



**UNIVERSIDAD DE IXTLAHUACA  
UICUI**

**FACULTAD DE INGENIERÍA**

**MATERIA**

**Tecnologías computacionales**

**Profesor**

**Jose Alfredo Zuñiga Lara**

**Proyecto 2do Parcial**

**Alumnos**

**Francisco Reginaldo Tapia Garcia 33.3%**

**Zarco Salinas Alejandro 33.3%**

**Garduño Alvarez Jessica 33.3%**

# Índice

|  |          |
|--|----------|
| <b>Índice</b>  | <b>2</b> |
| <b>Introducción</b>  | <b>5</b> |
| <b>Planteamiento del problema</b>                              | <b>5</b> |
| <b>Objetivo general</b>  | <b>5</b> |
| <b>Objetivos específicos</b>                                   | <b>5</b> |
| <b>Justificación</b>   | <b>6</b> |
| <b>Desarrollo</b>  | <b>7</b> |
| Herramientas y Tecnologías Utilizadas                          | 7        |
| Pantalla principal – Navegación básica                         | 15       |
| Componentes involucrados                                       | 15       |
| Lógica del Bloque - Evento Button1.Presionar                   | 15       |
| Uso típico   | 15       |
| Resumen funcional  | 16       |
| Descripción de eventos y componentes                           | 16       |
| Componentes involucrados:                                      | 16       |
| Bloques de evento:   | 16       |
| Pantalla: Screen4 – Inicio de sesión o verificación de usuario | 18       |
| Componentes involucrados                                       | 18       |
| Lógica del Bloque - Evento Botón1.Clic                         | 18       |
| Evento - Botón2.Clic   | 18       |
| Funcionamiento General   | 19       |
| Pantalla: Screen3 – Registro de usuario                        | 20       |
| Componentes involucrados                                       | 20       |
| Lógica del Bloque - Evento Botón1.Clic                         | 20       |
| Evento - CloudDB1.ObtenerValor                                 | 21       |
| Si el usuario no existe:                                       | 21       |
| Evento - Botón2.Clic   | 21       |
| Funcionamiento General   | 21       |
| Interfaz: Formulario de Denuncia Anónima                       | 25       |
| Propósito general  | 26       |
| Estructura y funcionamiento                                    | 26       |
| 1. Encabezado  | 26       |
| 2. Contenedor principal (.container)                           | 26       |
| Formulario de denuncia   | 26       |
| Campos del formulario:   | 27       |
| Botones:   | 27       |
| Funcionalidad del Script (JavaScript)                          | 27       |
| Nota informativa   | 28       |
| Componentes principales de la interfaz                         | 28       |
| 1. Encabezado (Header)   | 28       |
| 2. Panel principal: Consulta de denuncias                      | 29       |

|   |    |
|---|----|
| 3. Panel lateral: Preguntas Frecuentes (FAQ)                            | 29 |
| 4. Panel de Contacto  | 29 |
| Lógica de funcionalidad (JavaScript)                                    | 29 |
| Objetivo de la interfaz   | 29 |
| Interfaz: Formulario de Denuncia Digital                                | 29 |
| Encabezado  | 30 |
| Contenedor principal  | 30 |
| Datos del denunciante   | 31 |
| Categoría del incidente   | 31 |
| Ubicación del incidente   | 31 |
| Fecha y hora  | 31 |
| Alerta informativa  | 32 |
| Aplicación Web  | 32 |
| Interfaz: Página Principal del Sistema de Denuncias                     | 33 |
| Encabezado de navegación  | 34 |
| Introducción centrada   | 34 |
| Selector de modalidad   | 34 |
| Botón de consulta   | 34 |
| Tarjetas informativas   | 34 |
| Interfaz: Pantalla de Inicio de Sesión del Sistema de Denuncias         | 35 |
| Encabezado  | 35 |
| Contenedor central del formulario                                       | 35 |
| Opción de recordar sesión   | 36 |
| Botón de acción   | 36 |
| Acciones adicionales  | 36 |
| Interfaz: Pantalla de Registro de Usuario del Sistema de Denuncias      | 37 |
| Encabezado  | 37 |
| Contenedor central del formulario                                       | 38 |
| Formulario de registro  | 38 |
| Aceptación de términos  | 38 |
| Botones de acción   | 38 |
| Enlace para usuarios existentes   | 38 |
| Descripción de la Clase Control (app.py) – Proyecto Denuncias Digitales | 38 |
| 1. Inicialización y Configuración                                       | 39 |
| 2. Rutas de Interfaz (HTML)   | 39 |
| 3. Generación de Folios   | 39 |
| 4. Registro de Usuario  | 39 |
| 5. Inicio de Sesión (Login)   | 39 |
| 6. Cierre de Sesión   | 40 |
| 7. Envío de Denuncia Anónima  | 40 |
| 8. Envío de Denuncia Digital  | 40 |
| 9. Consulta de Denuncia por Folio                                       | 40 |

|                                       |           |
|---------------------------------------|-----------|
| 10. Ejecución Local                   | 40        |
| Observaciones Generales               | 40        |
| <b>Link y código QR para descarga</b> | <b>41</b> |
| <b>Conclusión</b>                     | <b>41</b> |
| <b>Bibliografía en formato APA</b>    | <b>41</b> |

# Introducción

En el marco de la asignatura de Tecnologías Computacionales, se desarrolló un proyecto de aplicación móvil con enfoque social, dirigido a mejorar la comunicación entre la ciudadanía del municipio de Ixtlahuaca y las autoridades locales. Esta aplicación permite registrar, enviar y dar seguimiento a denuncias de manera eficiente y accesible desde un dispositivo móvil.

La solución propuesta busca dar respuesta a la necesidad de una herramienta tecnológica que permita canalizar reportes ciudadanos de forma organizada, transparente y con posibilidad de anonimato, contribuyendo así al fortalecimiento de la participación ciudadana y la mejora del entorno comunitario.

## Planteamiento del problema

En el municipio de Ixtlahuaca, la ausencia de una plataforma digital unificada para reportar problemas de seguridad, irregularidades o situaciones delicadas ha generado un sistema ineficiente que depende principalmente de trámites presenciales y llamadas telefónicas. Este enfoque tradicional presenta múltiples limitaciones: falta de accesibilidad para quienes no pueden desplazarse a oficinas gubernamentales, ausencia de un mecanismo que garantice el anonimato y proteja a los denunciantes de posibles represalias, y la inexistencia de un sistema transparente que permita dar seguimiento a los reportes, lo que genera desconfianza en la ciudadanía.

Además, la centralización en procesos físicos excluye a personas con movilidad reducida o de zonas alejadas, limitando su participación. Como consecuencia, se dificulta la recopilación oportuna de denuncias y se reduce la capacidad de las autoridades para atender las problemáticas reportadas. Ante esta situación, se hace necesario desarrollar una aplicación web y móvil que funcione como un canal moderno, seguro e intuitivo, permitiendo a los ciudadanos reportar incidentes de manera anónima o identificable, consultar el estado de sus denuncias y, en última instancia, mejorar la comunicación entre la población y las instancias correspondientes.

## Objetivo general

Diseñar e implementar una aplicación móvil funcional y accesible para los habitantes de Ixtlahuaca, que facilite la realización de denuncias anónimas o identificables sobre problemas de seguridad, irregularidades o situaciones delicadas, así como el seguimiento de las mismas, utilizando la metodología SCRUM para garantizar un desarrollo colaborativo, incremental y centrado en la entrega continua de valor.

## Objetivos específicos

1. Desarrollar un módulo de denuncias anónimas que permita a los usuarios reportar problemas de seguridad o irregularidades sin revelar su identidad.

2. Implementar un sistema de denuncias ciudadanas con registro de datos identificables para casos que requieran seguimiento personalizado por parte de las autoridades.
3. Integrar una funcionalidad de consulta y seguimiento donde los usuarios puedan verificar el estado y las respuestas a sus denuncias previamente registradas.
4. Aplicar la metodología SCRUM en el desarrollo del proyecto, organizando el trabajo en Sprints con tareas definidas para asegurar una entrega incremental y eficiente.
5. Garantizar la accesibilidad y usabilidad de la aplicación mediante un diseño intuitivo y adaptable a distintos dispositivos móviles.

## Justificación

En el marco de la asignatura de Tecnologías Computacionales, se desarrolló este proyecto con enfoque social, buscando ofrecer una solución tecnológica que mejore la comunicación entre los habitantes del municipio de Ixtlahuaca y sus autoridades. La implementación de una aplicación móvil permitirá:

- Fortalecer la participación ciudadana al brindar un medio seguro, accesible y eficiente para reportar problemas, ya sea de forma anónima o identificable.
- Agilizar la respuesta de las autoridades mediante un sistema organizado que registre, clasifique y dé seguimiento a las denuncias en tiempo real.
- Promover la transparencia al permitir que los usuarios consulten el estado de sus reportes, reduciendo la opacidad en los procesos de atención.
- Incentivar el uso de tecnología con impacto social, demostrando cómo herramientas digitales pueden mejorar la calidad de vida en comunidades con problemáticas similares.

Esta aplicación no solo resuelve una necesidad inmediata en Ixtlahuaca, sino que también sienta un precedente para la implementación de soluciones tecnológicas en otros municipios que enfrenten desafíos similares. Al integrar metodologías ágiles como SCRUM, se asegura un desarrollo iterativo y centrado en el usuario, maximizando su funcionalidad y adaptabilidad.

