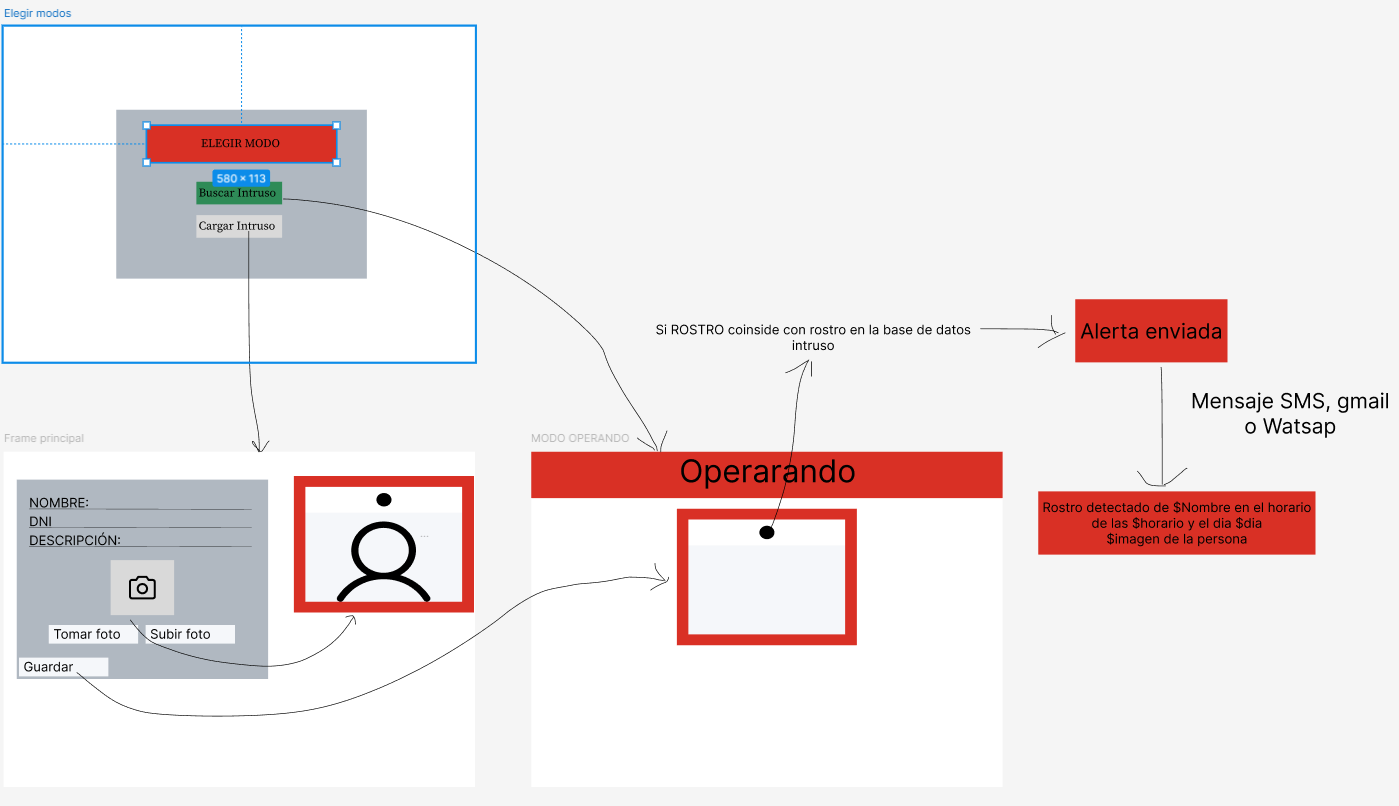
[**Agustin Pablo DICUNDO**](mailto:apdicundo@itel.edu.ar): **Diseñador, Programador -** Encargado de realizar la finalización del figma con sus detalles específicos y luego con maurin tendrá que hacer la interfaz para lograr el HITO 1 y 2 con maurin.

