

## Especificación de Requisitos según el estándar de IEEE 830

IEEE Std. 830-1998

30 de Enero de 2025

### **Resumen**

Este documento presenta, en español, el formato de Especificación de Requisitos de Software (ERS) según la última versión del estándar IEEE 830. Según el IEEE, un buen documento de requisitos, aunque no está obligado a seguir estrictamente la organización y el formato establecidos en el estándar 830, sí debería incluir, de una forma u otra, toda la información que este presenta. El estándar IEEE 830 no está libre de defectos ni de sesgos y, por ello, ha sido criticado por múltiples autores desde diversas perspectivas, hasta el punto de cuestionarse si realmente puede considerarse un estándar en el sentido habitual del término en otras ramas de la ingeniería. El presente documento no pretende posicionarse ni a favor ni en contra de estas críticas, sino únicamente reproducir, con fines principalmente educativos, cómo se organizaría un documento de requisitos según el estándar IEEE 830

**1. Introduccion 3**

- 1.1. Propósito 3
- 1.2. Ámbito del Sistema 3
- 1.3. Definiciones, Acrónimos y Abreviaturas 3
- 1.4. Referencias 3
- 1.5. Visión General del Documento 4

**2. Descripcion General 4**

- 2.1. Perspectiva del Producto4
- 2.2. Funciones del Producto 4
- 2.3. Características de los Usuarios 5
- 2.4. Restricciones 5
- 2.5. Suposiciones y Dependencias 5
- 2.6. Requisitos Futuros 6

**3. Requisitos Especificos 6**

- 3.1. Interfaces Externas 7
- 3.2. Funciones 7
- 3.3. Requisitos de Rendimiento 9
- 3.4. Restricciones de Diseño9
- 3.5. Atributos del Sistema 9
- 3.6. Otros Requisitos 9

**4. Apendices 9**

### **1. Introducción**

En este documento se van a hacer las especificaciones de software del proyecto desarrollado por Ricardo Reyes y Compañeros (Ian Santoyo, Alvaro Zermeño, Chris Lopez) 'Atenea' para la empresa 'Olimp Tec'

#### **1.1. Propósito**

Este documento, a quien corresponda, contiene los requerimientos del software orientado y dedicado a la protección de la mujer aguascalentense y ciudadanía en general. Estro contiene los requerimientos mínimos para el desarrollo, procesamiento y ejecutable de este proyecto.

#### **1.2. Ámbito del Sistema**

En esta subsección:

- Nombre elegido por el equipo de trabajo: 'Atenea'.
- Se hará descripción de detalles necesarios, procesos y diseños de la aplicación deseados para su posterior desarrollo.
- Se busca un beneficio en la seguridad de la ciudadanía en base a sus dispositivos móviles, una baja tasa de incidentes en los crímenes de género y una mejor seguridad en la ciudadanía en general.

#### **1.3. Definiciones, Acrónimos y Abreviaturas**

En esta subsección se definirán todos los términos, acrónimos y abreviaturas utilizadas en la ERS.

#### **1.4. Referencias**

En esta subsección se mostrará una lista completa de todos los documentos referenciados en la ERS.