# Desarrollo

**Materia:** Tecnologías Computacionales

**Plataforma de desarrollo:** App Inventor

**Metodología:** SCRUM

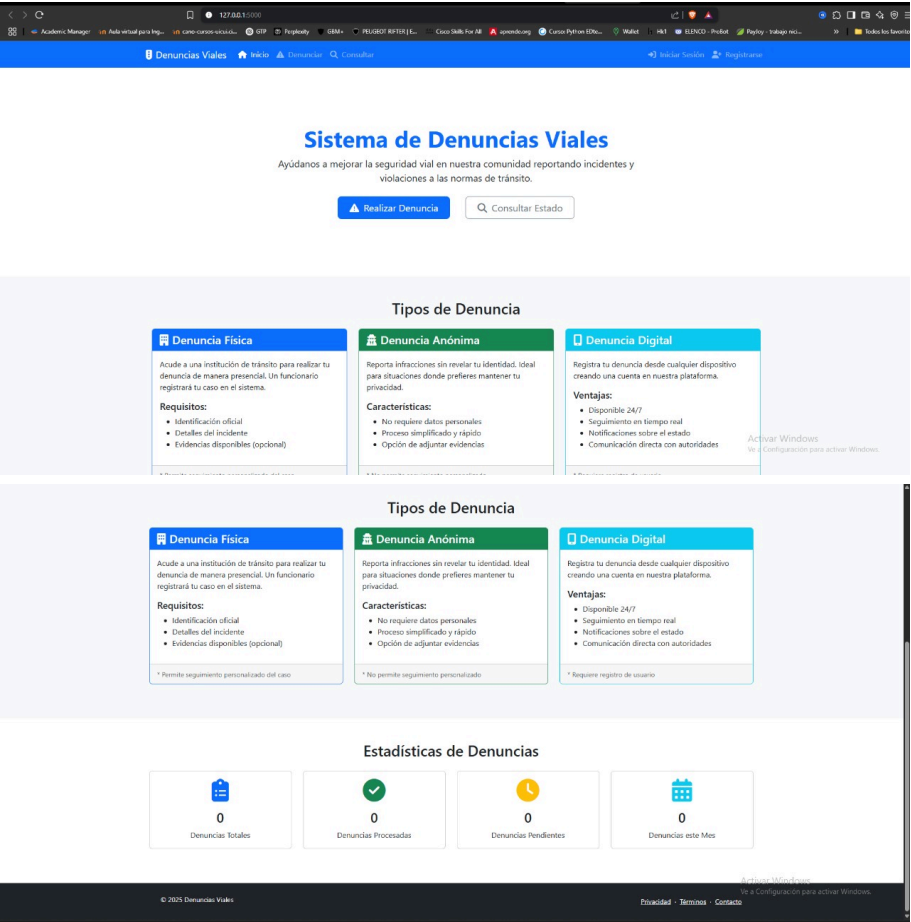
## Herramientas y Tecnologías Utilizadas

- **App Inventor:** Plataforma de desarrollo visual que facilita la creación de aplicaciones móviles mediante bloques de programación. Ideal para prototipos funcionales y proyectos educativos.
- **SCRUM:** Marco ágil de desarrollo de software centrado en la colaboración del equipo, revisiones periódicas y entregas parciales que permiten validar avances y adaptarse a cambios.

## Base de proyecto:

Para el desarrollo de este proyecto se tomó como base un proyecto web previo en donde se tenían como objeto el que los ciudadanos pudieran realizar denuncias ante un incidente, este proyecto permitía realizar las denuncias de 3 maneras:

1. **Denuncia física:** Se tuvo como objeto que un ciudadano no realizará la denuncia desde la página web, sino que esta misma le mostrará los lugares físicos en donde el ciudadano podría interactuar con el personal de las instituciones.
2. **Denuncia anónima:** este funcionaría para realizarlo de manera virtual en donde se detallaron los detalles del incidente, la ubicación del mismo, el tipo de incidente que se está desarrollando, pero sin la necesidad de que la persona dejará en el informe de los datos personales del informante y para seguir el caso y los detalles se le da un folio al denunciante.
3. **Denuncia formal:** Dentro de este tipo de denuncia está la existencia de un perfil, esto indicando que hubo un previo registro en la página web, y teniendo también la oportunidad de tener un historial en donde el podrá consultar sus denuncias y el seguimiento de estas además de las acciones que se están llevando y el resultado que se obtuvo en cada denuncia.



## Registro de usuarios

Denuncias Viales

Inicio

Denunciar

Consultar

Iniciar Sesión

Registrarse

Registro de Usuario

Complete el siguiente formulario para crear una cuenta. Los campos marcados con \* son obligatorios.

\* Nombre(s)

\* Apellido(s)

\* Correo Electrónico

Este correo se utilizará para iniciar sesión y recibir notificaciones.

\* Contraseña

\* Confirmar Contraseña

Teléfono

Formato: 10 dígitos sin espacios ni guiones.

CURP

El CURP es opcional pero ayuda a validar su identidad.

☐ \* Acepto los [términos y condiciones](#) y la [política de privacidad](#).

Crear Cuenta

Limpiar Formulario

¿Ya tiene una cuenta? [Iniciar Sesión](#)

## Inicio de Sesión



### Iniciar Sesión

Correo Electrónico

correo@ejemplo.com

Contraseña

☐ Recordar sesión

Iniciar Sesión

¿No tiene cuenta? [Registrarse](#)

¿Olvidó su contraseña?

Formulario de Denuncias

### Formulario de Denuncia Vial

Complete el siguiente formulario para reportar una infracción o incidente vial. Los campos marcados con \* son obligatorios.

\* Tipo de Denuncia

Seleccione un tipo de denuncia

La denuncia anónima no requiere datos personales, mientras que la digital permite dar seguimiento personalizado.

\* Categoría del Incidente

Seleccione una categoría

\* Ubicación del Incidente

Obtener mi Ubicación Actual

Latitud

Longitud

Dirección (calle, colonia, número)

Ej. Av. Insurgentes 123, Col. Roma

Si no tiene las coordenadas exactas, proporcione la dirección lo más detallada posible.

\* Fecha y Hora del Incidente

21/05/2025 09:38 a. m.

Placa del Vehículo

Ej. ABC-123

Si es posible, indique la placa del vehículo involucrado en el incidente.

\* Descripción del Incidente

Describe detalladamente lo que observó...

## Consultas

### Q Consulta de Denuncias

Ingrese el folio de su denuncia para verificar su estado actual. Los folios tienen el formato: DV20250501XXXXX.

Folio de Denuncia

# Ej. DV20250501XXXXX Q Consultar

El folio fue proporcionado cuando realizó su denuncia.

### Información sobre Estados

|                    |  |                  |                                       |
|--------------------|--|------------------|---------------------------------------|
| <b>Recibida</b>    | La denuncia ha sido registrada en el sistema | <b>Rechazada</b> | La denuncia no cumple los requisitos  |
| <b>En revisión</b> | Un funcionario está evaluando la denuncia    | <b>Procesada</b> | Se ha tomado acción sobre la denuncia |
| <b>Validada</b>    | La denuncia ha sido verificada               |                  |                                       |

### Preguntas Frecuentes

- ¿Cómo obtengo mi folio?
- ¿Qué debo hacer si perdí mi folio?
- ¿Cuánto tiempo tarda el proceso?
- ¿Puedo añadir información adicional?

### Contacto

Si tiene problemas para consultar su denuncia o requiere asistencia adicional, puede contactarnos:

- [sosporte@denunciavial.mx](mailto:sosporte@denunciavial.mx)
- [+52 \(55\) 1234-5678](tel:+525512345678)
- Lunes a Viernes, 9:00 - 18:00

## Descripción de proyecto base:

- Flask como Columna Vertebral

Flask funciona como el esqueleto central del sistema, gestionando todo el flujo de peticiones HTTP. Cuando un usuario accede a la URL /denunciar, Flask determina qué plantilla HTML enviar al navegador del usuario. Cuando se envía un formulario de denuncia, Flask procesa los datos recibidos, valida la información y orquesta el resto de componentes. El archivo app.py define todas las rutas mediante decoradores como `@app.route('/api/denuncias/crear', methods=['POST'])`, estableciendo así los endpoints disponibles y sus métodos HTTP permitidos (GET, POST, PUT, DELETE).

- Flask-SQLAlchemy como Puente a la Base de Datos

Esta extensión traduce las operaciones de base de datos a un lenguaje orientado a objetos. En lugar de escribir consultas SQL complejas, el sistema define clases Python como Usuario, Denuncia o Evidencia que reflejan las tablas de la base de datos. Cuando un usuario envía una denuncia, el sistema crea una instancia de la clase Denuncia con los datos recibidos

## Tecnologías Complementarias

- MySQL como Almacén de Datos

MySQL es un sistema de gestión de bases de datos relacionales que almacena toda la información de manera estructurada y eficiente. La aplicación define varias tablas interrelacionadas:

- Usuarios: Almacena información de los usuarios registrados
- Denuncias: Contiene los datos de las denuncias realizadas
- Evidencias: Guarda referencias a archivos subidos como evidencia
- Seguimientos: Registra actualizaciones y cambios de estado en las denuncias

La estructura relacional permite realizar consultas complejas, como obtener todas las denuncias de un usuario específico o filtrar denuncias por estado o categoría.

