

UAB

Universitat Autònoma de Barcelona

ESCUELA DE INGENIERÍA - Curso 2019/2020

LABORATORIO INTEGRADO DE SOFTWARE

CONTRATO DE TRABAJO

Agustín Tamayo - 1492214
Arnau Sarabia - 1492826
Jan Tugores - 1494134
Judith Bellido - 1455984
Angel Sacristán Ruiz - 1457243
Martin Susin Palacios - 1492417
Pau Borda - 1458601
Carles Milanés Horno - 1455091

Equipo docente:
Antonio Manuel López Peña
Lluís Gesa Bote
Jose Luis Gomez Zurita

7 de Junio de 2020



Índice

1	Introducción	2
2	Objetivos	3
3	Implementación	5
3.1	Herramientas	5
3.1.1	Front-end	6
3.1.2	Back-end	7
3.1.3	Organización de Actividades y Archivos	9
4	Pacto de Tareas	10
4.1	Propuesta para el 5	10
4.2	Propuesta para el 7	11
4.3	Propuesta para el 10	12

1. Introducción

Las problemáticas sociales son un concepto el cual afecta a todos los individuos por igual. Durante el año 2019, España ha experimentado distintas fluctuaciones en los índices de agresiones callejeras, incendios e accidentes en carretera. Lamentablemente, casuísticas como las agresiones sexuales han incrementado un 18 % desatando un miedo general en las personas.

La inseguridad, juntamente al sentimiento de estar indefenso, son sensaciones fruto de actuaciones no éticas y morales de la sociedad en la que vivimos. Actualmente, nadie puede afirmar con total seguridad que está protegido cuando sale a pasear. Sin embargo, a pesar de que nadie pueda otorgar una garantía absoluta en la seguridad de los individuos, podemos aportar pequeñas herramientas para cruzar el umbral establecido y aumentar el confort de los transeúntes.

Una de los principales motivaciones de los Objetivos Horizonte 2020 es la lucha por un cambio social. Actualmente, uno de sus mayores retos reside en la lucha contra la delincuencia para fomentar la sociedad segura.

Por este motivo surge SHiFT, una aplicación para todo tipo de usuarios que permite realizar llamadas de socorro a los servicios de emergencia independientemente de la situación en la que el afectado este inmerso. Esta, no únicamente permite realizar llamadas de ayuda, también transmite en directo la localización del socorrido y adquiere información de sus dispositivos cercanos para identificar a agresores, otras personas en peligro, entre otros.

Actualmente vivimos en la era de la información, el ser humano se está acostumbrando a realizar sus tareas de forma rápida y sencilla, sin estar ocupado en procedimientos los cuales pueden realizarse automáticamente. La aplicación SHiFT sigue este hilo de desarrollo posibilitando así activar alertas vía oral. De esta forma, el tiempo involucrado en usar el dispositivo móvil se ve reducido. Gracias a este factor, se incrementa el tiempo de defensa y decrementa el tiempo de socorro, el cual puede resultar clave para asistir a accidentados.

2. Objetivos

Antes de instanciar las distintas herramientas, conocimientos, prerequisites y otros factores necesarios para el desarrollo del proyecto en cuestión, resulta esencial entender los objetivos que SHiFT tiene. Mediante este paso, evitamos abrir puertas a confusiones y dejamos constancia de las distintas prioridades existentes.

Así pues, a continuación se reflejan los objetivos de la aplicación ordenados según su nivel de criticidad.

Objetivo	Criticidad
Ayudar en el proceso de socorro	Alta
Minimizar el tiempo de socorro	Alta
Facilitar la intervención de los servicios de emergencia	Alta
Crear una interfaz con un entorno <i>User Friendly</i>	Media
Comunicar emergencias a personas cercanas	Baja
Aportar recursos al peritaje posterior a la emergencia	Baja

Tabla 1: Tabla de objetivos.

