



# Proyecto de computación I



**Universidad  
Europea**

**LAUREATE** INTERNATIONAL UNIVERSITIES



2019



## Índice de contenidos del Proyecto



<b>RESUMEN.....</b>	<b>3</b>
<b>INTRODUCCIÓN Y JUSTIFICACIÓN TEÓRICA.....</b>	<b>3</b>
<b>MOTIVACIÓN.....</b>	<b>4</b>
<b>ESTADO DEL ARTE.....</b>	<b>4</b>
APLICACIONES MÓVILES REPRESENTATIVAS: .....	4
CONCLUSIONES SOBRE EL ESTADO DEL ARTE. ....	5
<b>OBJETIVOS .....</b>	<b>6</b>
OBJETIVO PRINCIPAL.....	6
OBJETIVOS ESPECÍFICOS.....	6
<b>SOLUCIÓN PROPUESTA.....</b>	<b>6</b>
INFORMACIÓN NECESARIA QUE SERÁ TRATADA ( VARIABLES ) .....	6
ÁREAS DE CONOCIMIENTO QUE INTERVIENEN.....	7
TECNOLOGÍAS INTERVINIENTES .....	7
MEDIOS EMPLEADOS .....	8
MARCO REGULADOR TÉCNICO-LEGAL SOBRE PROTECCIÓN DE DATOS.....	8
<b>PLAN DE TRABAJO .....</b>	<b>9</b>
METODOLOGÍA A USAR .....	9
ETAPAS DEL PROYECTO.....	10
<i>Calendario del proyecto</i> .....	11
<b>ANÁLISIS DEL SISTEMA .....</b>	<b>12</b>
ROLES EN FICHARIFY .....	12
REQUISITOS .....	12
<i>Requisitos de capacidad</i> .....	12
<i>Requisitos de restricción</i> .....	16
ESPECIFICACIÓN DE CASOS DE USO.....	19
<b>DISEÑO DEL SISTEMA .....</b>	<b>20</b>
ENTORNO DEL SISTEMA.....	20
ARQUITECTURA SOFTWARE .....	21



PATRÓN DE DISEÑO .....	21
DÍAGRAMAS DE SECUENCIA UML. ....	22
<i>Consultas credenciales</i> .....	24
ESPECIFICACIÓN DEL DISEÑO DE COMPONENTES.....	25
<i>Componente del Sistema</i> .....	25
<b>IMPLEMENTACIÓN .....</b>	<b>27</b>
SEGURIDAD Y AUTENTIFICACION DE LOS DATOS .....	28
<i>Vista de la tabla de usuarios y sus ids</i> .....	29
<b>PRUEBAS .....</b>	<b>30</b>
PRUEBAS DE INTERFAZ, BB.DD, API, SERVIDOR .....	30
PRUEBAS DE COMPATIBILIDAD .....	31
<b>CONCLUSIONES .....</b>	<b>31</b>



## Resumen

Aplicación de monitorización de las horas que se trabajan en la jornada laboral, incluyendo validación de horas, predicción de rutinas, administración de periodos vacacionales y bajas.

Esta se mantendrá sencilla a la vez que será capaz de proporcionar tanto a usuarios como administradores diversas funcionalidades para asegurarse de que el control de los horarios sea sencillo y correcto.

## Introducción y justificación teórica

El pasado 12 de marzo entró en vigor la nueva ley de registro de horarios. Entre los efectos de esta ley se encuentra la obligación de las empresas de incluir el horario concreto de inicio y de finalización de las jornadas de trabajo, con el objetivo de evitar abusos laborales y fraudes por no pagar o compensar horas extra y que, además, la Inspección de Trabajo y la Seguridad Social puedan llevar un mejor control.

Entrando más en detalle, la ley especifica como deben ser estos registros de horarios:

- Todas las empresas deberán llevar el registro de horario de sus empleados **independientemente de la jornada** que estos tengan.
- La empresa estará obligada a **guardar el registro de horario** de los empleados durante **cuatro años**.
- Así mismo, **deberá estar disponible** para los empleados y los sindicatos.
- Los empleados deben **conocer la distribución y duración** de la jornada laboral ordinaria.
- Los sindicatos deben **conocer mensualmente las horas extras** realizadas por los empleados.