- Leaflet para Geolocalización

Los mapas interactivos son fundamentales en un sistema de denuncias viales. Leaflet es una biblioteca JavaScript que integra OpenStreetMap para proporcionar esta funcionalidad. Cuando un usuario registra una denuncia, puede marcar la ubicación exacta del incidente

## **Pyngrok para Desarrollo**

Durante la fase de desarrollo, Pyngrok crea un túnel seguro que expone el servidor local (generalmente solo accesible desde una computadora) a internet. Esto es especialmente útil para:

- Probar la aplicación en dispositivos móviles reales.
- Demostrar el sistema a clientes o stakeholders sin necesidad de un despliegue formal.
- Probar integraciones con servicios externos que requieren una URL pública.
- El sistema configura este túnel automáticamente al iniciar.

## **Flujo Completo de Datos**

Para ilustrar cómo todas estas tecnologías trabajan juntas, consideremos el proceso completo de realizar una denuncia:

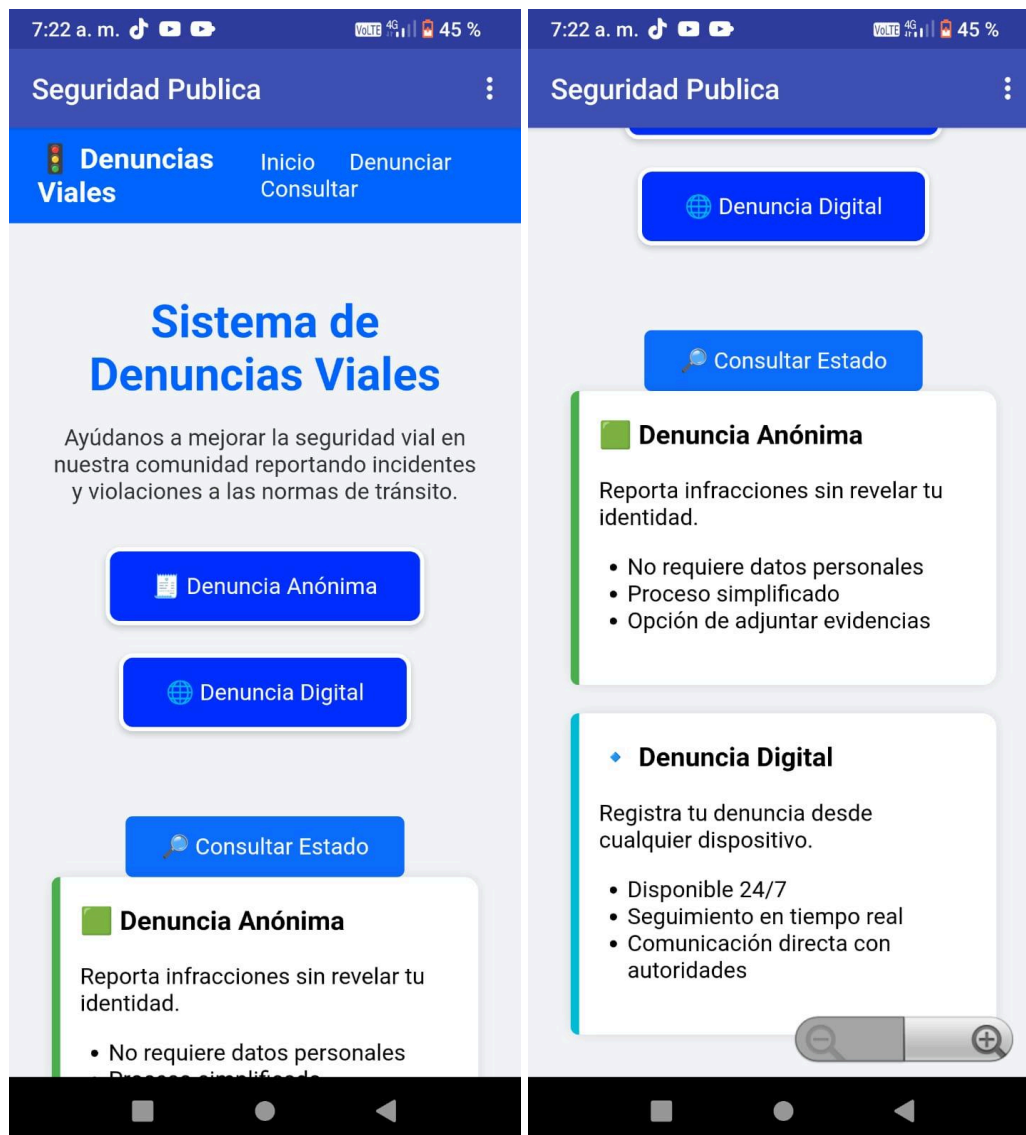
- Carga inicial: Flask sirve la página HTML de denuncia, que incluye Bootstrap para la estructura y estilos.
- Captura de ubicación: Al hacer clic en "Obtener ubicación", el navegador solicita permiso para acceder a la geolocalización. jQuery maneja este evento y, una vez obtenidas las coordenadas, inicializa un mapa Leaflet centrado en esa ubicación.
- Envío del formulario: Cuando el usuario completa los datos y envía el formulario, jQuery captura el evento y, mediante AJAX, envía los datos al servidor incluyendo el token JWT en los encabezados.
- Procesamiento en servidor: Flask recibe la petición y la ruta correcta del endpoint. Flask-JWT-Extended verifica la autenticación del usuario. La función controladora procesa los datos del formulario, crea una nueva instancia de Denuncia y la guarda en la base de datos mediante Flask-SQLAlchemy.
- Almacenamiento de evidencias: Si se adjuntan archivos, se guardan en el servidor y se crean registros en la tabla de evidencias.
- Respuesta al cliente: Flask genera una respuesta JSON con información sobre la denuncia creada y el folio asignado.
- Actualización de la interfaz: El código JavaScript recibe la respuesta, guarda el folio en localStorage y redirige al usuario a la página de confirmación, donde se muestra el folio generado.

## Desarrollo actual:

Para este proyecto se decidió que a los usuarios y las tecnologías más activas en la vida cotidiana, que son los smartphones se adaptaría la página web para dispositivos móviles en los cuales sería una forma mas comoda y practica para la realización de una denuncia, además de que con ella se podrían obtener las evidencias de manera instantánea y cómoda, la geolocalización permitirá que se pudiera ubicar los lugares de incidente de una manera más rápida y cómoda además de intuitiva.

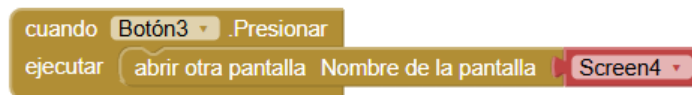
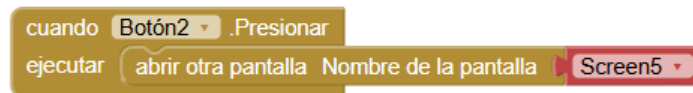
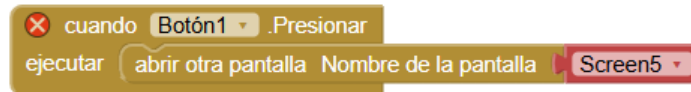
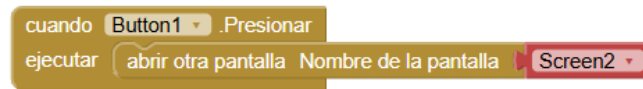
## Aplicación móvil

- Interfaz de Inicio



Esta interfaz permitirá a cualquier usuario seleccionar el tipo de denuncia disponible, además de presentar información acerca de dicho tipo de denuncia, y por último permite seleccionar consultar consulta para darle seguimiento a la misma.

- Interfaz de Inicio (Bloques)



## Pantalla principal – Navegación básica

### Componentes involucrados

- **Button1:** Botón utilizado para cambiar de pantalla.

### Lógica del Bloque - Evento Button1.Presionar

Cuando el usuario presiona **Button1**, se ejecuta la siguiente acción:

- **Abrir otra pantalla (Screen2):**  
Se abre la pantalla llamada **Screen2**. Este bloque permite navegar entre diferentes pantallas de la app, lo cual es útil para dividir funcionalidades o flujos del usuario, por ejemplo:
  - Pantalla de inicio
  - Pantalla de login
  - Pantalla de registro
  - Pantalla de reportes o denuncias

### Uso típico

Este tipo de bloque es ideal para una navegación simple. No se pasa ningún dato entre pantallas, solo se realiza el cambio de vista.

| Resumen funcional  |                     |
|--------------------|---------------------|
| Acción del usuario | Resultado en la app |

|                  |                               |
|------------------|-------------------------------|
| Presiona Button1 | Se abre la pantalla "Screen2" |
|------------------|-------------------------------|

## Descripción de eventos y componentes

La imagen muestra bloques de programación desarrollados en la plataforma **MIT App Inventor**, los cuales definen eventos asociados a tres botones dentro de una aplicación móvil. Cada bloque representa un evento que se ejecuta cuando el usuario presiona un botón específico en la interfaz.

### Componentes involucrados:

1. **Botón1**
2. **Botón2**
3. **Botón3**

Cada uno de estos botones se asocia a un evento del tipo **.Presionar** (equivalente a "Click" o "Click Event").

### Bloques de evento:

1. **Cuando Botón1 .Presionar**
  - Acción: Ejecuta el bloque **abrir otra pantalla**.
  - Pantalla destino: **Screen5**.
2. **Cuando Botón2 .Presionar**
  - Acción: Ejecuta el bloque **abrir otra pantalla**.
  - Pantalla destino: **Screen5**.
3. **Cuando Botón3 .Presionar**
  - Acción: Ejecuta el bloque **abrir otra pantalla**.
  - Pantalla destino: **Screen4**.