### 1.5. Visión General del Documento

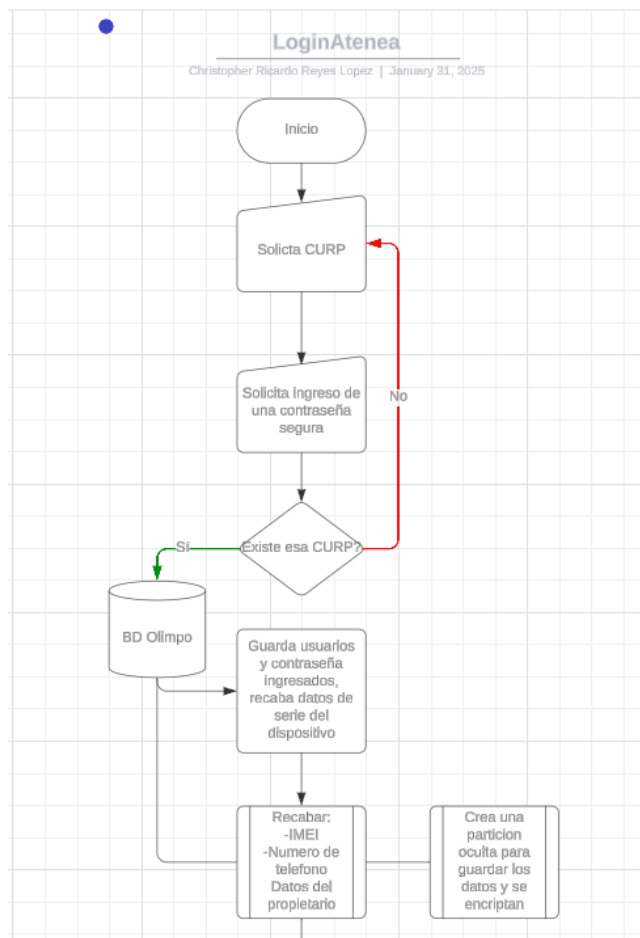
Se busca en este documento, dar la descripción general de los requerimientos, funciones y diseños de la aplicación deseados que están sujetos a edición y alteración permitida por los encargados del desarrollo y los mismos desarrolladores.

## 2. Descripción General

En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. En el contexto de el rastreo de personas en situación de crisis, víctimas de crimen y personas en potencial crimen, se hace uso de las tecnologías de la Información y la tecnología que pueden ser útiles para la prevención de estas incidencias. Utilizando técnicas y códigos orientados para el apoyo a la ciudadanía de manera ética.

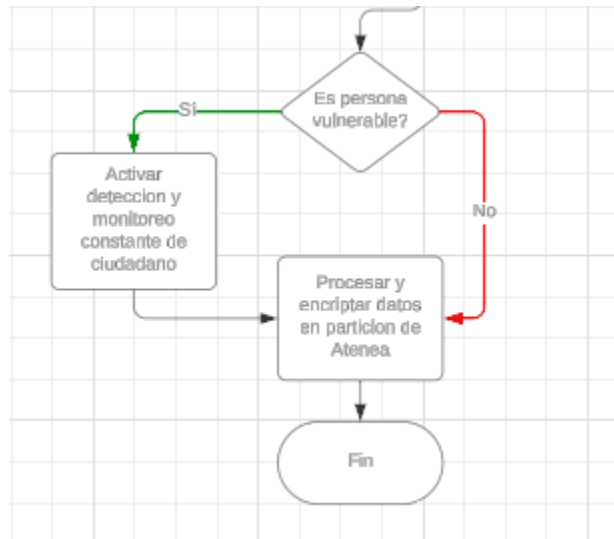
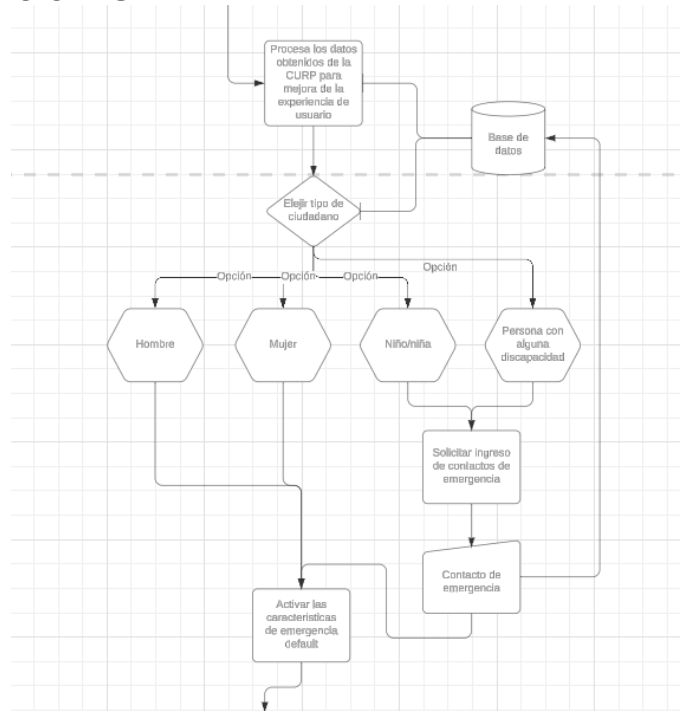
### 2.1. Perspectiva del Producto