Esta serie de tareas administrativas se muestra más complicada para personas que trabajan de forma remota o de forma móvil, dado que no disponen de lugares físicos donde corroborar sus horas trabajadas.

No cumplir con estas obligaciones es una infracción grave y esta conlleva una multa, en su grado mínimo, de 626 – 1.250 euros; en su grado medio de 1.251 – 3.125 euros; y en su grado máximo de 3.126 – 6.250 euros.



## Motivación

El proyecto se sustenta sobre el objeto de crear la herramienta que estas empresas necesitan para asegurar el cumplimiento de esta anteriormente mencionada ley. La gestión de recursos humanos se vuelve más complicada cuando se va a llevar un control estricto sobre todas las rutinas de trabajo de tu plantilla, y automatizarla es una gran ayuda para la misma.

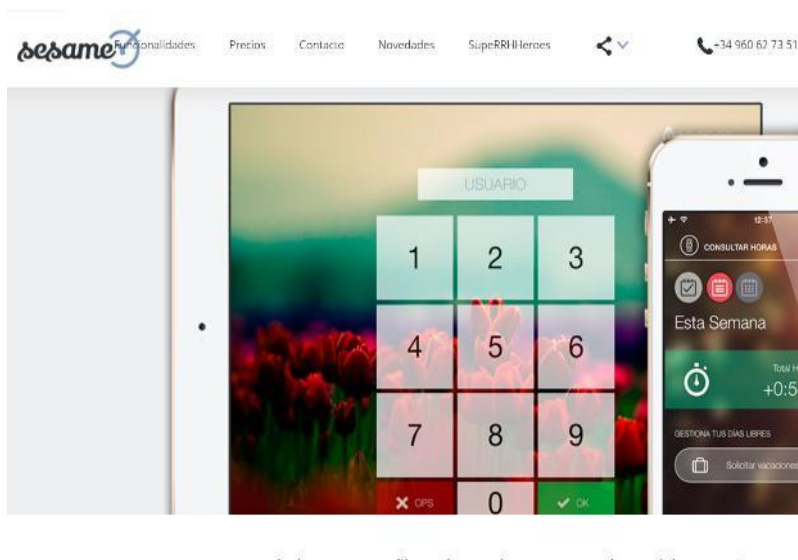
Además de ayudar con la elaboración de registros y el control de los horarios, se quiere ir un paso más allá, e introducir nuevas funcionalidades útiles para esa gestión [la de recursos humanos]

## Estado del arte.

Aplicaciones móviles representativas:

### Sesame

**Sesame** se trata de una de las empresas más reputadas del mercado nacional y cuyos servicios ya utilizan firmas como Ikea o Black&Decker. Además del control horario también ofrecen otros servicios como la gestión de las vacaciones y las ausencias, un gestor de proyectos para poder asignarlos a empleados o un panel de control ampliado. Esta empresa cobra una cuota fija en función del número de empleados que va desde los 29 euros al mes por cinco trabajadores.



### Intratime

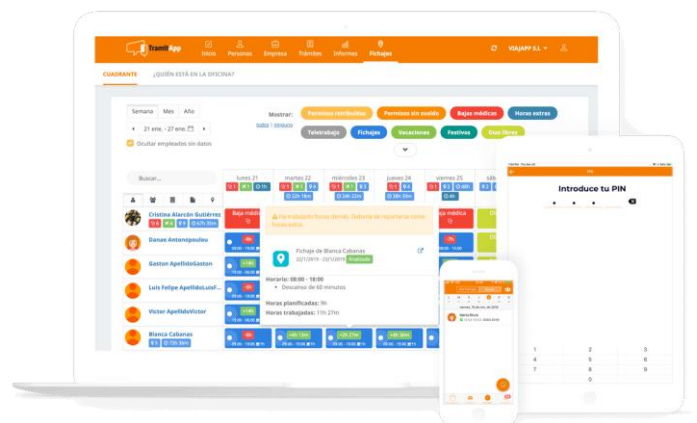


Una de las aplicaciones mejor valoradas en la sección de Productividad. Disponible tanto para Android como para Iphone, se puede controlar desde el ordenador y también se puede **geolocalizar** al trabajador y obtener listados con las horas que ha trabajado cada empleado.