### Funcionamiento

Cuando el usuario interactúa con la aplicación presionando cualquiera de los botones mencionados, se desencadena un evento de navegación hacia otra pantalla:

- **Botón1 y Botón2:** Al presionarse, abren la pantalla llamada **Screen5**. Esto sugiere que ambos botones pueden estar diseñados para llevar al usuario a la misma sección o funcionalidad dentro de la app (por ejemplo, un formulario, menú avanzado o vista de detalles).
- **Botón3:** Al presionarse, redirige a la pantalla llamada **Screen4**, lo que indica una navegación hacia una sección distinta del aplicativo.

### Resumen de navegación:

| Botón | Evento | Acción realizada | Pantalla destino |
|-------|--------|------------------|------------------|
|-------|--------|------------------|------------------|

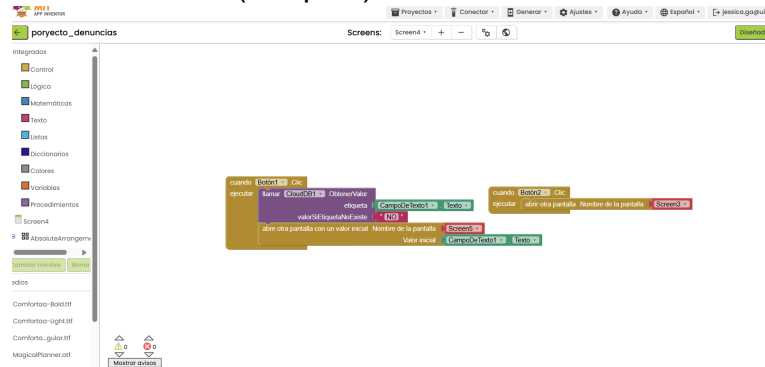
|        |          |                     |         |
|--------|----------|---------------------|---------|
| Botón1 | Presiona | Abrir otra pantalla | Screen5 |
| Botón2 | Presiona | Abrir otra pantalla | Screen5 |
| Botón3 | Presiona | Abrir otra pantalla | Screen4 |

- Interfaz de Inicio de sesión

The screenshot shows a mobile application interface for 'Seguridad Publica'. At the top, there's a status bar with the time '7:23 a. m.' and various icons. Below it, a blue header bar contains the app name 'Seguridad Publica' and a menu icon. A navigation bar below the header has three items: 'Denuncias Viales' (with a car icon), 'Inicio' (highlighted), and 'Denunciar Consultar'. The main content area features a large blue button labeled 'Iniciar Sesión' with a key icon. Below this are two input fields: one for email (containing 'correo@ejemplo.com') and one for password (labeled 'Contraseña'). There's a checkbox labeled 'Recordar sesión'. Another blue button labeled 'Iniciar Sesión' with a right arrow icon is positioned below the checkbox. At the bottom, there are two links: '¿No tiene cuenta? Registrarse' and '¿Olvidó su contraseña?'. The bottom of the screen shows a standard Android navigation bar.

En esta interfaz se puede iniciar sesión por medio de un correo electrónico y una contraseña, esta se despliega a partir del apartado “Denuncia Digital”, ya que para poder realizar esta se necesita de una cuenta registrada. En caso de no contar con una cuenta, hay un apartado que permite dirigir al usuario a otra interfaz para el registro de una nueva cuenta, y otro apartado que se tiene pensado que permita modificar la contraseña de la cuenta.

- Interfaz de Inicio de sesión (Bloques)



## Pantalla: Screen4 – Inicio de sesión o verificación de usuario

### Componentes involucrados

- **Botón1:** Inicia la verificación de usuario en CloudDB y navega a otra pantalla.
- **Botón2:** Navega directamente a la pantalla de registro (Screen3).
- **CloudDB1:** Base de datos en la nube, usada para comprobar si el usuario existe.
- **CampoDeTexto1:** Campo donde el usuario escribe su nombre de usuario o ID.

### Lógica del Bloque - Evento Botón1.Clic

Cuando el usuario hace clic en el **Botón1** (por ejemplo, un botón de "Iniciar sesión" o "Entrar"), se ejecutan dos acciones:

1. Se llama a **CloudDB1.ObtenerValor** para buscar si existe un usuario con la **etiqueta** igual al contenido del campo **CampoDeTexto1**.
  - Si no se encuentra el valor, devuelve "NO" como respuesta predeterminada.
2. Inmediatamente después, se abre otra pantalla:
  - **Pantalla destino:** **Screen5**
  - **Valor inicial:** Se le pasa el valor de **CampoDeTexto1.Texto**, es decir, el nombre de usuario, para que pueda ser utilizado en la siguiente pantalla (por ejemplo, para mostrar sus datos o permitirle hacer denuncias).

Nota: Aunque la llamada a **CloudDB1.ObtenerValor** se hace, en este bloque no se está utilizando el resultado directamente. Es probable que el propósito sea que en **Screen5** se use ese nombre de usuario para consultar los datos del usuario.

### Evento - Botón2.Clic

Cuando el usuario presiona el **Botón2** (posiblemente un botón que diga "Registrarse" o "Crear cuenta nueva"):

- Se ejecuta la acción **abrir otra pantalla**, específicamente:



- **Pantalla destino:** **Screen3** (pantalla de registro de usuario que analizamos en la imagen anterior).

Esto le permite al usuario acceder al formulario para registrarse si aún no tiene una cuenta.

## Funcionamiento General

Esta pantalla actúa como el **punto de entrada** o **pantalla de inicio de sesión** de la app. El flujo es el siguiente:

- El usuario escribe su nombre de usuario.
- Si desea iniciar sesión, pulsa **Botón1**, que obtiene el valor en CloudDB (aunque no lo usa directamente aquí) y lo manda a **Screen5**.
- Si no tiene cuenta, puede presionar **Botón2** y acceder al registro en **Screen3**.

### ● Interfaz de Registro de usuario

**Seguridad Publica**

**Registro de Usuario**

Complete el siguiente formulario para crear una cuenta. Los campos marcados con \* son obligatorios.

\* **Nombre(s)**

\* **Apellido(s)**

\* **Correo Electrónico**

Este correo se utilizará para iniciar sesión y recibir notificaciones.

\* **Contraseña**

\* **Confirmar Contraseña**

**Teléfono**

10 dígitos sin espacios ni guiones.

**CURP**

Opcional. Ayuda a validar tu identidad.

☐ \* Acepto los [términos y condiciones](#) y la [política de privacidad](#).

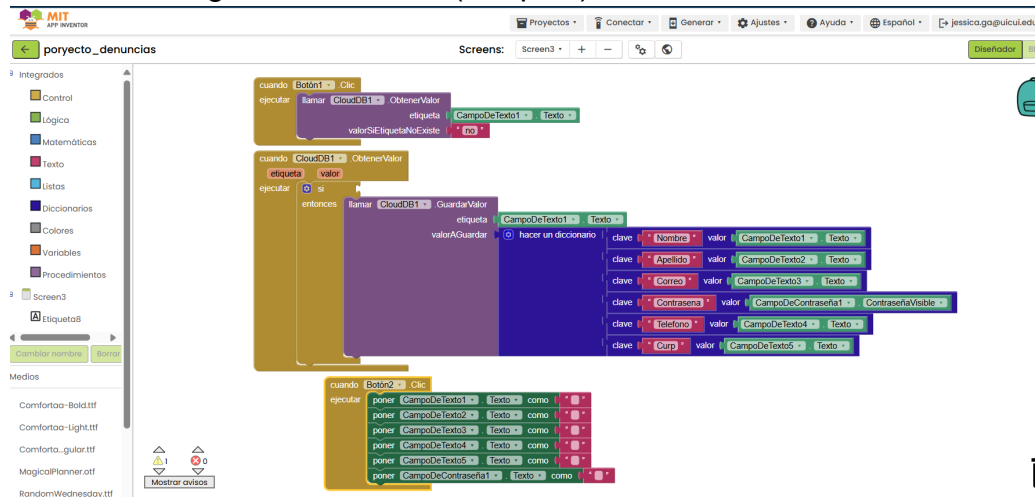
**Crear Cuenta**

**Limpiar Formulario**

¿Ya tiene una cuenta? [Iniciar Sesión](#)

Esta interfaz se despliega a partir de un apartado en la interfaz de Inicio de sesión, y permite el registro de una nueva cuenta, de la cual son necesarios datos tales como, nombre completo, correo electrónico, contraseña, teléfono y CURP, estos dos últimos son datos opcionales, además de que cada uno cuenta con su respectiva caja de texto. Cuenta con 2 botones, de los cuales, el primero crea la cuenta una vez que se han llenado los datos obligatorios, el segundo permite borrar los datos ingresados en todas las cajas de texto del formulario. Por último, cuenta con un apartado que redirige al usuario a la interfaz de Inicio de sesión en caso de ya contar con una cuenta.

