



Instituto Politécnico  
Nacional



Escuela superior de cómputo

Plan Ejecutivo

---

## APP - Stay Quiet

---

*Autores:*

Hernandez Soriano Gerardo  
Moreno Sánchez José Rodolfo  
Perez Montiel Ulises  
Salas Hernandez Abiran Natanael  
Rincón Vieyra Alan

*Asesor:*

Ulises Vélez Saldaña



**S.E For Mobile Devices**

6 de septiembre de 2017

# Índice

<b>1. Abreviaturas.</b>	<b>3</b>
<b>2. Introducción.</b>	<b>3</b>
<b>3. Análisis del problema.</b>	<b>4</b>
3.1. Contexto. . . . .	4
3.2. Planteamiento del Problema. . . . .	4
3.2.1. Descomposición del problema. . . . .	4
3.3. Propuesta de solución. . . . .	4
3.4. Planeación del Alcance. . . . .	5
3.5. Objetivos. . . . .	5
3.5.1. General. . . . .	5
3.5.2. Específicos. . . . .	5
3.6. Análisis de Requerimientos. . . . .	6
3.6.1. Requerimientos Funcionales. . . . .	6
3.6.2. Requerimientos No Funcionales. . . . .	8
<b>4. Planeación de tiempo</b>	<b>9</b>
4.1. Metodología . . . . .	9
4.1.1. Mobile-D . . . . .	9
4.1.2. Cronograma . . . . .	11
4.2. Roles: . . . . .	12
<b>5. Planeación de Recursos Humanos:</b>	<b>12</b>
5.1. Roles: . . . . .	12
5.2. Responsabilidades. . . . .	13
5.3. Contacto. . . . .	17
<b>6. Referencias.</b>	<b>17</b>

## **1. Abreviaturas.**

Se usan las siguientes abreviaturas:

- API Interfaz de programación de aplicaciones
- BD Base de Datos
- IDE Entorno de desarrollo integrado
- GPS Sistema de posicionamiento global
- RF Requerimiento Funcional
- RNF Requerimiento no Funcional
- SDK Kit de desarrollo de Software
- UML Lenguaje Unificado de Modelado

## **2. Introducción.**

El desarrollo de la tecnología en los últimos tiempos ha tenido grandes avances, un ejemplo de ello es en el desarrollo de dispositivos y aplicaciones móviles, los cuales tienen distintos componentes que cuantifican diversos parámetros útiles a los usuarios en su vida diaria.

Hasta el momento, son pocas las aplicaciones que se enfoquen en el cuidado de personas con determinadas características que requieran supervisión geográfica para su cuidado.

### **3. Análisis del problema.**

#### **3.1. Contexto.**

Para reportar una persona desaparecida o extraviada se requiere esperar 72 horas.

Tiempo durante el cual una familia o un conjunto de personas designadas al cuidado de alguien requieren valerse de sus recursos para localizar a la persona de quien no saben donde se encuentra.

La señal de GPS cubre grandes áreas donde gran numero de personas habitan, por lo cual es altamente probable que se extravié un ser humano dentro de esta área

#### **3.2. Planteamiento del Problema.**

Hay seres humanos propensos a extraviarse o sufrir secuestro, por ejemplo, niños, ancianos, personas que padecen de sus facultades mentales y todos en general somos propensos a atravesar circunstancias semejantes

##### **3.2.1. Descomposición del problema.**

Con la intención de apoyar a las personas con determinadas necesidades de cuidar remotamente de otras, proponemos el desarrollo de un sistema para monitorio e los usuarios registrados por la persona interesada.

En el sistema se requiere identificar los siguientes módulos: - Módulo de registro de información personal, para que el usuario ingrese su nombre, correo electrónico, número telefónico. - Módulo de información geográfica, para que el usuario pueda visualizar la ubicación de sus "protegidos". - Módulo de historial, para registrar los eventos relevantes y poder consultarlos después. - Módulo de trabajo oculto al usuario, para vigilar la actividad y advertir al "protector" de la anomalía geográfica del "protegido", notificándolo de forma inmediata. - Emplear la tecnología GPS proporcionada por el dispositivo móvil.

#### **3.3. Propuesta de solución.**

Desarrollar una app móvil que mediante el uso de GPS que apoye a supervisar la ubicación de una o múltiples personas dentro de un área pre establecida

### **3.4. Planeación del Alcance.**

### **3.5. Objetivos.**

#### **3.5.1. General.**

Diseñar y desarrollar una aplicación móvil la cual se apoyará de los parámetros proporcionados por el GPS para supervisar la ubicación de una o múltiples personas dentro de un área preestablecida.