### Tramitapp

TramitApp Control Horario permite a las compañías digitalizar las gestiones más habituales de recursos humanos como control de horarios, gastos de viaje y empresa y horas extras. Además, de otras funcionalidades como teletrabajo, la gestión documental o la firma electrónica. Todo ello, sin necesidad de que las compañías desarrollen sus propios sistemas, lo que les ahorra tiempo y dinero.



### Conclusiones sobre el estado del arte.

Del análisis de las aplicaciones móviles arriba citadas concluyo que :

1. A pesar de que existen bastantes aplicaciones disponibles, muy pocas muestran a los usuarios la información que quieren conocer, ni les permiten gestionarla de forma adecuada
2. Algunas de estas aplicaciones tienen un precio deseado para lo que ofrecen



3. Ninguna hace uso de algoritmos de aprendizaje para predecir rutinas de los empleados ni otros datos útiles para la gestión de recursos humanos

## Objetivos

### Objetivo principal

El objetivo principal es el de crear una aplicación de fichaje laboral que permita facilitar la administración de los recursos humanos, especialmente en el ámbito del control de horarios laborales, para automatizar dicha administración y aumentar la eficiencia de las gestiones

### Objetivos específicos

1. Que la empresa pueda saber que el empleado está trabajando las horas que le corresponden
2. Que los empleados puedan saber cuántas horas llevan trabajando durante ciertos periodos de tiempo
3. Predecir las horas de llegada de los empleados
4. Que el fichaje se pueda hacer en cuestión de segundos
5. Evitar reportes falsos, que son fáciles de generar con firmas manuales

## Solución propuesta

Se propone el desarrollo de un sistema conformado por una aplicación móvil para la gestión del horario laboral. Considerando que toda persona empleada tiene con toda seguridad un smartphone, se propone que la forma de distribución de la aplicación sea mediante descarga de las tiendas PlayStore de Android y Apple Store.

### Información necesaria que será tratada ( variables )

Las variables controladas en el estudio se dividen en dos grupos:

1. **Variables de introducción manual.** - serán obtenidas mediante la introducción manual del usuario vía la aplicación móvil.
2. **Variables de introducción automática.** - datos capturados diariamente durante las jornadas laborales de cada persona.

### Variables de introducción manual

1. Datos personales anónimos





- a. Edad (años),
- b. Sexo (mujer/varón),
- c. Correo electrónico
- d. Nombre y apellidos

### **Variables de introducción automática**

1. Aplicación
  - a. Horas trabajadas
  - b. Hora de entrada
  - c. Hora de salida
  - d. Localización

Áreas de conocimiento que intervienen en el proyecto.

- Ciencias de la computación
- Administración de empresas
- Marketing

Tecnologías intervinientes

### **Lenguajes de programación:**

- Typescript
- JavaScript



### **Frameworks**

- Ionic
- Angular



### **Control de versiones**

- Git
- SourceTree



Google Cloud Platform

### **Base de datos**

- Firebase
- Google Cloud Storage





### **Servidor web**

- Google Firebase

### **API's**

- Google Cloud Vision

### **Tratamiento de imágenes**

- Adobe Photoshop

### **Gestor de paquetes**

- Npm

### **Virtualización**

- Docker



### **Medios empleados**

- Dispositivos iPhone. Y Android- para testear la funcionalidad de la aplicación
- Licencia de Adobe Photoshop
- Licencia de desarrollador de Android y Apple
- Suscripción la plataforma Google Cloud. - nos proveen el dominio ¿????? , hosting, servidor web, SDK de autenticación, Api de reconocimiento fácil y la BB.DD.
- Licencia de Ionic Framework
- Ordenador Mac para generar la aplicación para iOS vía Xcode.

### **Marco regulador Técnico-Legal sobre protección de datos.**

En este punto describiremos las medidas legales que se han tenido en cuenta para la realización de nuestra aplicación.