Para facilitar la comprensión de los distintos puntos mostrados, se recorrerán las menciones con criticidad alta para otorgar detalles al lector. Las referentes a criticidades media y baja, quedan implícitas en las explicadas:

1. Ayudar en el proceso de socorro

Los servicios de emergencia, gracias a la aplicación SHiFT, recibirán de forma inmediata todas las llamadas de socorro de los usuarios. A pesar de reducir el tiempo requerido para pedir ayuda, se consigue ir un paso más allá mediante la transmisión de la geolocalización del socorrido y el reconocimiento de los dispositivos cercanos a él.

Gracias a los factores adicionales mencionados, aumentamos el nivel de precisión durante el rescate y obtenemos un reconocimiento lógico de su entorno mediante *beacons*. Así pues, podemos notificar automáticamente a otros usuarios de la aplicación la ayuda que pueden aportar a la situación.

2. Minimizar el tiempo de socorro

Un accidente es la perfecta escenificación para este objetivo. Muchas veces, la vida de un accidentado se debate en cuestión de minutos. Por este motivo, reduciendo el tiempo necesario para realizar la llamada de socorro y aumentando la libertad del ayudante para intervenir en la situación, producimos un incremento en las posibilidades de supervivencia del afectado.

Gracias a la automatización de procesos de SHiFT y su reconocimiento de voz, el usuario puede activar alertas sin necesidad de ocupar sus manos.

3. **Facilitar la intervención de los servicios de emergencia**

Toda alerta generada por un usuario en peligro será mostrada en una página privada para los servicios de emergencia. Esta, cuya recarga y sincronización con la base de datos se realizará de forma constante y automática, alertará de una persona en peligro a través de la web citada.

Esta, una vez notifique al técnico de emergencias, otorgará información crucial para evaluar la situación del usuario en peligro y actuar al respecto:

- Geolocalización.
- Edad.
- Nombre y Apellidos.
- Número de teléfono.
- Dónde vive (Ciudad, calle, número de casa o piso).

3. Implementación

Una vez contextualizado el problema y sus objetivos a abarcar, podemos proceder a tratar temáticas más procedimentales para constatar los elementos involucrados.

Primeramente, empezando desde una perspectiva global, se especificarán las herramientas necesarias para desarrollar SHiFT.

3.1. Herramientas

Para comprender y justificar las tecnologías que el proyecto envuelve, se muestra el siguiente esquema general de su arquitectura o distribución de elementos:

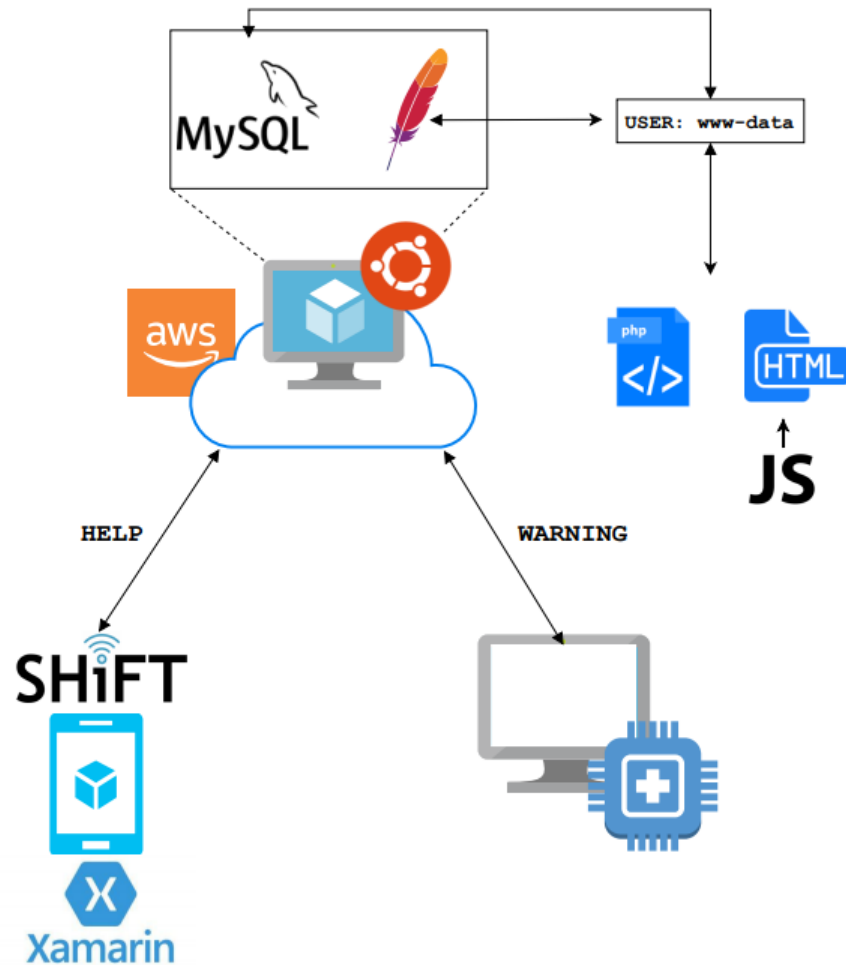


Figura 1: Representación general de la arquitectura de SHiFT.

Como puede apreciarse, hay una separación clara entre *front-end* y *back-end*. A pesar de su conexión, los factores que los fundamentan son muy distintos.

3.1.1. Front-end

En esta capa se separa la representación de los datos de la manipulación de estos. Por este motivo, está conformado por:

1. La aplicación SHiFT programada en C# mediante la plataforma *open-source* Xamarin.
2. La interfaz de usuario del servicio de emergencia desarrollado mediante HTML, CSS, JS y PHP.

Únicamente esta sección requiere de múltiples conocimientos como:

- El dominio de los lenguajes C#, HTML, CSS, JS y PHP.
- Conocimientos y habilidades avanzadas con la aplicación de desarrollo multiplataforma Xamarin y su organización MVVM.
- Entender perfectamente el funcionamiento de conexiones cliente-servidor y establecer un método de comunicación pactado.
- Dominar la validación de campos introducidos por el usuario en el lado del cliente.
- Conocimientos de desarrollo y aplicación de funciones relacionadas con la transmisión de la geolocalización.
- Conocimientos de desarrollo y aplicación de funciones relacionadas con la detección de voz (personal para cada individuo) y su debida traducción a texto.
- Diferenciación y comprensión de las metodologías de petición e introducción de datos GET, POST, PUT, DELETE, HEAD ...
- Conocimientos de desarrollo para interactuar con funciones internas del dispositivo móvil con tal de realizar zumbidos u otras indicaciones sensoriales.
- Proceso de *testing* y refinamiento con *white & black box tests*.
- Dominio de AJAX para sincronizar y actualizar el *browser* de emergencias con la base de datos sin necesidad de interactuar con el ordenador.
- Aplicación de serialización de datos para almacenar las configuraciones de las alertas y datos del perfil de forma local. Así pues, una vez se reinicie la aplicación estos seguirán existiendo y permanecerán totalmente privadas. A su vez, se reduce el número de peticiones al servidor y, por ende, la saturación de la base de datos.
- Control de la tecnología *beacon* para reconocer y notificar a los usuarios cercanos de SHiFT la existencia de una persona en peligro.

A pesar de todos los elementos técnicos especificados, nada puede llevarse a cabo sin una reflexión teórico-conceptual de la arquitectura del proyecto:

- Diagrama de casos de uso.

- Documentación detallada de los casos de uso.
- Diagramas de secuencia.
- Diagrama de clases acorde con el modelo MVVM.
- Diagrama de Entidad-Relación para organizar la base de datos del *backend*.
- Mockups de la aplicación SHiFT.
- Análisis de Requisitos Funcionales y No Funcionales.
- Planificación cronológica y diagrama de Gantt.
- Identificación y definición de objetivos del proyecto.
- Identificación de la razón de ser de SHiFT.

Las tareas involucradas en esta parte conforman una gran lista que implica largas horas de trabajo y estudio. Posteriormente, el cumplimiento de los distintos puntos mencionados se realizará acorde con la cualificación a considerar.

3.1.2. Back-end

En esta sección nos encargamos del desarrollo de la lógica de SHiFT. Este, debe de ser completamente transparente para el usuario final.

Retomando la Figura 1, podemos identificar numerosos elementos que lo conforman:

- Máquina virtual con SO Ubuntu corriendo en AWS.
- Base de datos MySQL.
- Servidor Apache para crear un *web service*.