- Interfaz de Registro de usuario (Bloques)



## Pantalla: Screen3 – Registro de usuario

### Componentes involucrados

- **Botón1:** Inicia la verificación y guardado del usuario en CloudDB.
- **Botón2:** Limpia los campos de texto del formulario.
- **CloudDB1:** Base de datos en la nube utilizada para almacenar los datos del usuario.
- **Campos de Texto:**
  - **CampoDeTexto1:** Nombre de usuario (clave primaria).
  - **CampoDeTexto2:** Apellido.
  - **CampoDeTexto3:** Correo electrónico.
  - **CampoDeTexto4:** Teléfono.
  - **CampoDeTexto5:** CURP.
  - **CampoDeContraseña1:** Contraseña.

### Lógica del Bloque - Evento Botón1.Clic

Cuando el usuario presiona el **Botón1** (por ejemplo, "Registrar"), ocurre lo siguiente:

1. **Obtener el valor del usuario en CloudDB1** usando como etiqueta el contenido de **CampoDeTexto1** (que se usa como identificador único del usuario).
  - Si la etiqueta no existe, retorna "no".

### Evento - CloudDB1.ObtenerValor

Este evento se activa automáticamente tras solicitar un valor desde la base de datos.

- **Condición si:** Verifica si el valor recibido es igual a "no", lo que indica que el usuario **no existe** aún en la base de datos.

#### Si el usuario no existe:

Se guarda un nuevo valor en CloudDB con:

- **Etiqueta:** El contenido de **CampoDeTexto1** (nombre de usuario).
- **Valor a guardar:** Un **diccionario** que contiene los siguientes pares clave-valor:
  - "Nombre": **CampoDeTexto1.Texto**
  - "Apellido": **CampoDeTexto2.Texto**
  - "Correo": **CampoDeTexto3.Texto**
  - "Contrasena": **CampoDeContrasena1.ContrasenaVisible**
  - "Telefono": **CampoDeTexto4.Texto**
  - "Curp": **CampoDeTexto5.Texto**

Esto almacena la información del usuario bajo una única etiqueta en la base de datos, organizando todos sus datos personales en forma de diccionario.

### Evento - Botón2.Clic

Cuando el usuario presiona **Botón2** (por ejemplo, "Limpiar" o "Cancelar"), se ejecuta la siguiente lógica:

- Todos los campos de texto (**CampoDeTexto1** a **CampoDeTexto5** y **CampoDeContrasena1**) son vaciados, asignándoles el valor "" (cadena vacía), dejando el formulario limpio para una nueva entrada.

### Funcionamiento General

Esta pantalla permite registrar usuarios nuevos en una base de datos en la nube:

- Verifica si ya existe un usuario con ese nombre.
- Si no existe, lo registra con todos sus datos organizados.
- Se puede limpiar el formulario para registrar a otro usuario o para corregir datos.

- Interfaz de Denuncia Digital.

7:24 a. m.

VOLTE 4G

44 %

Seguridad Publica

Denuncia Digital

Folio asignado automáticamente

DV202505280004

\* Nombre del Denunciante

Francisco

\* Teléfono

7227857187

\* Categoría del Incidente

Seguridad civil

\* Ubicación del Incidente

Obtener mi Ubicación Actual

Latitud

Longitud

19.572109

-99.754970

Dirección del incidente

Centro Universitario Ixtlahuaca (CUI), Calle Saitai

\* Fecha y Hora del Incidente

28/05/2025, 1:24 p.m.

7:25 a. m.

VOLTE 4G

44 %

Seguridad Publica

Dirección del incidente

Ej. Calle 123, Colonia, Ciudad

\* Fecha y Hora del Incidente

28/05/2025, 1:25 p.m.

Placa del Vehículo

Ej. ABC-123

\* Descripción del Incidente

Describe detalladamente lo ocurrido...

Evidencias (imágenes)

Elegir archivos Sin archivos seleccionados

Enviar Denuncia

Cerrar Sesión

¡Importante! Esta denuncia será registrada con tus datos. En caso de emergencia, llama directamente al 911.

Esta interfaz contiene los datos necesarios para realizar una denuncia digital, entre ellos se encuentran unos de carácter obligatorio tales como: nombre del denunciante, teléfono, categoría del incidente, ubicación del incidente (en este se hace uso de un sensor de ubicación), fecha y hora del Incidente, y descripción del incidente, y otros de carácter opcional, tales como, placa del vehículo (en caso de seleccionar en el apartado de categoría de incidente este se activa) y evidencias (en este apartado está un componente que permite subir un archivo desde el dispositivo). Cuenta además con dos botones los cuales tienen la funcionalidad de enviar denuncia y cerrar sesión.

**Folio asignado automáticamente**

DV202505280001

**\* Categoría del Incidente**

Seleccione una categoría

**\* Ubicación**

Obten

Selecciona un elemento de la lista

**Latitud**

**Longitud**

Además tiene la funcionalidad de que en caso de no llenar un campo obligatorio aparezca una alerta que indique que se deba llenar u seleccionar dicho campo obligatorio.

- Interfaz de Denuncia Anónima

7:23 a. m.
Seguridad Publica

**Formulario de Denuncia Vial**

Complete el siguiente formulario para reportar una infracción o incidente vial. Los campos marcados con \* son obligatorios.

**Folio asignado automáticamente**

DV202505280004

**\* Categoría del Incidente**

Seguridad civil

**\* Ubicación del Incidente**

Obtener mi Ubicación Actual

**Latitud**

19.572098

**Longitud**

-99.754984

**Dirección (calle, colonia, número)**

Centro Universitario Ixtlahuaca (CUI), Calle Saitama,

**\* Fecha y Hora del Incidente**

28/05/2025, 1:23 p.m.

7:23 a. m.
Seguridad Publica

**Latitud**

19.572098

**Longitud**

-99.754984

**Dirección (calle, colonia, número)**

Centro Universitario Ixtlahuaca (CUI), Calle Saitama,

**\* Fecha y Hora del Incidente**

28/05/2025, 1:23 p.m.

**Placa del Vehículo (solo si aplica)**

Ej. ABC-123

**\* Descripción del Incidente**

Describe detalladamente lo que observó...

**Evidencias**

Elegir archivos Sin archivos seleccionados

Limpiar Formulario

Enviar Denuncia

**¡Importante!** Este sistema es para denuncias que no requieren atención inmediata. En caso de emergencias o accidentes con heridos, llame al 911.

Este formulario de denuncia anónima forma parte de una aplicación móvil desarrollada en App Inventor para reportar incidentes sin revelar la identidad del

usuario. Su propósito es permitir a los ciudadanos informar sobre problemas de seguridad de manera rápida y confidencial.

El formulario genera un folio único para cada denuncia, asegurando su registro. El usuario puede seleccionar la categoría del incidente, especificar la ubicación, ya sea ingresando una dirección o activando el GPS, y registrar la fecha y hora del suceso.

También incluye campos para proporcionar detalles adicionales, como placas de vehículos involucrados y una descripción del incidente. Existe la opción de adjuntar imágenes o documentos como evidencia, y botones para enviar la denuncia o limpiar el formulario.

Además, se muestra un mensaje de advertencia, indicando que el sistema no es para emergencias y que, en casos graves, se recomienda llamar al 911.

## Interfaz: Formulario de Denuncia Anónima

 **Formulario de Denuncia Digitales**

Complete el siguiente formulario para reportar una infracción o incidente vial. Los campos marcados con \* son obligatorios.

**Folio asignado automáticamente**

ERROR

**\* Categoría del Incidente**

Seleccione una categoría

**\* Ubicación del Incidente**

Obtener mi Ubicación Actual

| Latitud | Longitud |
|---------|----------|
|         |          |

**Dirección (calle, colonia, número)**

Ej. Av. Insurgentes 123, Col. Roma

**\* Fecha y Hora del Incidente**

04/06/2025 03:39 a. m.

**Placa del Vehículo (solo si aplica)**

Ej. ABC-123

**\* Descripción del Incidente**

The screenshot shows a web form titled "Denuncias Digitales". It contains several sections: "Ubicación del Incidente" with a button to get the current location and fields for latitude and longitude; "Dirección (calle, colonia, número)" with a text input; "Fecha y Hora del Incidente" with a date and time picker; "Placa del Vehículo (solo si aplica)" with a text input; "Descripción del Incidente" with a large text area; "Evidencias" with a file upload button; and "Enviar Denuncia" with a blue button. A yellow warning box at the bottom states that the system is for non-emergency reports and that 911 should be called for emergencies.

\* Ubicación del Incidente

Obtener mi Ubicación Actual

Latitud

Longitud

Dirección (calle, colonia, número)

Ej. Av. Insurgentes 123, Col. Roma

\* Fecha y Hora del Incidente

04/06/2025 03:39 a. m.

Placa del Vehículo (solo si aplica)

Ej. ABC-123

\* Descripción del Incidente

Describe detalladamente lo que observó...

Evidencias

Elegir archivos Sin archivos seleccionados

Limpiar Formulario

Enviar Denuncia

**¡Importante!** Este sistema es para denuncias que no requieren atención inmediata. En caso de emergencias o accidentes con heridos, llame al 911.

## Propósito general

Esta interfaz web permite a los ciudadanos realizar denuncias anónimas sobre distintos tipos de incidentes relacionados con la seguridad (vial, civil, infantil, médica, policiaca y cibernética). Está diseñada para recolectar información precisa del incidente, incluyendo ubicación, fecha, descripción, y evidencia multimedia.

## Estructura y funcionamiento

### 1. Encabezado

- Título: " Denuncias Digitales"
- Navegación: Enlaces a las secciones "Inicio" (/) y "Consultar" (/consulta).

### 2. Contenedor principal (.container)

- Estilizado con un diseño centrado, limpio y responsivo.
- Muestra una introducción breve sobre el propósito del formulario.