#### **3.5.2. Específicos.**

##### **Uso de GPS.**

La aplicación utiliza el servicio de GPS del celular para establecer la ubicación en un mapa

##### **Perfil de usuario protector.**

Permite tener contactos con los cuales puede compartir su ubicación cada determinado tiempo, y se llama observador cuando se vincula a un perfil protegido

##### **Perfil de usuario protegido.**

Para las personas que se quiere cuidar, la aplicación se mantiene funcionando como GPS, se requiere introducir un password para acceder a la aplicación

##### **Selección de área de cuidado .**

El observador puede establecer una zona geografica entorno a un punto fijo en el mapa ó a una distancia de una trayectoria

##### **Mensaje de alerta fuera de área establecida.**

Si el dispositivo con el perfil protegido sale del área de cuidado el observador recibe una alerta y puede iniciar una llamada

##### **Mensajes de alerta del dispositivo protegido .**

Si el dispositivo con el perfil protegido tiene batería baja, pocos datos o señal débil, se envía un mensaje al observador

### 3.6. Análisis de Requerimientos.

El desarrollo del análisis de requerimientos consiste en la definición de la aplicación, se basa en los objetivos que debe cumplir, así como las funcionalidades que ha de implementar.

La información que se debe aportar de cada requerimiento es

- **Identificador.** Código con el que se identifica dicho requisito.
- **Nombre.** Nomenclatura breve y descriptiva.
- **Descripción.** Desarrollo de los objetivos del requisito y las condiciones que pueden limitarlo.
- **Dependencia.** Código de los RF que son necesarios cumplimentar para poder acceder a ellos.
- **Prioridad.** Urgencia con la que debe desarrollarse dicho requisito.

#### 3.6.1. Requerimientos Funcionales.

Los Requerimientos Funcionales (RF) reúnen los servicios que ofrece el sistema, la reacción que debe tener en situaciones particulares y, en ciertos casos, especificará qué no debe hacer.

Datos para control de cambios	
<b>Versión actual:</b>	1.0
<b>Autor:</b>	Rodolfo
<b>Status:</b>	En espera de revisión
<b>Revisor:</b>	Ulises

Requerimientos funcionales identificados		
Id	Nombre	Descripción
<b>RF-01</b>	Registrar Usuario	El sistema debe registrar un perfil de usuario, con los datos personales que el usuario brinde
<b>RF-02</b>	Mostrar Ubicación geográfica de protegidos	El sistema debe contar con un apartado donde se muestre la ubicación geográfica de sus protegidos
<b>RF-03</b>	Alerta y Notificación	El sistema debe alertar al usuario mediante una notificación sobre la irregularidad en la localización o el estado de los dispositivos de sus contactos
<b>RF-04</b>	Sensibilidad en monitoreo	El sistema debe presentar alta sensibilidad a los cambios de ubicación de los usuarios protegidos (actualización de ubicación en periodos de tiempo cortos).
<b>RF-05</b>	Actualización de ubicación geográfica	El sistema debe actualizar los datos de la ubicación de los protegidos en la aplicación móvil del protector; en intervalos menores de 1 minuto para el estado de reposo, y menor de 5 segundos cuando el estado es de alerta.
<b>RF-06</b>	Alerta de riesgos	El sistema debe notificar al usuario si surgen imprevistos, (batería baja, pocos datos móviles disponibles o señal deficiente)
<b>RF-07</b>	Comunicación con protegido	El sistema proporciona comunicación entre los usuarios, ésta puede ser llamada telefónica y/o mensajes
<b>RF-08</b>	Rango de protección	El sistema brinda al protector la opción de definir un rango o una zona preestablecida de protección para sus protegidos
<b>RF-09</b>	Agregar protegidos	El sistema brinda al protector la opción de agregar n número de protegidos a su cuenta.

Requerimientos funcionales identificados		
Id	Nombre	Descripción
<b>RF-10</b>	El usuario protector podrá agregar n número de protegidos a su cuenta.	El sistema debe implementar un sistema de control de acceso a los usuarios que tengan el rol de protegido, vía pin y correo electrónico proporcionados por el protector.

### 3.6.2. Requerimientos No Funcionales.

Los Requerimientos No Funcionales (RNF) no hacen referencia a funciones del sistema, describen propiedades que éste debe cumplir. De este modo, se restringe de forma indirecta la aplicación.