Aunque los datos que se almacenan en la aplicación y en la base de datos NO son considerados como “datos personales” art. 7 L.O.P.D (Ley Orgánica de Protección de Datos) y dado que estos datos por si solos NO permiten identificar a personas físicas identificables o identificadas.

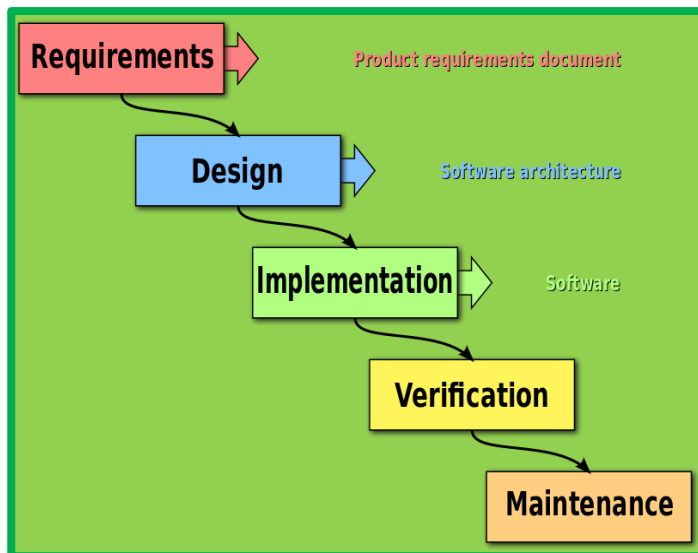
Sin embargo, debemos tener en cuenta una serie de medidas legales que han de cumplirse y son:



Cuando un empleado se logue por primera vez en la aplicación móvil deberá aceptar un [“Consentimiento Informado”](#) por los que da derecho al equipo de investigación y/o responsable de la aplicación a almacenar sus datos y tener acceso a los mismos.

## Plan de trabajo

### Metodología a usar



El sistema Ficharify no sigue una secuencia lineal, está expuesto a constantes cambios, aumento de requisitos, etc., etc. Esto hace que implementar siguiendo el modelo en cascada nos lleve al fracaso. En cascada la de creación del software tarda mucho tiempo ya que debe pasar por el proceso de prueba y hasta que el software no esté completo no se opera.

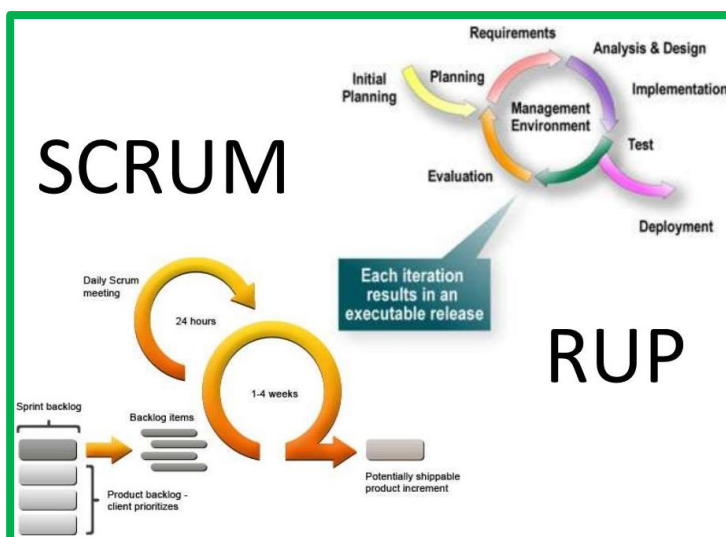
Cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo.

Además, una etapa determinada del proyecto no se puede llevar a cabo a menos de que se haya culminado la etapa anterior. Por lo tanto, decidimos NO usar metodología en cascada.

Sabíamos que debíamos usar metodologías ágiles, pero ¿cuál? ¿Scrum o RUP?