## Formulario de denuncia

El formulario se envía mediante método **POST** a la ruta **/api/anonima**, aceptando archivos (**multipart/form-data**).

### Campos del formulario:

| Campo              | Tipo             | Descripción   | Obligatorio       |
|--------------------|------------------|---|-------------------|
| folio              | Texto (readonly) | Se llena automáticamente con un número de folio obtenido desde <code>/api/ultimo_folio</code> . | Sí (autoasignado) |
| categoria          | Select           | Categoría del incidente. Al seleccionar "Seguridad vial", se activa el campo de placa.          |                   |
| latitud / longitud | Texto            | Coordenadas obtenidas con geolocalización.  | Opcional          |
| direccion          | Texto            | Dirección aproximada usando OpenStreetMap (Nominatim).  | Opcional          |
| fecha_hora         | datetime-local   | Fecha y hora del incidente. Se llena automáticamente con la hora actual.                        |                   |
| placa_vehiculo     | Texto            | Solo se activa si la categoría es "Seguridad vial".   | Opcional          |
| descripcion        | Textarea         | Descripción libre y detallada del incidente.  |                   |
| evidencias         | Archivo          | Permite subir imágenes, videos o documentos.  | Opcional          |

### Botones:

- **" Obtener mi Ubicación Actual"**: Usa la API de geolocalización para llenar los campos de latitud, longitud y dirección.
- **"Limpiar Formulario"**: Resetea todos los campos del formulario.
- **" Enviar Denuncia"**: Envía la denuncia.

### Funcionalidad del Script (JavaScript)

1. **Al cargar la página:**
  - Se obtiene el último folio disponible desde `/api/ultimo_folio`.
  - Se autocompleta la fecha y hora del sistema.
2. **Ubicación del usuario:**
  - Al hacer clic en el botón de ubicación, se obtiene la posición GPS del usuario (latitud y longitud).
  - Con esta información, se consulta la API de OpenStreetMap (Nominatim) para convertirla en una dirección legible.
3. **Validación dinámica de campos:**
  - Si el usuario selecciona "Seguridad vial", se habilita el campo de "Placa del vehículo".



- Si se cambia la categoría, el campo se desactiva y limpia automáticamente.

## Nota informativa

Al final del formulario se incluye un mensaje de advertencia:

Este sistema es para denuncias que no requieren atención inmediata. En caso de emergencias o accidentes con heridos, llame al 911.

## Estilo visual (CSS)

- Colores suaves (fondo blanco, encabezados azules).
- Diseño responsivo y amigable.
- Sombras, bordes redondeados y espaciado adecuado para mejorar la experiencia del usuario.

## Resumen

La interfaz de "**Denuncia Anónima**" está orientada a facilitar la recolección de reportes ciudadanos sobre incidentes sin necesidad de identificar al denunciante. Es interactiva, accesible, y aprovecha tecnologías del navegador como la geolocalización para mejorar la precisión del reporte.

## Interfaz: Consulta de Denuncias Digitales

Esta interfaz permite a los usuarios **consultar el estado actual de una denuncia previamente realizada**, utilizando el número de folio asignado al momento de su registro.

## Componentes principales de la interfaz

### 1. Encabezado (Header)

- En el cual se muestra el título “ Denuncias Digitales”, que además incluye navegación hacia:
  - **Inicio**
  - **Formulario de denuncia anónima**

## 2. Panel principal: Consulta de denuncias

- **Campo de entrada de folio** (`<input id="folio">`): permite al usuario ingresar el folio de la denuncia.
- **Botón "Consultar"**: al hacer clic, ejecuta una función JavaScript que consulta el backend mediante una API (`/api/consulta/{folio}`).
- **Resultado de la consulta** (oculto por defecto, visible tras una búsqueda exitosa):
  - Estado de la denuncia con una **etiqueta de color (badge)** según el estado:
    - **Recibida**: gris
    - **Rechazada**: rojo
    - **Procesada**: verde
    - **En revisión**: naranja
    - **Validada**: azul
  - **Descripción del seguimiento**: texto explicativo sobre el estado actual.
  - **Datos del personal** que atendió la denuncia:
    - Nombre
    - Cargo

## 3. Panel lateral: Preguntas Frecuentes (FAQ)

- Ofrece respuestas rápidas mediante elementos `<details>` a dudas comunes como:
  - Cómo obtener el folio
  - Qué hacer si se pierde
  - Tiempo de procesamiento
  - Posibilidad de agregar información adicional

## 4. Panel de Contacto

- Brinda **medios de comunicación para soporte**:
  - Correo electrónico
  - Teléfono
  - Horario de atención


## Lógica de funcionalidad (JavaScript)

- Valida que se haya ingresado un folio.
- Realiza una **consulta asincrónica a la API** usando `fetch`.
- Muestra mensajes de error si no se encuentra la denuncia.
- Actualiza dinámicamente los elementos del DOM con la información obtenida.

## Objetivo de la interfaz

Brindar a los ciudadanos una herramienta simple, accesible y segura para dar **seguimiento al estado de sus denuncias**, sin necesidad de registrarse, fomentando así la **transparencia y confianza en el proceso** de atención.

## Interfaz: Formulario de Denuncia Digital

 Denuncia Digital

Folio asignado automáticamente

ERROR

\* Nombre del Denunciante

{{ session.get('nombre', '') }}

\* Teléfono

{{ session.get('telefono', '') }}

\* Categoría del Incidente

Seleccione una categoría

\* Ubicación del Incidente

📍 Obtener mi Ubicación Actual


Latitud

Longitud

Dirección del incidente

Ej. Calle 123, Colonia, Ciudad

\* Fecha y Hora del Incidente

04/06/2025 04:01 a. m. 

Placa del Vehículo

Placa del Vehículo

Ej. ABC-123


\* Descripción del Incidente

Describe detalladamente lo ocurrido...

Evidencias (imágenes)

Elegir archivos

No se eligió ningún archivo

 Enviar Denuncia

¡Importante!

 Esta denuncia será registrada con tus datos. En caso de emergencia, llama directamente al 911.

Esta interfaz presenta un formulario moderno y centrado para registrar denuncias de manera digital. Su diseño limpio y responsivo ofrece una experiencia clara y guiada para el usuario. A continuación, se describen sus principales componentes y funcionalidades:

## Encabezado

La parte superior de la página cuenta con un **header de color azul** que muestra el título "**Denuncias Digitales**" y un enlace a la página de inicio. Su propósito es mantener la identidad visual y facilitar la navegación.

## Contenedor principal

El formulario se encuentra dentro de un **contenedor blanco con sombra y bordes redondeados**, centrado en la pantalla. Este contenedor agrupa todos los elementos y los presenta de manera organizada, comenzando con un título azul que indica que es una denuncia digital.

### Datos del denunciante

El usuario debe ingresar su **nombre y número telefónico**, los cuales se pueden autocompletar si la sesión ya tiene datos precargados. Estos campos son obligatorios y, si se detecta una sesión activa, se vuelven de solo lectura para evitar modificaciones.

### Categoría del incidente

Se ofrece un menú desplegable para seleccionar la categoría de la denuncia: seguridad vial, civil, infantil, médica, policiaca o ciberseguridad. Al seleccionar **"Seguridad vial"**, se habilita dinámicamente el campo para ingresar la **placa del vehículo involucrado**, mientras que en otras categorías este campo permanece deshabilitado.

### Ubicación del incidente

Un botón permite obtener la ubicación actual del usuario utilizando la **API de geolocalización del navegador**. Al hacer clic, se capturan automáticamente la **latitud, longitud y dirección** del lugar, que se llenan en los campos correspondientes. Esta funcionalidad facilita que los reportes se registren con datos geográficos precisos.

### Fecha y hora

El formulario incluye un campo de tipo **datetime-local** que se autocompleta con la fecha y hora actuales al cargar la página. Esto permite registrar con exactitud el momento del incidente.

### Descripción y evidencia

El denunciante debe ingresar una **descripción detallada** del incidente en un área de texto amplia. También puede **adjuntar múltiples archivos**, como imágenes, videos o documentos, que sirvan como evidencia.

### Envío y seguimiento

Al enviar el formulario, los datos se procesan mediante una petición **fetch** a una API interna (**/api/digital**). Si la denuncia se envía correctamente:

- Se muestra una alerta de confirmación.
- Se reinicia el formulario (excepto los datos de sesión).
- Se genera un **nuevo folio automático**.
- Se actualiza la fecha y hora.
- Se ocultan los botones innecesarios y se **habilita el botón "Cerrar sesión"**.
- Se deshabilitan todos los campos para prevenir modificaciones posteriores.

## Alerta informativa

Al final del formulario, se muestra una **advertencia destacada** en color amarillo que recuerda al usuario que, en caso de emergencia, debe comunicarse directamente al **911**.

## Aplicación Web

Estas son las herramientas clave que se utilizaron para desarrollar una página web funcional y segura, combinando tanto el backend (la parte lógica y el servidor) como el frontend (lo que ven los usuarios).