Diagrama de Login



## 2 DESCRIPCIÓN GENERAL

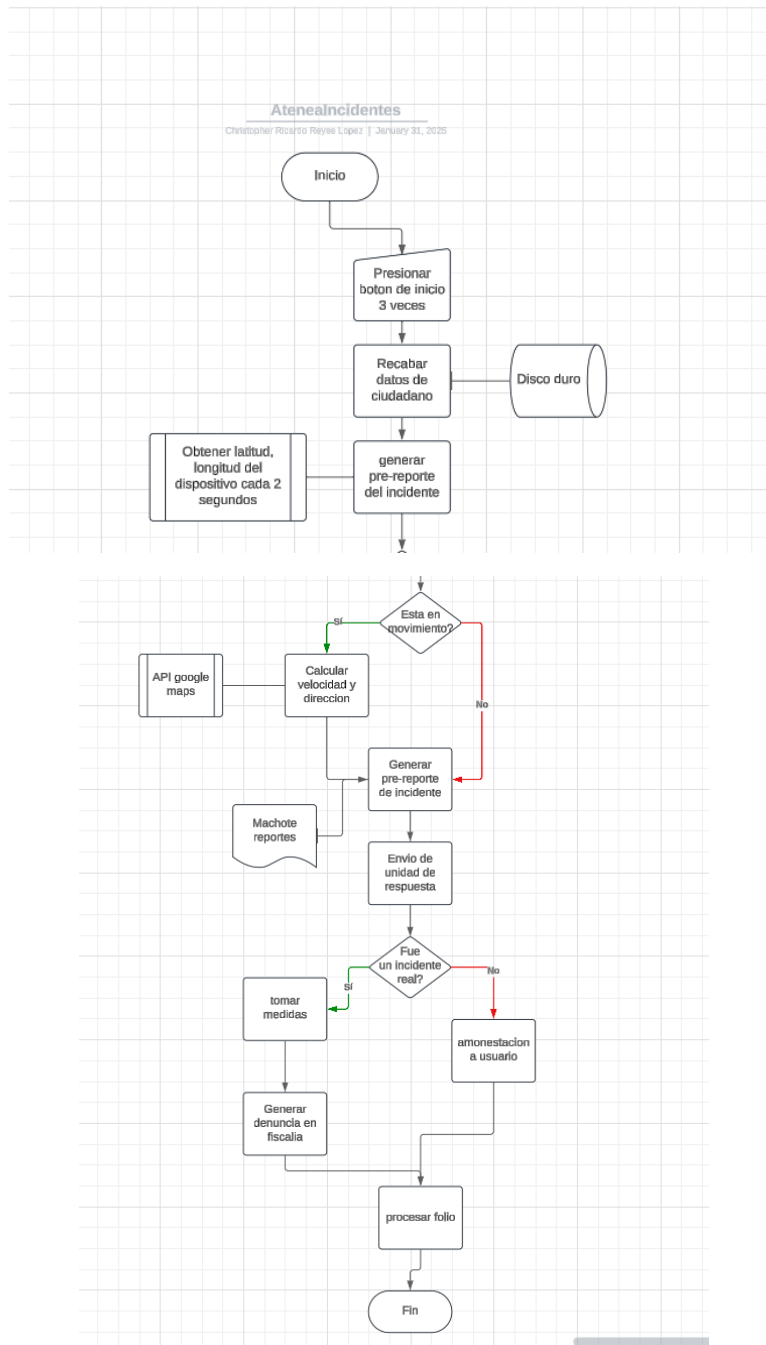
5



## 2 DESCRIPCIÓN GENERAL

6

### Diagrama Localización



## **2 DESCRIPCIÓN GENERAL**

7

### **2.2. Funciones del Producto**

Se buscan tener las siguientes funcionalidades:

1. Rastreo: Se busca que tenga un acceso inicial a los datos y localización de la ubicación del usuario y poder determinar su ubicación en caso de ser solicitada por el usuario en caso de un incidente.
2. Login: inicio de sesión sencillo, se solicita usuario (CURP) y contraseña que será protegida bajo un hasheo AES-256.
3. Registro: se solicitan datos como: Nombres, Apellido Paterno, Apellido Materno, CURP, contraseña y número de teléfono.
4. Validación de datos: se valida la CURP con api de terceros para obtener más datos como nombre completo, estado de origen (aquí se cuestiona sobre donde reside actualmente) género y año de nacimiento.
5. Chat de proximidad: Chat inteligente a largo plazo, el cual avisara sobre zonas de peligro o zonas seguras para transitar en base a informes policiales y prevenir un incidente por terceros.
6. Servicios: recargas YOVOY, citas CAM o atención ciudadana.

### 2.3. Características de los Usuarios

Educación mínima para uso: N/A

Edad Rango de uso: 2 años

Edad máxima de uso: N/A

capacitación previa: Mínima

### 2.4. Restricciones

- **Políticas de la empresa:** La Entidad 'olimpo Tec' No se hace responsable por el uso indebido del software hecho para el cuidado de la ciudadanía, una falta administrativa derivada del uso inapropiado y sin consentimiento del software llevara a problemas legales.

### 2.5. Suposiciones y Dependencias

- Requisitos que pueden cambiar en base al avance del desarrollo del proyecto:
  - Lugar de Hosteo la base de datos
  - Plataforma de difusión
  - Interfaces
  - Colores
  - Formas



#### 2.6. Requisitos Futuros

- Comunicación mediante antena incorporada al celular
- Intercambio de datos entre celulares cercanos
- Partición de disco interno del dispositivo
- Adición a un dispositivo IoT para fácil transporte.

### 3. Requisitos Específicos

#### 1) Principios Generales de los Requisitos

- a) Los siguientes principios deben aplicarse a los requisitos:
- b) **Claridad y legibilidad:** El documento debe ser comprensible para personas con distintas formaciones.
- c) **Referencias a documentos relevantes:** Deben incluirse referencias a normativas de seguridad, privacidad y tecnologías aplicables.
- d) **Identificación unívoca:** Cada requisito tendrá un código de identificación para su trazabilidad.

#### 2) Características de los Requisitos

##### a) 3.1 Corrección

- i) Cada requisito reflejará una necesidad real para garantizar la seguridad de las usuarias. Ejemplo:
- ii) [REQ-001] La aplicación debe permitir enviar una alerta de emergencia vía SMS a un contacto de confianza con la ubicación en caso de peligro.

#### 3) 3.1.2 No Ambigüedad

- a) Cada requisito será preciso y verificable. Ejemplo:
  - i) [REQ-002] La aplicación debe permitir la configuración de hasta 5 contactos de emergencia.
- b) Incorrecto: "Debe enviar mensajes rápidamente" (ambigua la definición de "rápidamente").

#### 4) 3.1.3 Completitud

- a) Los requisitos incluirán todas las funcionalidades principales y respuestas del sistema ante diferentes escenarios:
  - i) [REQ-003] Si la conexión SMS no está disponible, la aplicación intentará enviar la alerta mediante Bluetooth a dispositivos cercanos previamente emparejados.

#### 5) 3.1.4 Consistencia

- a) Los requisitos no deben contradecirse. Ejemplo:
  - i) [REQ-004] La aplicación debe poder enviar alertas automáticamente al presionar el botón de emergencia sin requerir confirmación adicional.
  - ii) [REQ-005] La aplicación debe solicitar confirmación antes de enviar la alerta (Inconsistencia: debe decidirse una opción clara o permitir configurabilidad).

### 3 REQUISITOS ESPECÍFICOS

10

#### 6) 3.1.5 Clasificación

- a) Los requisitos se clasificarán según su importancia:
  - i) **Esenciales:** Funciones críticas para la seguridad de la usuaria (ej. envío de alerta por SMS).
  - ii) **Condicionales:** Mejoras que optimizan el uso (ej. integración con dispositivos de seguridad externa).
  - iii) **Opcionales:** Funcionalidades adicionales no críticas (ej. personalización de tonos de alerta).