Aunque R.U.P es también una metodología ágil, las 2 utilizan el enfoque iterativo, RUP no está recomendada para equipos pequeños; y la principal diferencia es que R.U.P, tiene y sigue un plan formal, de principio a fin mientras que SCRUM el plan se genera en la siguiente iteración por lo tanto descartamos RUP y elegimos



SCRUM ya que nos ofrece muchas ventajas, la principal es que está orientada a equipos pequeños y a las aptitudes de cada uno, además la lista de objetivos está definida en el Project Back log y son reevaluados en cada Sprint. Además de que en RUP debemos entregar artefactos y en Scrum el único entregable es el propio software. Lo que nos otorga flexibilidad en el desarrollo.

En Scrum cada reunión nos preguntábamos:

- ¿Que hicimos ayer?
- ¿Qué vamos a hacer hoy ?
- ¿Hay obstáculos en el camino?

### Etapas del desarrollo del proyecto

El proyecto de investigación tiene 8 etapas, el presente proyecto solo abarca hasta la etapa de **“Recogida de datos.”**

Etapas / meses del 2019.	1	2	3	4	5	6	7	8	9	10	11	12
1) Análisis de los requerimientos	X											
2) Diseño arquitectónico	X											



3) Desarrollo		X	X	X	X							
4) Implementación					X	X	X	X				
5) Pruebas									X	X	X	
6) Redacción de las conclusiones												X

## Calendario del proyecto

Para este proyecto se ha establecido en siguiente calendario.

Nombre de la fase	Fecha Comienzo	Fecha Final	Numero horas
<b>Requisitos</b>	<b>10/8/19</b>	<b>11/9/19</b>	<b>150</b>
Recopilación de información	10/8/19	15/8/19	
Análisis de requisitos	15/8/19	20/8/19	
Entrevistas	20/8/19	25/8/19	
Presentación propuesta	26/8/19	30/8/19	
Modelo de Casos de Uso	01/9/19	06/9/19	
Especificación de Casos de Uso	07/9/19	08/9/19	
Especificaciones Adicionales	09/9/19	11/9/19	
<b>Diseño</b>	<b>12/9/19</b>	<b>20/10/19</b>	<b>190</b>
Aprendizaje de Typescript, Ionic, Angular	12/9/19	12/10/19	
Diseño lógico, físico, diagramas, arquitecturas	13/10/19	15/10/19	
Solución propuesta	15/10/19	20/10/19	
Tecnologías a usa / Modelo de Datos / Interfaces	14/10/19	18/10/19	
<b>Implementación</b>	<b>20/10/19</b>	<b>22/11/19</b>	<b>164</b>
Prototipos de Interfaces de Usuario	23/10/19	15/11/19	
Modelo de Implementación	15/11/19	22/11/19	
<b>Pruebas</b>	<b>02/12/19</b>	<b>06/12/19</b>	<b>20</b>
Alpha / Beta / Rendimiento / Instalación / Funcionales	02/12/19	06/12/19	



<b>Producción</b>	<b>20/12/19</b>	<b>05/01/20</b>	<b>22</b>
Modelo de Despliegue y salida a producción	27/12/19	02/01/20	
Publicación en las tiendas de aplicaciones	02/01/20	05/01/20	
<b>Redacción Memoria</b>	<b>04/01/20</b>	<b>09/01/20</b>	<b>23</b>
<b>Presentación Proyecto</b>	<b>15/01/18</b>	<b>15/01/18</b>	
<b>Total horas</b>			<b>569</b>

## Análisis del sistema

Una vez presentado y analizado el estado de la cuestión, pasaremos a comentar la siguiente etapa en el desarrollo del sistema que es la del análisis. En este apartado se estudiará el problema real con el cliente acordando el alcance del sistema y los requisitos, funcionalidades que deben satisfacer, como también los distintos casos de uso. En el siguiente paso se extraerá toda la información obtenida de los diferentes requisitos software, hardware, que especificarán detalladamente y especificarán comprobando su validez.

Para la obtención de los requisitos de usuario y los casos de uso se han realizado una serie de entrevistas con:

## Roles en Ficharify

Pendiente de redactar

Según lo visto en clase hemos podido determinar los siguientes:

- a) Requisitos de usuario
- b) Casos de uso
- c) Requisitos de software

A continuación, se detallan los requisitos que fueron presentados al cliente para su verificación y confirmación.