- Core y Utilidades
  - Blinker:  
Proporciona un sistema de señales (eventos) para enviar notificaciones entre componentes en una aplicación Python.
  - click:  
Biblioteca para crear interfaces de línea de comandos (CLI) elegantes y configurables.
  - colorama:  
Permite imprimir texto en colores en la terminal (Windows, Linux, macOS).
  - itsdangerous:  
Ofrece seguridad para firmar datos (como tokens o cookies) para prevenir su manipulación.
  - packaging:  
Utilidades para manejar versiones de paquetes y dependencias (usado internamente por pip/setuptools).
  - python-dotenv  
Carga variables de entorno desde un archivo .env para configurar aplicaciones fácilmente.
  - typing\_extensions  
Proporciona tipos adicionales para el sistema de tipado de Python (compatibilidad con versiones antiguas).
- Flask y Extensiones
  - Flask:  
Microframework web ligero para construir aplicaciones web en Python.
  - flask-cors  
Maneja CORS (Cross-Origin Resource Sharing) para permitir peticiones entre dominios en APIs Flask.
  - Flask-JWT-Extended:  
Añade soporte para JSON Web Tokens (JWT) en Flask (autenticación y autorización).
  - Flask-Login:  
Gestiona sesiones de usuarios y autenticación en aplicaciones Flask.
  - Flask-SQLAlchemy:  
Integración de SQLAlchemy (ORM para bases de datos) con Flask.
  - Jinja2:  
Motor de plantillas para renderizar HTML dinámico en Flask.
  - MarkupSafe:

- Escapa caracteres especiales en cadenas HTML/XML (usado por Jinja2).
- Werkzeug:  
Biblioteca WSGI (base de Flask) para manejar peticiones HTTP, routing, etc.
  - Bases de Datos
    - SQLAlchemy:  
ORM (Mapeo Objeto-Relacional) para interactuar con bases de datos SQL.
    - greenlet:  
Implementa corrutinas ligeras para concurrencia (usado por SQLAlchemy para operaciones async).
    - mysql-connector-python:  
Controlador oficial de MySQL para Python.
    - PyMySQL:  
Cliente MySQL puro en Python (alternativa a mysql-connector).
    - pytz  
Manejo de zonas horarias (usado en aplicaciones con fechas/horas).
  - Otros
    - PyJWT:  
Implementación de JSON Web Tokens (JWT) para autenticación.
    - pyngrok:  
Integración con ngrok para exponer servidores locales a internet (útil para desarrollo/debug).
    - PyYAML:  
Analiza y genera YAML (formato de serialización de datos).
    - gunicorn  
Servidor WSGI HTTP para producción (ejecuta aplicaciones Flask en entornos reales).

## Interfaz: Página Principal del Sistema de Denuncias



Esta interfaz representa la pantalla de bienvenida al sistema de denuncias digitales. Tiene un diseño visualmente limpio, centrado en la experiencia del usuario y con navegación rápida hacia las diferentes modalidades de denuncia. A continuación se detallan sus componentes clave:

## Encabezado de navegación

La parte superior muestra un **header azul** con el título “ **Denuncias Digitales**” en negritas. A su derecha, una barra de navegación con enlaces rápidos permite acceder a:

- **Denunciar:** Redirige a la sección de denuncias anónimas.
- **Consultar:** Lleva al usuario a una página para revisar el estado de denuncias.

## Introducción centrada

El contenido principal está centrado verticalmente, con:

- Un **título grande** (“Sistema de Denuncias”) en color azul.
- Un **subtítulo explicativo** que invita a los ciudadanos a participar reportando incidentes para mejorar la seguridad comunitaria.

## Selector de modalidad

Se muestra un grupo de botones (**.btn-group**) grandes y vistosos con bordes redondeados, diseñados para facilitar la elección de modalidad de denuncia:

- **Denuncia Anónima:** Redirige al formulario para denunciar sin registrar datos personales.
- **Denuncia Civil:** Lleva a la sección donde ciudadanos registrados pueden levantar una denuncia formal, con seguimiento en línea.

Los botones tienen un efecto animado al pasar el cursor, lo que mejora la interacción visual.

## Botón de consulta

Debajo de los botones principales se encuentra un **botón azul con borde redondeado** que permite a los usuarios **consultar el estado de una denuncia previamente registrada**. Este botón tiene un estilo prominente para captar la atención del usuario.

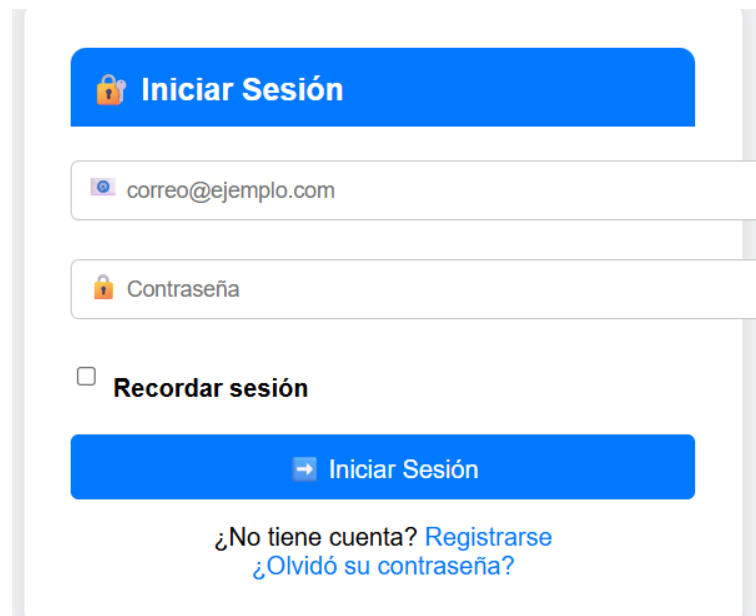
## Tarjetas informativas

En la parte inferior de la interfaz se encuentran **dos tarjetas (cards)** que explican brevemente cada tipo de denuncia:

- **Denuncia Anónima**
  - Fondo blanco con una franja lateral verde.
  - Explica que no requiere datos personales.
  - Menciona un proceso simple y la posibilidad de adjuntar evidencia.
- **Denuncia Civil**
  - Fondo blanco con franja azul turquesa.
  - Indica que está disponible 24/7.
  - Incluye seguimiento en tiempo real y contacto con autoridades.

Ambas tarjetas tienen **bordes redondeados**, **sombra suave** y están alineadas horizontalmente (o en columnas si hay poco espacio), lo que las hace amigables y accesibles.

## Interfaz: Pantalla de Inicio de Sesión del Sistema de Denuncias



**Iniciar Sesión**

correo@ejemplo.com

Contraseña

☐ Recordar sesión

**Iniciar Sesión**

[¿No tiene cuenta? Registrarse](#)  
[¿Olvidó su contraseña?](#)

Esta interfaz permite a los usuarios registrados acceder al sistema mediante sus credenciales. Presenta un diseño centrado, limpio y claro, optimizado para una experiencia sencilla y directa. A continuación se describen sus componentes:

### Encabezado

En la parte superior se encuentra una **barra azul (header)** con el título “ **Denuncias Digitales**”. A la derecha, un enlace llamado “**Inicio**” redirige a la página principal del sistema. Este diseño mantiene la coherencia visual con otras pantallas del sistema.

### Contenedor central del formulario

El elemento principal de la interfaz es una **tarjeta blanca con bordes redondeados y sombra suave**, ubicada al centro de la pantalla. Este contenedor alberga el formulario de inicio de sesión. Está estructurado de forma vertical y presenta un diseño intuitivo:

- **Título del formulario:** Un encabezado azul con texto blanco que dice “ **Iniciar Sesión**”, resaltando el propósito de la página.

### Campos del formulario

El formulario se compone de los siguientes campos, estilizados con iconos a la izquierda y bordes redondeados:



- **Correo electrónico**  
Un campo de entrada de tipo **email** con un placeholder “correo@ejemplo.com”. Está precedido por un ícono decorativo de sobre .
- **Contraseña**  
Campo de entrada de tipo **password** con el placeholder “Contraseña”, acompañado de un ícono de candado

Ambos campos están insertados en **form-group** que añaden espaciado, padding especial e íconos fijos para mejorar la experiencia visual del usuario.

### **Opción de recordar sesión**

Debajo de los campos, se encuentra una casilla de verificación acompañada de la etiqueta **“Recordar sesión”**, que permite mantener la sesión iniciada en dispositivos de confianza.

### **Botón de acción**

Un botón azul con ícono de flecha (-->) y texto **“Iniciar Sesión”** ocupa todo el ancho del contenedor. Este botón es claramente visible y está optimizado para el uso en dispositivos móviles y de escritorio.

### **Acciones adicionales**

Al pie del formulario, dentro de un bloque **.extra** centrado, se ofrecen enlaces adicionales para los usuarios que aún no están registrados o que han olvidado su contraseña:


- **“Registrarse”** enlaza a la página de creación de cuenta.
- **“¿Olvidó su contraseña?”** es un enlace de ayuda aún sin destino definido (#).

## Interfaz: Pantalla de Registro de Usuario del Sistema de Denuncias

**CURP**


Opcional. Ayuda a validar tu identidad.

☐ \* Acepto los [términos y condiciones](#) y la [política de privacidad](#).

 Crear Cuenta

☐ Limpiar Formulario

¿Ya tiene una cuenta? [Iniciar Sesión](#)

 **Registro de Usuario**

Complete el siguiente formulario para crear una cuenta. Los campos marcados con \* son obligatorios.

**\* Nombre(s)**

**\* Apellido(s)**

**\* Correo Electrónico**

Este correo se utilizará para iniciar sesión y recibir notificaciones.

**\* Contraseña**

**\* Confirmar Contraseña**

**Teléfono**

10 dígitos sin espacios ni guiones.