Datos para control de cambios	
<b>Versión actual:</b>	1.0
<b>Autor:</b>	Rodolfo
<b>Status:</b>	En espera de aprobación del asesor
<b>Revisor:</b>	Ulises

Requerimientos No funcionales identificados		
Id	Nombre	Descripción
<b>RNF-01</b>	Política de privacidad	Al termino del registro del usuario en el sistema, se le deberá presentar la Política de Privacidad, la cual le explicará que sus datos serán confidenciales..
<b>RNF-02</b>	Disponibilidad del Sistema	El sistema deberá estar disponible cuando el usuario así lo requiera.
<b>RNF-03</b>	Documentación del sistema	El sistema deberá disponer de una documentación actualizable que permita realizar las operaciones de mantenimiento con la mayor eficacia posible.
<b>RNF-04</b>	Almacenamiento de Información	El sistema deberá almacenar la información del usuario en el almacenamiento local del teléfono inteligente.
<b>RNF-05</b>	Copia de Seguridad	El sistema deberá ser integrado a la API de copia de seguridad automática del sistema operativo nativo del teléfono inteligente.



## 4. Planeación de tiempo

### 4.1. Metodología

La IEEE define la ingeniería de software como la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de ingeniería al software.

La ingeniería de software se fundamenta en el compromiso con la calidad.

Para el óptimo desarrollo de este proyecto se acordó aplicar Ingeniería de Software, con base en la metodología de desarrollo para móviles, Mobile-D.

Esta técnica consiste en la creación de software dividida en cuatro etapas fundamentales:

**1. Análisis de requerimientos:** Se determinan los requisitos que debe de cumplimentar el software. Esta etapa es crucial para llegar a lograr los objetivos finales.

**2. Diseño y arquitectura:** Se establece un diseño conceptual de la aplicación a través de los Casos de Uso. Se muestran también en este apartado los primeros mockups de la aplicación.

**3. Desarrollo e implementación:** Se traslada el diseño realizado en el paso anterior a código. Se percibirán los primeros resultados de la aplicación en funcionamiento.

**4. Pruebas:** Es la fase final. Se comprueba el correcto funcionamiento de la aplicación.

#### 4.1.1. Mobile-D

Esta metodología está basada en diversas tecnologías como Rational UnifiedProcess, Extreme Programming y Crystal Methodologies, y su finalidad es intentar obtener pequeños ciclos de desarrollo de forma rápida en dispositivos pequeños.

Un ciclo de proyecto con la metodología Mobile-D está compuesto por cinco fases:

##### Fase de Exploración

Esta fase es la encargada de la planificación y educación de requisitos del proyecto, donde tendremos la visión completa del alcance del proyecto y también todas las funcionalidades del producto.

##### Fase de inicialización

La fase de inicialización es la implicada en conseguir el éxito en las próximas fases del proyecto, donde se preparará y verificará todo el desarrollo y todos los recursos que se necesitarían. Esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, la planificación inicial, el día de prueba y día de salida.

##### Fase de producción

En la fase de producción, se vuelve a repetir la programación de los tres días, iterativamente hasta montar (implementar) las funcionalidades que se desean. Aquí usamos el desarrollo dirigido por pruebas (TDD), para verificar el correcto funcionamiento de los desarrollos.

**Fase de estabilización**

Se llevarán a cabo las últimas acciones de integración donde se verificará el completo funcionamiento del sistema en conjunto. De toda la metodología, esta es la fase más importante de todas ya que es la que nos asegura la estabilización del desarrollo. También se puede incluir en esta fase, toda la producción de documentación.

**Fase de pruebas**

Es la fase encargada del testeo de la aplicación una vez terminada. Se deben realizar todas las pruebas necesarias para tener una versión estable y final. En esta fase, si nos encontramos con algún tipo de error, se debe proceder a su arreglo pero nunca se han de realizar desarrollos nuevos de última hora, ya que nos haría romper todo el ciclo. [A]