#### 7) 3.1.6 Verificabilidad

- a) Cada requisito debe poder ser probado con un caso de prueba específico:
  - i) [REQ-006] "La aplicación debe enviar un SMS de emergencia en menos de 5 segundos tras la activación del botón de emergencia."
  - ii) Test: Medir el tiempo de envío en distintas condiciones de red.

#### 8) 3.1.7 Modificabilidad

- a) El sistema de requisitos se estructurará para permitir cambios fáciles y consistentes. Uso de herramientas de gestión de requisitos como Jira o RequisitePro facilitará la trazabilidad de cambios.

#### 9) 3.1.8 Trazabilidad

- a) Cada requisito debe ser rastreable hasta su origen y hasta los componentes del sistema que lo implementan.
- b) **Trazabilidad hacia atrás:** Cada requisito indicará de qué necesidad o documento de referencia proviene.
- c) **Trazabilidad hacia adelante:** Cada requisito debe estar vinculado a su implementación en código y sus pruebas asociadas.

### 3 REQUISITOS ESPECÍFICOS


11

#### 3.1. Interfaces Externas (EXPERIMENTALES)



Login

Registrarse



Atenea

CURP

Contraseña

[¿Olvidaste tu contraseña?](#)







#### 3.2. Funciones

##### Funciones especiales requeridas:

- **Intercambio de datos:** Registro y baja de datos de una base de datos la cual se mantendrá en constante monitoreo para su escalabilidad y mantenimiento.
- **Cifrado de datos y Seguridad de variables:** Cifrado de datos, tales como contraseñas, Datos personales de carácter SENSIBLE, uso de archivos '.env' para almacenamiento de variables, uso de protectores de código JS para prevenir ingeniería inversa del sitio web o de bloques de código protegido.
- **geolocalización:** Mediante herramientas web se obtendrá latitud y longitud del usuario a la base de datos para recuperación posterior.
- **Procesamiento de datos:** Se busca procesar los datos del usuario mediante apis de terceros y externas para almacenamiento y funciones de la misma aplicación.
- **Función de geolocalización:** Mediante Bluetooth de baja energía se hará una señal intermitente a una antena de baja frecuencia la cual buscará al dispositivo y validará tokens de usuario para su reporte y proceso administrativo.



### 3.3. Requisitos de Rendimiento

#### Dispositivos Moviles:

- 1 Gb de RAM.
- 500 MB de Espacio.
- Acceso a wifi.
- Acceso a Bluetooth.
- Acceso a archivos.

#### Dispositivos de escritorio (Desarrolladores y equipo de Atención C5)

- 8GB de RAM.
- 2 TB de almacenamiento en disco duro ó SSD.
- **SO:** Linux y/ó Windows 11
- Interfaces de Red y Bluetooth.
- Software de acceso remoto (credenciales protegidas por equipo de Ciberseguridad).
- Compiladores de código (Visual Estudio IDE, VIM, etc...)
- Compilador de código Python 3.1 o superior.

### 3.4. Restricciones de Diseño

- Colores amigables al usuario.
- Posible adición para condiciones autistas o de carácter especial.
- Interfaz intuitiva para el usuario.
- Botones grandes y claros para su fácil entendimiento y comprensión.

### 3.5. Atributos del Sistema

#### 1) 3.5.1 Fiabilidad

- a) La aplicación debe garantizar un funcionamiento estable en distintas condiciones.

Ejemplo:

- i) [REQ-007] La aplicación debe funcionar correctamente con al menos un 95% de fiabilidad en dispositivos Android con versiones 8.0 o superiores.

#### 2) 3.5.2 Mantenibilidad

- a) El código de la aplicación debe estar documentado y modular para facilitar actualizaciones:

- i) [REQ-008] El sistema debe permitir la actualización de contactos de emergencia sin necesidad de reinstalar la aplicación.

#### 3) 3.5.3 Portabilidad

- a) Debe garantizarse su compatibilidad con diferentes dispositivos:

- i) [REQ-009] La aplicación debe ser compatible con Android e iOS.