## Requisitos

Se han identificado dos tipos de requisitos:



1. **Requisitos de Capacidad:** Son los que definen las funcionalidades que el sistema / aplicación debe tener.
2. **Requisitos de Restricción:** nos indican las propiedades de las capacidades. Indicando restricciones temporales, de rendimiento, de seguridad, de disponibilidad, de estabilidad.

Para la especificación de cada uno de los requisitos se ha utilizado la siguientes nomenclatura y campos:

**Identificador:** Permite reconocer a cada requisito de manera única. Este campo tiene a su vez 2 valores:

- a) **UR** para los requisitos de capacidad
- b) **RUR** para los requisitos de restricción.

**Nombre:** Indica el nombre del objetivo.

**Descripción:** descripción detallada del requisito.

**Fuente:** muestra que persona ha solicitado el requisito o de donde proviene.

**Necesidad:** nivel de importancia de realizar el requisito y pueden ser : Opcional, Deseable y Esencial.

**Prioridad:** nivel de importancia del requisito y pueden ser Bajo, Medio y Alto.

**Claridad:** nivel de comprensión que tiene el requisito.

**Estabilidad:** Indica la estabilidad del requisito en relación con los demás requisitos y pueden ser Estable o Inestable.

**Verificabilidad:** nivel de comprobación del requisito, demuestra si se cumple o no.

**Pre-requisito:** Indica los requisitos necesarios para poder cumplir este requisito.

## Requisitos de capacidad

UR-01			
Nombre	Loguearse por primera vez usuario en la aplicación		
Descripción	El usuario podrá logarse en la aplicación Ficharify con la clave proporcionada por el administrador <ol style="list-style-type: none"><li>1. Email</li><li>2. Contraseña</li></ol>		
Fuente	Cliente	Necesidad	Esencial
Prioridad	Alta	Claridad	Alta



<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	<b>Alta</b>
<b>Pre-requisito</b>	<b>Ninguno</b>		

UR-02			
<b>Nombre</b>	<b>El usuario deberá introducir sus datos personales al iniciar la aplicación por primera vez</b>		
<b>Descripción</b>	<b>Deberá rellenar el cuestionario que tiene 3 apartados:</b> <b>1. Datos personales</b> <b>2. Datos de rendimiento</b>		
<b>Fuente</b>	Cliente	<b>Necesidad</b>	<b>Esencial</b>
<b>Prioridad</b>	Alta	<b>Claridad</b>	<b>Alta</b>
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	<b>Alta</b>
<b>Pre-requisito</b>	<b>Ninguno</b>		

UR-03			
<b>Nombre</b>	<b>Registrar tiempo trabajado</b>		
<b>Descripción</b>	<b>La aplicación deberá registrar el tiempo que haya trabajado el empleado al final de cada jornada</b>		
<b>Fuente</b>	Cliente	<b>Necesidad</b>	<b>Esencial</b>
<b>Prioridad</b>	Alta	<b>Claridad</b>	<b>Alta</b>
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	<b>Alta</b>
<b>Pre-requisito</b>	<b>Ninguno</b>		

UR-04	
<b>Nombre</b>	<b>Predecir rutina empleado</b>





<b>Descripción</b>	<b>La aplicación debe permitir al administrador conocer una predicción de la hora más probable así como el tiempo de trabajo esperado de cada empleado</b>		
<b>Fuente</b>	Cliente	Necesidad	<b>Alta</b>
<b>Prioridad</b>	Media	Claridad	<b>Alta</b>
<b>Estabilidad</b>	Estable	Verificabilidad	<b>Alta</b>
<b>Pre-requisito</b>	<b>Ninguno</b>		

<b>UR-05</b>			
<b>Nombre</b>	<b>Validar registros de fichaje</b>		
<b>Descripción</b>	<b>El administrador deberá poder tener acceso y validar a los registros de horas de jornada generados</b>		
<b>Fuente</b>	Cliente	Necesidad	<b>Esencial</b>
<b>Prioridad</b>	Alta	Claridad	<b>Alta</b>
<b>Estabilidad</b>	Estable	Verificabilidad	<b>Alta</b>
<b>Pre-requisito</b>	<b>Ninguno</b>		