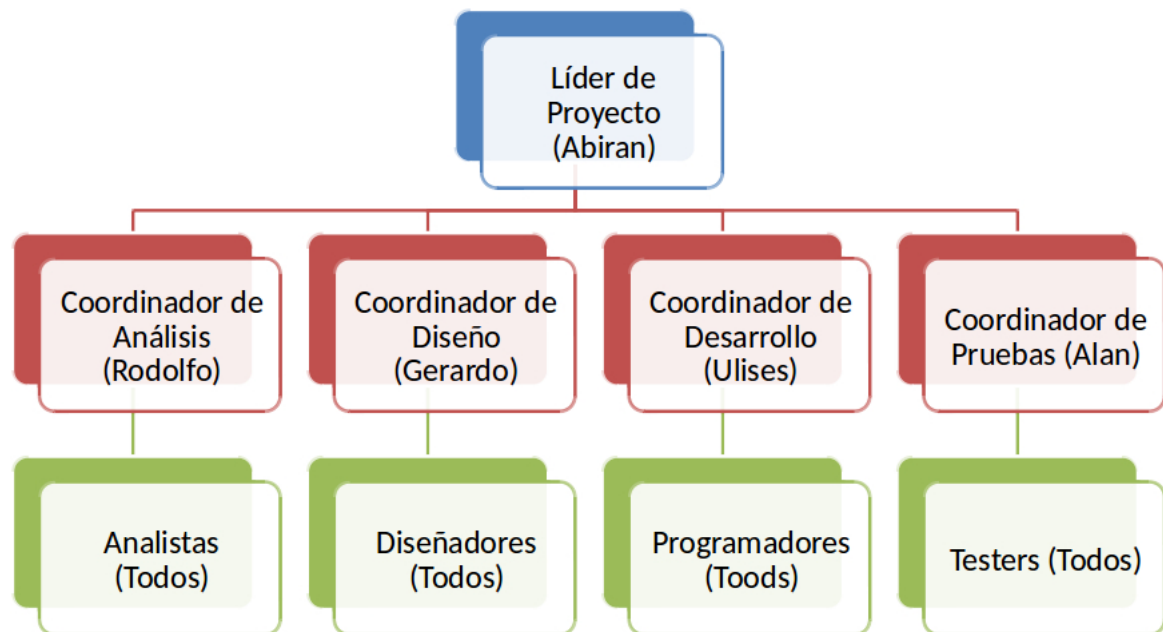
**4.1.2. Cronograma**

Actividad	Sep 10	Sep 26	Oct 10	Oct 27	Nov 10	Nov 21	Dic 05
Exploración	—		—		—		
Inicialización	—	—	—	—	—	—	
Producción		—		—		—	
Estabilización		—		—		—	—
Pruebas		—		—		—	—

#### 4.2. Roles:

### 5. Planeación de Recursos Humanos:

#### 5.1. Roles:



## 5.2. Responsabilidades.

### Líder de proyecto

- Reconocer los riesgos que puedan impactar la probabilidad de éxito del proyecto.
- Crear políticas para reducir el impacto de los riesgos.
- Planear la ejecución del proyecto en su totalidad.
- Resolver los problemas que se presenten durante la ejecución del proyecto.
- Crear acuerdos entre los coordinadores de las áreas involucradas en el proyecto.

### Coordinador de Análisis

- Establecer la forma de trabajo de los analistas y darla a conocer a todo el equipo.
- Asignar tareas a los miembros del equipo de análisis.
- Registrar los tiempos requeridos para realizar las tareas de análisis.
- Realizar el mapeo de procesos.
- Documentar requerimientos.
- Documentar casos de uso.
- Identificar y documentar reglas y términos de negocio.
- Diseñar pantallas.
- Realizar mapas de navegación.
- Responder las dudas en torno al análisis.
- Supervisar el avance del equipo de análisis.
- Establecer acuerdos con el resto de los coordinadores.
- Verificar el trabajo de los analistas.
- Participar en las reuniones de mejora de procesos.
- Documentar lecciones aprendidas.

**Analista**

- Realizar el mapeo de procesos.
- Documentar requerimientos.
- Documentar casos de uso.
- Identificar y documentar reglas y términos de negocio.
- Diseñar pantallas.
- Realizar mapas de navegación.
- Notificar avance al Coordinador de Análisis.
- Notificar cualquier cambio al alcance al Coordinador de Análisis.
- Documentar lecciones aprendidas.
- Participar en las reuniones de mejoras de procesos.

**Coordinador de Diseño**

- Establecer la forma de trabajo de los diseñadores y darla a conocer a todo el equipo.
- Planear las actividades de la etapa de diseño.
- Asignar tareas a los miembros del equipo de diseño.
- Traducir los elementos de análisis a en clases y entidades.
- Realizar los diagramas de secuencia para los procesos del sistema.
- Generar el diagrama entidad-relación de la base de datos.
- Responder las dudas en torno al diseño del sistema.
- Supervisar el avance del equipo de diseño.
- Establecer acuerdos con el resto de los coordinadores.
- Verificar el trabajo de los diseñadores.
- Apoyar al arquitecto de software en la definición de la infraestructura.
- Participar en las reuniones de mejora de procesos.
- Registrar los tiempos requeridos para realizar las tareas de diseño.
- Documentar lecciones aprendidas.

**Diseñador**

- Traducir los elementos de análisis en clases y entidades.
- Generar el diagrama entidad-relación de la base de datos.
- Realizar los diagramas de secuencia para los procesos del sistema.
- Informar al Coordinador de Diseño sobre posibles cambios o problemas en la arquitectura del sistema.
- Notificar al Coordinador de Diseño el avance.
- Participar en las reuniones de mejoras de procesos.
- Documentar lecciones aprendidas.