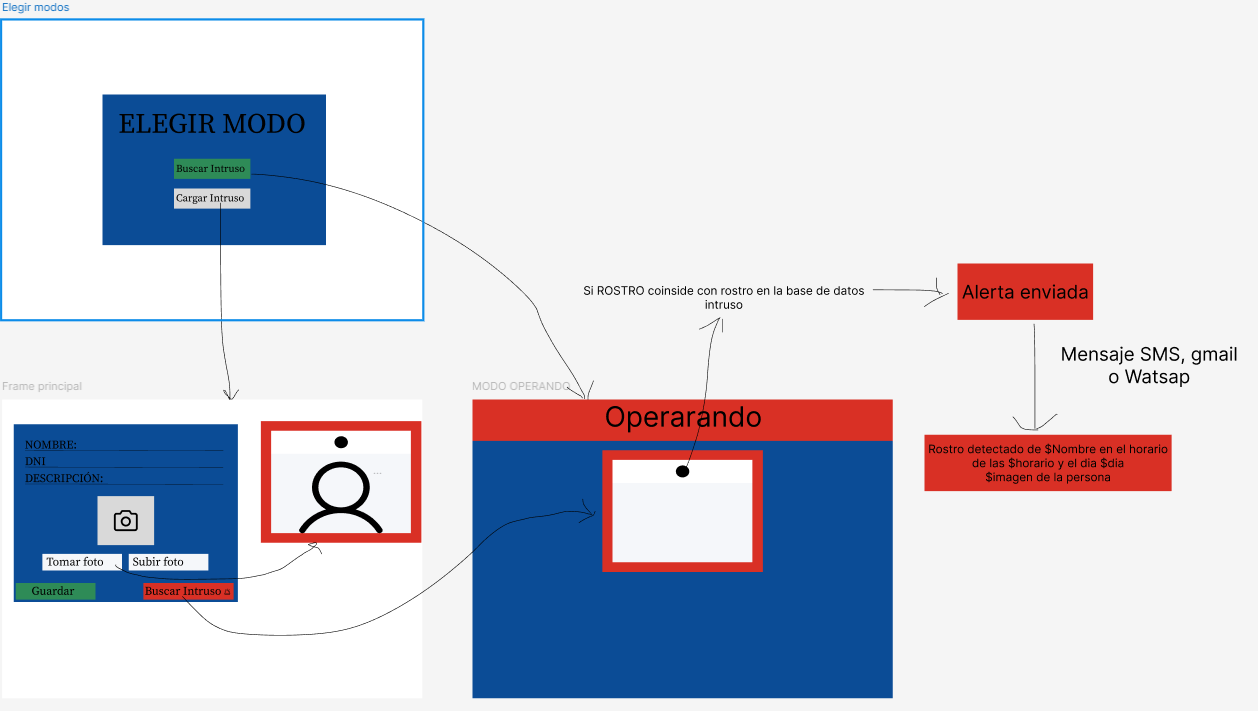
**Valentín SOVERON ISELLE:**  **Diseñador, Testeador, Detalles finales, Diseño -** Se encargará mas que nada en los detalles finales del diseño y tendrá intervención en el HITO 2 y 3 más orientado a la detección de rostros y procesamiento de imágenes

**Santiago Alberto MAURIN:** **Programador base de datos, programación general -** Más que nada se encarga de la base de datos a nivel general. Que toda la base de datos funcione y ande en óptimas condiciones. También se verá involucrado en la interfaz gráfica con agustín. Maurin estara actuando mas en el hito 2

[**Brian Gabriel Marino BAGNATO**](mailto:bgmbagnato@itel.edu.ar)**: Líder, coordinador de tareas -** Responsable de la organización del proyecto, la distribución de roles. Encargado de la **Programación** del HITO 2 Y 3. Finalización de los detalles en el figma y definir librerías para comenzar

# **IDEA DE FLUJO. Procesos de mejora**

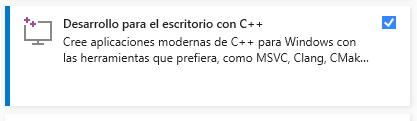




# **Librerías Recomendadas**

ANTES DE TODO HACER LO SIGUIENTE:  
Instalar CMake:

* + Descarga CMake desde [cmake.org/download](https://cmake.org/download/)
  + Ejecuta el instalador y marca la opción:  
    ✅ *"Add CMake to the system PATH for all users"*

1. Instalar Microsoft Visual C++ Build Tools:  
   
   * Descarga desde [Visual Studio Build Tools](https://visualstudio.microsoft.com/es/visual-cpp-build-tools/)
   * Ejecuta el instalador y selecciona:  
     ✅ *"Desktop development with C++"* (Incluye Windows 10/11 SDK)
2. Reinicia tu computadora (para que los cambios en el PATH surtan efecto).