#### 4) 3.5.4 Seguridad

## 4 APÉNDICES

18

- a) Debe implementarse un mecanismo de autenticación para evitar accesos no autorizados:
  - i) [REQ-010] Solo la usuaria principal podrá modificar los contactos de emergencia mediante un PIN de seguridad.
  - ii) [REQ-011] La aplicación debe bloquear el acceso después de 3 intentos fallidos de autenticación.

### 3.6. Otros Requisitos

- Acceso a antena de comunicación interna.
- Acceso a memoria y cifrados.
- Accesos a micrófonos y cámaras.
- Accesos a registros telefónicos.
- Accesos a Localización en tiempo real o ultima registrada.

## 4. Apéndices

### 4.1. Diagramas de Arquitectura del Sistema

- Diagrama de Componentes: Representación gráfica de los módulos principales de la aplicación, incluyendo el servicio de geolocalización, el botón de emergencia, la base de datos, y la interfaz de usuario.
- Diagrama de Secuencia: Flujo de interacción entre el usuario, la aplicación, el servidor y los servicios externos (como APIs de mapas y servicios de emergencia) cuando se activa el botón de emergencia.

### 4.2. Especificaciones Técnicas

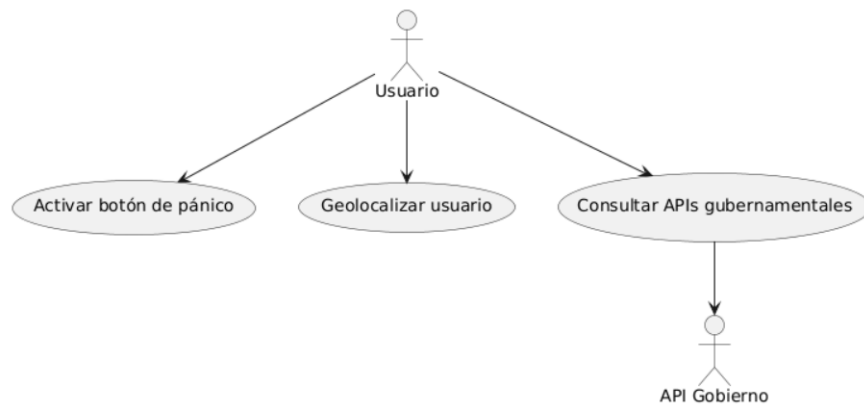
- **Requisitos de Hardware:**
  - Dispositivos móviles compatibles (versiones mínimas de iOS y Android).
  - Sensores necesarios (GPS, acelerómetro, etc.).
- **Requisitos de Software:**
  - Versiones mínimas de los sistemas operativos.
  - Librerías y frameworks utilizados (por ejemplo, Google Maps API, Firebase para notificaciones).

### 4.3. Protocolos de Comunicación

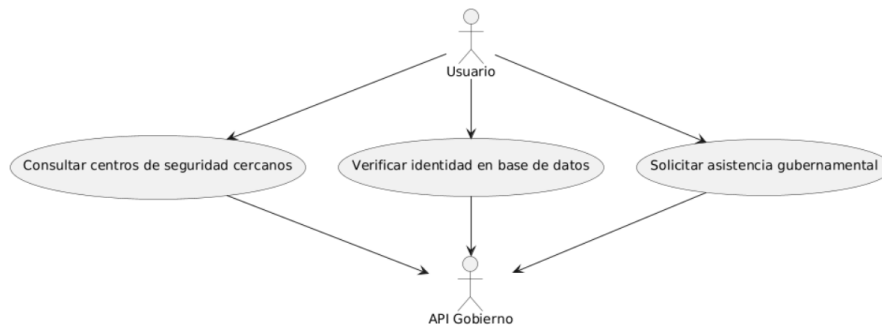
- Formato de Datos: Estructura JSON utilizada para enviar la ubicación y detalles de la emergencia al servidor.
- Protocolos de Seguridad: Uso de HTTPS y cifrado de extremo a extremo para proteger la información del usuario.