Esta sección requiere de los siguientes conocimientos:

- Gestión de AWS para levantar una máquina virtual con las características deseadas y sistema operativo Linux (Ubuntu en este caso).
- Configuración del entorno virtual:
 - Instalación de PHP7.2.
 - Instalación de Apache.
 - Instalación de MySQL y PDO para gestionarlo mediante PHP.
 - Configuración de conexiones SSH a través de criptografía asimétrica (Pública y Privada).
 - Configuración de usuarios privilegiados y no privilegiados para evitar modificaciones inesperadas.

- Configuración del servidor Apache para que escuche a través del puerto deseado y tome como archivo principal *index.php* en vez de *index.html*.
- Configuración del servidor Apache para que el código ubicado en *index.php* sea ejecutado al hacer *request* y no mostrado por pantalla como un simple texto.
- Configuración de MySQL para crear la base de datos *shift.db* y su correspondiente acceso a través de usuarios establecidos.

- Gestión de privilegios de acceso a los ficheros para que *www-data* (usuario de Apache) pueda acceder a los directorios y realizar modificaciones sin problemas.
- Entendimiento del funcionamiento básico de los entornos Linux y directorios de configuración así como la importancia de los grupos y usuarios del sistema.
- Habilidad para emplear el editor de texto Vim.
- Creación de las tablas de la base de datos mediante MySQL.
- Comprensión sobre sesiones y buen manejo de *cookies*.
- Dominio de funciones de codificación y decodificación en base64 para ocultar información mediante el intercambio de *cookies*.
- Habilidad para emplear PHP y la capa de abstracción de acceso a datos PDO para establecer y manipular la base de datos mediante instrucciones.
- Buena comunicación con los desarrolladores de *Front-end* para pactar los protocolos establecidos en las distintas peticiones y los campos enviados vía GET o POST, por ejemplo.
- Dominio del formato Json.
- Control de *multi-threading* y semáforos para permitir el acceso de múltiples usuarios de forma simultánea sin colapsar el servicio.

En esta sección, se requiere un dominio notable sobre sistemas Linux debidas las configuraciones necesarias para establecer las bases del proyecto.

Una vez está todo listo para empezar, empieza una larga etapa de desarrollo en la que se gestionan las peticiones de *Register*, *Login*, Modificar Perfil, Activar Alarma, Desactivar Alarma ...

3.1.3. Organización de Actividades y Archivos

Finalmente, mencionar que las herramientas involucradas en la gestión de actividades y organización de avances del equipo son Jira y Bitbucket, respectivamente.

No se profundizará más en este punto debido el conocimiento que todo individuo de la mención de *Software* tiene sobre él.

4. Pacto de Tareas

Llegados a este punto ya se han especificado los objetivos, motivaciones y herramientas y procedimientos involucrados para la creación del producto SHiFT.

Partiendo de las raíces expuestas, a continuación se realiza una propuesta de calificación acorde con el volumen de trabajo involucrado.

4.1. Propuesta para el 5

La nota final expuesta recoge las siguientes obligaciones necesarias para tener una aplicación simple y básica. En esta se cumplen:

1. TAREAS DE RAZONAMIENTO Y PLANTEAMIENTO

- Diagrama de casos de uso.
- Documentación detallada de los casos de uso.
- Diagramas de secuencia.
- Diagrama de clases acorde con el modelo MVVM.
- Diagrama de Entidad-Relación para organizar la base de datos del *backend*.
- Mockups de la aplicación SHiFT.
- Análisis de Requisitos Funcionales y No Funcionales.
- Planificación cronológica y diagrama de Gantt.
- Identificación y definición de objetivos del proyecto.
- Identificación de la razón de ser de SHiFT.

2. TAREAS PROCEDIMENTALES

- Desarrollar una aplicación simple la cual no aplique ningún sistema de transmisión de geolocalización y detección de voz. Las alarmas se generarían mediante la selección de tres botones distintos, eliminando la parte de IA y autonomía prometida.
- La aplicación no aplicaría ninguna serialización de objetos ya que las alarmas estarían definidas de forma predeterminada, evitando su personalización.
- Los usuarios de SHiFT tendrán la posibilidad de hacer *Login*, *Register* y modificar sus perfiles sin ningún problema.
- El servidor recibe las alertas y las añade en la base de datos para dejar constancia del problema ocurrido.