<b>UR-06</b>			
<b>Nombre</b>	<b>Solicitar periodos</b>		
<b>Descripción</b>	<b>El usuario debe poder solicitar 2 tipos de periodos de inactividad, contando con la validación de la aplicación y del administrador:</b> <b>1. Periodos vacacionales</b> <b>2. Periodos de baja</b>		
<b>Fuente</b>	Cliente	Necesidad	<b>Esencial</b>
<b>Prioridad</b>	Media	Claridad	<b>Alta</b>
<b>Estabilidad</b>	Estable	Verificabilidad	<b>Alta</b>
<b>Pre-requisito</b>	<b>Ninguno</b>		



UR-07			
<b>Nombre</b>	<b>Geolocalización</b>		
<b>Descripción</b>	<b>La aplicación podrá registrar la ubicación de los empleados que no trabajen en un lugar fijo (ej: repartidores)</b>		
<b>Fuente</b>	Cliente	Necesidad	<b>Alta</b>
<b>Prioridad</b>	Media	Claridad	<b>Alta</b>
<b>Estabilidad</b>	Estable	Verificabilidad	<b>Alta</b>
<b>Pre-requisito</b>	<b>Ninguno</b>		

#### Requisitos de restricción

RUR-01			
<b>Nombre</b>	<b>Compatibilidad con el Smartphone</b>		
<b>Descripción</b>	<b>La aplicación será compatible con cualquier versión de Android e iOS, NO funciona en la plataforma Windows Phone</b>		
<b>Fuente</b>	Cliente y Jefe de Proyectos	Necesidad	<b>Esencial</b>
<b>Prioridad</b>	Alta	Claridad	<b>Alta</b>
<b>Estabilidad</b>	Estable	Verificabilidad	<b>Alta</b>
<b>Pre-requisito</b>	<b>Ninguno</b>		

RUR-02			
<b>Nombre</b>	<b>Idioma</b>		
<b>Descripción</b>	<b>El idioma de la aplicación deberá ser castellano, ya que está pensado para el mercado español.</b>		
<b>Fuente</b>	Cliente	Necesidad	<b>Esencial</b>
<b>Prioridad</b>	Alta	Claridad	<b>Alta</b>
<b>Estabilidad</b>	Estable	Verificabilidad	<b>Alta</b>
<b>Pre-requisito</b>	<b>Ninguno</b>		



RUR-03			
Nombre	Acceso a datos de usuario registrado		
Descripción	La aplicación solo mostrara la información relativa al usuario registrado.		
Fuente	Jefe de Proyecto	Necesidad	Esencial
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta
Pre-requisito	RUR-05		

RUR-04			
Nombre	Consentimiento informado		
Descripción	La aplicación deberá mostrar al usuario que se ha logado por primera vez un consentimiento informado con la opción de aceptar.		
Fuente	Jefe de Proyecto	Necesidad	Esencial
Prioridad	Media	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta
Pre-requisito	Ninguna		

**RUR-05**

<b>Nombre</b>	<b>Conexión a bluetooth y conexión a internet</b>		
<b>Descripción</b>	<b>El smartphone deberá estar dotado de conexiones wifi y/o GSM, HSDPA, UMTS para el envío en tiempo real de los datos</b>		
<b>Fuente</b>	Jefe de Proyecto y Cliente	<b>Necesidad</b>	<b>Esencial</b>
<b>Prioridad</b>	Alta	<b>Claridad</b>	<b>Alta</b>
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	<b>Alta</b>
<b>Pre-requisito</b>	<b>Ninguno</b>		