**Coordinador de Desarrollo**

- Establecer la forma de trabajo de los programadores y darla a conocer a todo el equipo.
- Planear las actividades de la etapa de implementación.
- Asignar tareas a los miembros del equipo de desarrollo.
- Implementar los componentes necesarios para los casos de uso asignados, definidos en el diseño.
- Realizar pruebas funcionales sobre los elementos programados.
- Enrolar buenas prácticas de programación para el proyecto.
- Asegurar la integración de los módulos del sistema.
- Responder las dudas en torno a la programación del sistema.
- Supervisar el avance del equipo de desarrollo.
- Establecer acuerdos con el resto de los coordinadores.
- Verificar el trabajo de los programadores.
- Apoyar al arquitecto de software (Coordinador de diseño) en la definición de la infraestructura.
- Participar en las reuniones de mejora de procesos.
- Definir la forma de solucionar las incidencias reportadas en la programación.
- Registrar los tiempos requeridos para realizar las tareas de desarrollo.
- Documentar lecciones aprendidas.

**Programador**

- Implementar los componentes necesarios para los casos de uso asignados, definidos en el diseño.
- Realizar pruebas funcionales sobre los elementos programados.
- Aplicar las buenas prácticas de programación definidas para el proyecto.
- Asegurar la integración de los módulos del sistema.
- Notificar el avance al Coordinador de Programación.
- Participar en las reuniones de mejoras de procesos.
- Documentar lecciones aprendidas.

**Coordinador de Pruebas**

- Establecer la forma de trabajo de los testers y darla a conocer a todo el equipo.
- Planear las actividades de la etapa de pruebas.
- Asignar tareas a los miembros del equipo de pruebas.
- Identificar los escenarios a probar en un caso de uso, módulo o sistema.
- Documentar los guiones de prueba del sistema.
- Documentar los datos de prueba.
- Ejecutar las pruebas y documentar los resultados.
- Dar retroalimentación a los programadores sobre los resultados.
- Dar seguimiento a las incidencias encontradas.
- Responder las dudas en torno a las pruebas e incidencias encontradas dentro del sistema.
- Supervisar el avance del equipo de pruebas.
- Supervisar que el proceso de pruebas se realice correctamente.
- Establecer acuerdos con el resto de los coordinadores.
- Verificar el trabajo de los testers.
- Participar en las reuniones de mejora de procesos.
- Registrar los tiempos requeridos para realizar las tareas de pruebas.
- Documentar lecciones aprendidas.



**Tester**

- Identificar los escenarios a probar en un caso de uso, módulo o sistema.
- Documentar los guiones de prueba del sistema.
- Documentar los datos de prueba.
- Generar los scripts de prueba para cada escenario.
- Ejecutar las pruebas y documentar los resultados.
- Dar retroalimentación a los programadores sobre los resultados.
- Dar seguimiento a las incidencias encontradas.
- Notificar el avance al Coordinador de Pruebas.
- Notificar al Coordinador de Pruebas de posibles fallos en el análisis o implementación.
- Participar en las reuniones de mejoras de procesos.
- Documentar lecciones aprendidas.

**5.3. Contacto.****Hernández Soriano Gerardo**

Alumno de la carrera de Ingeniería en Sistemas Computacionales en ESCOM

Tel. 55 2974 7782

Correo electrónico: ghernandezs1609@gmail.com

**Moreno Sánchez José Rodolfo**

Alumno de la carrera de Ingeniería en Sistemas Computacionales en ESCOM

Tel. 55 2801 2620

Correo electrónico: rodmorz@gmail.com

**Pérez Montiel Ulises**

Alumno de la carrera de Ingeniería en Sistemas Computacionales en ESCOM

Tel. 55 1617 9751

Correo electrónico: uperezm1001@gmail.com

**Salas Hernández Abiran Natanael**

Alumno de la carrera de Ingeniería en Sistemas Computacionales en ESCOM

Tel. 55 7275 3650

Correo electrónico: abisahe@gmail.com

**Rincón Vieyra Alan**

Alumno de la carrera de Ingeniería en Sistemas Computacionales en ESCOM

Tel. 55 4062 3689

Correo electrónico: alanVieyra376@gmail.com

**6. Referencias.****[A] Metodología Mobile-D. 2015. Disponible en:**

<http://manuelguerrero.blogspot.es/1446543763/metodologia-mobile-d-para-desarrollos-de-aplicaciones-moviles/>