#### 4.4. Casos de Uso Detallados

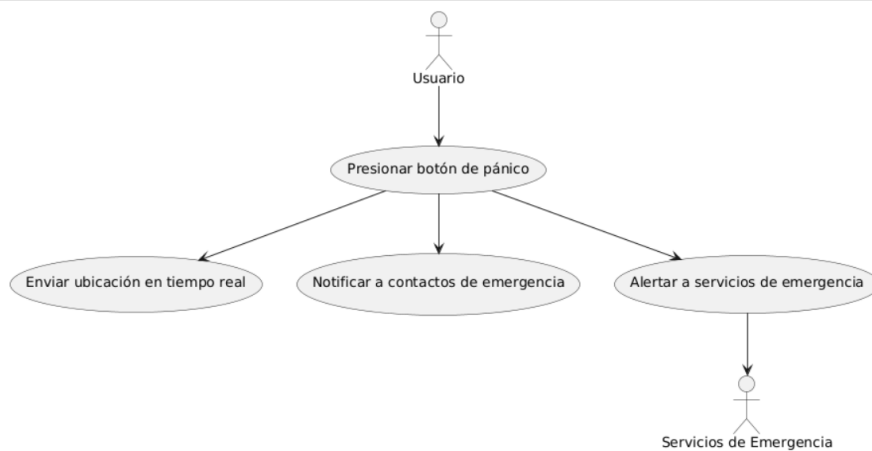
##### Uso general de usuario - Administrador



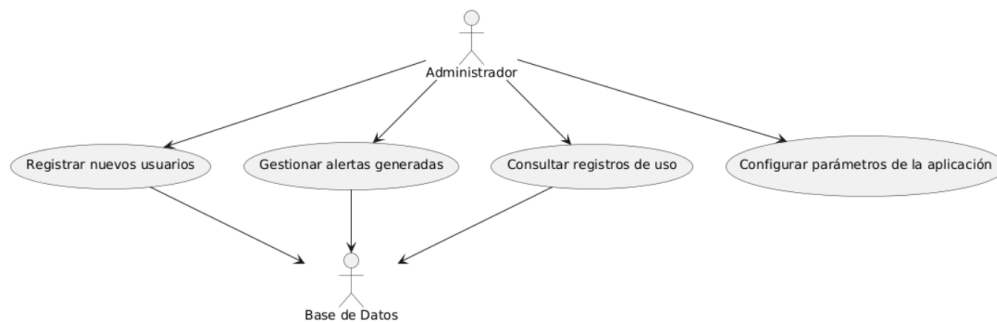
##### Consumo de apis de 3eros (Google maps, api de registro – login, uso de boton)



##### Activación de botón de emergencia



## Acciones de administrador(es)

**4.5. Pruebas y Validación**

- Pruebas de Usabilidad: Resultados de las pruebas realizadas con usuarios finales para evaluar la facilidad de uso del botón de emergencia.
- Pruebas de Rendimiento: Evaluación del tiempo de respuesta del sistema al activar el botón de emergencia en diferentes condiciones de red.
- Pruebas de Seguridad: Análisis de vulnerabilidades y medidas de protección implementadas.

**4.6. Glosario de Términos**

- Geolocalización: Proceso de identificar la ubicación geográfica de un dispositivo.
- Botón de Emergencia: Función que permite al usuario solicitar ayuda inmediata en situaciones críticas.
- API: Interfaz de programación de aplicaciones que permite la comunicación entre componentes de software.

**4.7. Referencias**

- Documentación oficial de Google Maps API.
- Estándares de seguridad para aplicaciones móviles (OWASP Mobile Top 10).
- Normativas de privacidad y protección de datos (GDPR, CCPA).

**4.8. Historial de Versiones**

- Versión 1.0: Primera versión funcional de la aplicación con geolocalización y botón de emergencia.
- Versión 1.1: Mejoras en la precisión de la geolocalización y optimización del tiempo de respuesta.

**Este documento es informativo para los directivos de 'olimpo tec' y Profesores de la Universidad Tecnológica De Aguascalientes (UTA), los cuales evalúan lo relacionado a Tecnologías de la información de cuatrimestre Enero – Abril. Su distribución sin previa consulta será tomada como usurpación de información y contenido ante un tribunal federal por derechos de autor y plagio comercial.**