4.2. Propuesta para el 7

Dado el aumento de la calificación final, se añaden factores que aumentan la complejidad de la aplicación y la acerca más al objetivo final planteado.

1. Creación de la página de emergencias, con su respectivo *Login*, la cual se sincronizará de forma automática y transparente con la base de datos para recibir las alertas más recientes.
2. Mostrar en la página de emergencias información básica sobre el socorrido (Geolocalización, Nombre, Fecha de Nacimiento, Calle donde vive, Ciudad, Código postal, Móvil).
3. Asegurar una buena transmisión de la Geolocalización en directo entre Cliente - Servidor - Emergencias y el posicionamiento del usuario en el mapa.
4. Otorgar al usuario la posibilidad de desactivar la alerta mediante la introducción manual de su contraseña. Esta parte implica una rápida eliminación del *warning* generado y mostrado en la página de Emergencias.
5. *Testing* y refinamiento para asegurar el buen funcionamiento de SHiFT y sus funcionalidades implementadas.

Como puede apreciarse la dificultad de implementación se ve aumentada dada la aplicación de la transmisión de Geolocalización en directo. Para cumplir esta casuística, deben de contemplarse aspectos como la frecuencia de envío de coordenadas y refresco. También es necesaria la gestión y apertura automática de puertos secundarios del servidor únicamente dedicados a la escucha de dicha información.

La creación de la página de emergencias también supone un valor añadido puesta la dificultad de diseño de una interfaz intuitiva y amigable. Todo esto debe facilitar la gestión de alertas y disminuir el tiempo de respuesta ante ocasiones de socorro.

Finalmente, la sincronización automática entre la página de emergencias y el servidor para obtener valores actualizados puede suponer grandes problemas. En primer lugar, mediante Java Script, debe de aplicarse un sistema con una frecuencia de tiempo X para obtener información nueva de la BBDD. Todo esto, evitando saturar la máquina virtual y su gestión intermediaria de alertas.

4.3. Propuesta para el 10

Finalmente, para llegar a la máxima puntuación posible, se alcanza la mayor independencia entre SHiFT y el usuario. También se implementa una mayor eficiencia en el servicio que usa el personal de Emergencias.

1. Configuración de alertas personalizadas mediante reconocimiento de voz.
2. Serialización de las configuraciones y perfiles establecidos para evitar su volatilidad y decrementar la carga del servidor. Cada vez que la aplicación se abra, recargará automáticamente las llamadas de socorro establecidas y los datos del perfil de usuario.
3. Activación de alertas en 2 tiempos:
 - Invocación de la aplicación vía voz (Ejemplo: Oye SHiFT).
 - Manifiesto, mediante la vibración del móvil, de la escucha de SHiFT al usuario.
 - Activación de la alerta citando las palabras configuradas.
4. El usuario no necesitará realizar ninguna interacción manual con el teléfono para invocar su llamada de socorro. Las situaciones de peligro podrán ser gestionadas mediante la voz.
5. *Testing* y refinamiento para asegurar el buen funcionamiento de SHiFT y demostrar su utilidad.
6. Implementación de sistema multi-threading para posibilitar la gestión de múltiples usuarios en peligro de forma simultánea.
7. Mejora de la página de Emergencias para diferenciar entre *Warnings* atendidos y no atendidos. Así pues, cada alerta constaría de distintas fases y estados.
8. Implementación de tecnología *beacon* para advertir a usuarios SHiFT del peligro que corre una persona cercana. El socorrido también debe de ser un usuario de la aplicación.

Las dificultades añadidas para alcanzar la máxima puntuación son claramente notorias. La implementación del sistema de reconocimiento de voz en dos tiempos, serialización de configuraciones y perfiles y la aplicación de detección de *nearby devices*, son factores los cuales demoran notablemente el tiempo de desarrollo.

A todo esto, debemos de sumarle la complejidad de gestión de múltiples usuarios simultáneamente y la introducción de estados en los *warnings* generados.