·  [opencv-python](https://docs.opencv.org/4.x/) (*Procesamiento de imágenes y video*)

·  [face\_recognition](https://face-recognition.readthedocs.io/) (*Reconocimiento facial fácil*)

·  [numpy](https://numpy.org/doc/) (*Arrays numéricos y cálculos*)

·  [Pillow](https://pillow.readthedocs.io/) (*Manipulación de imágenes*)

·  [sqlite3](https://docs.python.org/3/library/sqlite3.html) (*Base de datos local - Documentación oficial de Python*)

·  [twilio](https://www.twilio.com/docs/libraries/python) *(*Envío de SMS/WhatsApp*)*

·  [smtplib](https://docs.python.org/3/library/smtplib.html) *(*Envío de emails - Documentación oficial de Python*)*

·  [pygame](https://www.pygame.org/news)  *(*Reproducir sonidos*)*

·  [tkinter](https://docs.python.org/3/library/tkinter.html) *(*Interfaz gráfica - Documentación oficial de Python*)*

*py -m pip install opencv-python face-recognition numpy Pillow pygame*

# **HITOS**

## **HITO 0: Creación e inicialización del proyecto**

***Fecha de Inicialización: 3/7***

***Fecha de finalización: 6/7***

**Objetivo 0.1:** Definir el diseño del figma total antes de programar. En base a eso ajustar los objetivos de cada HITO depende si se saca o agregan cosas

**Objetivo 1.1: Instalaciones**

* Python 3.10 (versión 3.10 si o si)
* vsCode
* GitHub Desktop
* Figma Desktop (Se actualiza en tiempo real)
* draw.io para diagramar (Para maurin)

**Objetivo 2:** Instalar extensiones en vsCode:

· Live server

· Live Server (Five Server)

· Live share

· Pylance

· Python

· Python debugger

· Python ident

· GitLens

· ChatGPT GPT-4o - BHITO AI Code Assistant

· GitHub Pull Requests

**Objetivo 3:** Clonar repositorio de GitHub al vsCode

**Objetivo 4:** Qué librerías se usarán (Analizar)

**Objetivo 4.1:** Instalar las librerías

## **HITO 1: Creación de la pantalla operaciones**

***Fecha de Inicialización: 7/7***

***Fecha de finalización: 14/7***

**Objetivo 1:** Crear la interfaz con Tkinter

**Objetivo 2:** agregar los botones

**Objetivo 3:** Que cada botón tenga su dirección específica a cada frame

## **HITO 2: crear el frame de cargar intrusos**

***Fecha de Inicialización: 14/7***

***Fecha de finalización: 21/7***

**Objetivo 1:** Diseñar la base de datos en draw

**Objetivo 1.1:** Crear la base de datos (usando sqlite)

**Objetivo 2:** Crear la funcionalidad del botón de tomar una foto del rostro

**Objetivo 2.1:** crear la funcionalidad de cargar una foto y que solamente tome el recuadro de la cara

**objetivo 3:** Que el botón guardar capture los datos de nombre, DNI, descripción, foto, fecha de carga. El único dato obligatorio es la foto.

## **HITO 3: cargar el frame de Detectar intruso**

***Fecha de Inicialización: 21/7***

***Fecha de finalización: 4/8***

**objetivo 1:** La camara este abierta constantemente

**objetivo 2:** Si se detecta un rostro, que se compare con todos los que hay en la base de datos

**objetivo 2.1:** Si el rostro coincide, que se mande una alerta a whatsapp, SMS o gmail o los 3 al mismo tiempo. Con el mensaje: “La persona X con el dni X cuya descripción X se lo encontró a las X horas. La foto que se capturó este en el mensaje

**objetivo 3:** La tecla barra espaciadora termina el programa

**objetivo 4:** El botón volver te lleva a elegir el modo de operación