**CURP**

Opcional. Ayuda a validar tu identidad.

☐ \* Acepto los [términos y condiciones](#) y la [política de privacidad](#).

Esta interfaz está diseñada para que nuevos usuarios creen una cuenta en el sistema de **Denuncias Digitales**. El diseño es moderno, claro y funcional, con un enfoque en la accesibilidad, seguridad y facilidad de uso.

### Encabezado

La parte superior contiene una **barra azul (header)** con el título “ **Denuncias Digitales**” a la izquierda. A la derecha, un enlace de navegación “**Inicio**” permite regresar a la página principal del sistema. El diseño mantiene la identidad visual del sitio.

## Contenedor central del formulario

El cuerpo principal de la página es una **tarjeta blanca** centrada, con sombra suave y bordes redondeados, que da al usuario una sensación de enfoque y estructura.

- En la parte superior del contenedor aparece el título “ **Registro de Usuario**” en un recuadro azul con letras blancas.
- Debajo, un pequeño texto guía indica que los campos con asterisco (\*) son obligatorios.

## Formulario de registro

El formulario contiene campos cuidadosamente organizados para recopilar información esencial del nuevo usuario. Cada campo tiene su etiqueta correspondiente, algunos con ayudas contextuales o validaciones específicas:

- **Nombre(s) y Apellido(s)** (*text*)  
Campos obligatorios para identificar al usuario.
- **Correo Electrónico** (*email*)  
Obligatorio. Incluye un mensaje auxiliar que aclara su uso para notificaciones y acceso.
- **Contraseña y Confirmar Contraseña** (*password*)  
Campos protegidos y obligatorios para establecer la seguridad de la cuenta.
- **Teléfono** (*text*)  
Campo opcional con una restricción de formato (10 dígitos sin espacios ni guiones).
- **CURP** (*text*)  
Campo opcional con función de verificación de identidad.

## Aceptación de términos

Antes de finalizar el registro, el usuario debe aceptar una casilla obligatoria para los **términos y condiciones** y la **política de privacidad**, con enlaces incluidos.

## Botones de acción

Dos botones están disponibles al final del formulario:

- **Crear Cuenta:** Envía el formulario para registrar al usuario.
- **Limpiar Formulario:** Restablece todos los campos a sus valores iniciales.

Ambos botones mantienen coherencia con el diseño general: bordes redondeados y colores bien definidos (azul para acción principal, gris claro para secundario).

## Enlace para usuarios existentes

Al final del formulario, un mensaje amigable invita a los usuarios que ya tienen cuenta a “**Iniciar Sesión**”, con un enlace directo.

## Descripción de la Clase Control (*app.py*) – Proyecto *Denuncias Digitales*

Este archivo actúa como el **controlador principal** del sistema web, coordinando las vistas (interfaces HTML), la lógica del negocio (registro, login, denuncias) y el acceso a la base de datos (SQLite). Está construido con **Flask**, y permite gestionar formularios, sesiones, archivos multimedia y peticiones a través de distintas rutas.

## 1. Inicialización y Configuración

- Se importa Flask y módulos necesarios para formularios, sesiones, JSON y plantillas.
- Se establece la conexión con la base de datos SQLite a través de una ruta fija.
- Se define una clave secreta para usar sesiones de usuario.

```
app = Flask(__name__)
DB_PATH = os.path.join(os.path.dirname(__file__), "denuncias.db")
app.secret_key = "clave-secreta-segura"
```

## 2. Rutas de Interfaz (HTML)

Estas funciones sirven páginas estáticas al usuario.

- `/` → Página principal ([index.html](#))
- `/anonimo` → Formulario de denuncia anónima.
- `/digital` → Formulario de denuncia digital (requiere login).
- `/consulta` → Formulario para consultar folios de denuncia.
- `/registro` → Página de registro de nuevos usuarios.
- `/login` → Página de inicio de sesión.

## 3. Generación de Folios

- `/api/ultimo_folio`  
Calcula y devuelve el siguiente **folio único** en formato:  
`DV + fecha + número incremental`  
Ejemplo: `DV202506030001`

## 4. Registro de Usuario

- `/api/registro` (POST)  
Recibe los datos del formulario de registro.
  - Inserta nombre, apellidos, correo, contraseña, teléfono, CURP y aceptación de términos en la tabla `usuarios`.
  - Si todo es correcto, redirige al login.
  - Maneja errores con `try-except`.

## 5. Inicio de Sesión (Login)

- `/api/login` (POST)  
Verifica las credenciales (correo y contraseña) contra la base de datos.
  - Si son correctas, guarda `nombre` y `telefono` del usuario en la sesión.
  - Redirige al formulario digital.
  - Si no coinciden, muestra error de autenticación.

## 6. Cierre de Sesión

- `/logout`  
Limpia la sesión y redirige al inicio.

## 7. Envío de Denuncia Anónima

- `/api/anonima` (POST)  
Procesa el formulario anónimo:
  - Genera folio automáticamente.
  - Guarda los datos de la denuncia en la tabla `denuncias`.
  - Guarda los archivos adjuntos (imágenes, videos, etc.) en `static/uploads`.
  - Asocia las evidencias con el ID de denuncia en la tabla `evidencias`.

## 8. Envío de Denuncia Digital

- `/api/digital` (POST)  
Requiere que el usuario esté autenticado:
  - Verifica nombre y teléfono para obtener el ID del usuario.
  - Guarda la denuncia en la tabla `denuncias` con tipo "digital".
  - Almacena archivos de evidencia en `static/uploads`.
  - Asocia las evidencias a la denuncia.

## 9. Consulta de Denuncia por Folio

- `/api/consulta/<folio>`  
Permite consultar el **estado**, seguimiento, y quién atiende la denuncia mediante el número de folio.
  - Si existe, retorna un JSON con los detalles.
  - Si no existe, responde con error 404.

## 10. Ejecución Local

Al final, la aplicación corre en modo debug:

```
if __name__ == "__main__":  
    app.run(debug=True)
```

## Observaciones Generales

- El código sigue el patrón Modelo-Vista-Controlador (MVC), donde esta clase representa el **Controlador**.
- Las vistas están en HTML renderizado por `render_template`.
- Las operaciones se comunican con la base de datos SQLite.
- Usa sesiones para mantener el estado del usuario logueado.
- Maneja evidencias multimedia con subida de archivos a una carpeta local.

## Link y código QR para descarga



<https://denuncias-ni81.onrender.com/>

## Conclusión

El desarrollo de esta aplicación web y móvil representa una solución integral para mejorar la interacción entre los ciudadanos de Ixtlahuaca y las autoridades locales en materia de denuncias ciudadanas. Al ofrecer una plataforma accesible desde múltiples dispositivos, se eliminan barreras geográficas y operativas, permitiendo que los usuarios reporten problemas de seguridad, irregularidades o situaciones delicadas de manera rápida, segura y de ser necesario, anónima. La versión móvil garantiza la portabilidad y el acceso inmediato desde cualquier lugar, mientras que la interfaz web asegura compatibilidad con diferentes sistemas y una gestión más amplia desde computadoras.

Además, el sistema de seguimiento en tiempo real incrementa la transparencia y confianza en el proceso, al permitir que los ciudadanos verifiquen el estado de sus reportes. La implementación de metodologías ágiles como SCRUM aseguró un desarrollo eficiente, centrado en las necesidades reales de los usuarios y adaptable a futuras mejoras.

En conjunto, esta herramienta no solo optimiza la gestión de denuncias, sino que también fomenta la participación ciudadana, promueve la rendición de cuentas y contribuye a construir un municipio más seguro y transparente. Su escalabilidad la convierte en un modelo replicable para otras localidades que enfrentan desafíos similares, demostrando el potencial de la tecnología como motor de cambio social.

# Bibliografía en formato APA

Universidad de Ixtlahuaca. (2025). Sistema de Denuncias Ciudadanas: Proyecto de aplicación móvil y web. <https://denuncias-ni81.onrender.com/>

App Inventor. (s.f.). MIT App Inventor. <https://appinventor.mit.edu/>

OpenStreetMap Foundation. (s.f.). Nominatim – Geocodificación y geolocalización de direcciones. <https://nominatim.openstreetmap.org/>

Leaflet. (s.f.). Leaflet: An open-source JavaScript library for interactive maps. <https://leafletjs.com/>

Flask. (s.f.). Flask Documentation. <https://flask.palletsprojects.com/>

onacher, A. (2024). Flask: Web Development One Drop at a Time (v3.0). Pallets Project. <https://flask.palletsprojects.com/>

SQLAlchemy. (2024). SQLAlchemy ORM Documentation. <https://docs.sqlalchemy.org/>

MySQL. (2024). MySQL Connector for Python Developer Guide. Oracle. <https://dev.mysql.com/doc/connector-python/en/>

PyJWT. (2024). JSON Web Token implementation in Python. <https://pyjwt.readthedocs.io/>

Pyngrok. (2024). Expose your local server to the internet with ngrok. <https://pyngrok.readthedocs.io/>

11. Python Software Foundation. (2024). python-dotenv: Reads key-value pairs from .env file and adds them to environment variables. <https://pypi.org/project/python-dotenv/>

Flask-CORS. (2024). A Flask extension for handling Cross Origin Resource Sharing (CORS). <https://flask-cors.readthedocs.io/>

Flask-JWT-Extended. (2024). Secure JWT Authentication for Flask APIs. <https://flask-jwt-extended.readthedocs.io/>