**RUR-06**

<b>Nombre</b>	<b>Validar credenciales</b>		
<b>Descripción</b>	<b>Dado que la aplicación estará disponible en Google y Apple Store cualquier persona puede descargarlas pero solo los usuarios con credenciales validas podrán acceder a ella</b>		
<b>Fuente</b>	Jefe de Proyecto	<b>Necesidad</b>	<b>Esencial</b>
<b>Prioridad</b>	Media	<b>Claridad</b>	<b>Alta</b>
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	<b>Alta</b>
<b>Pre-requisito</b>	<b>UR-02</b>		

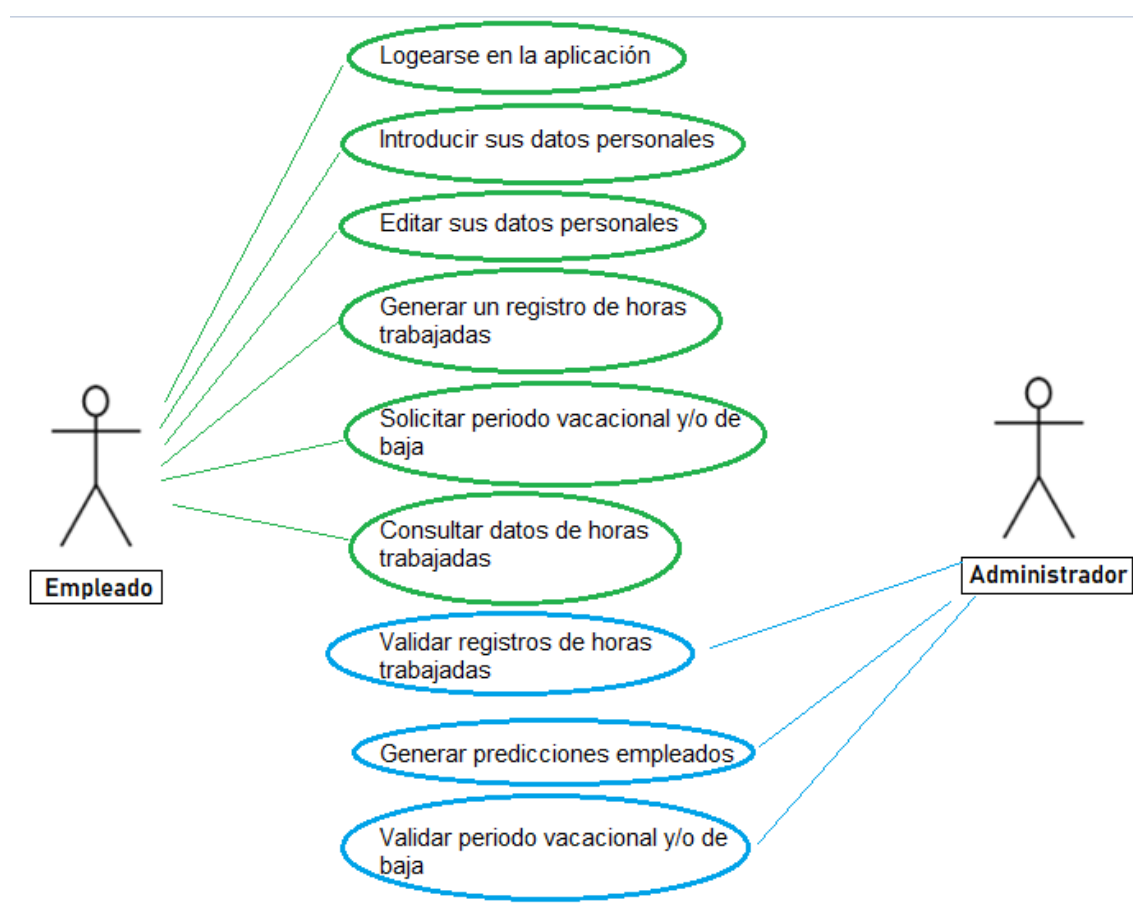
**RUR-07**

<b>Nombre</b>	<b>Protección de datos</b>		
<b>Descripción</b>	<b>Los datos almacenados en la aplicación tras el registro deberán estar anonimizados.</b>		
<b>Fuente</b>	Jefe de Proyecto	<b>Necesidad</b>	<b>Esencial</b>
<b>Prioridad</b>	Alta	<b>Claridad</b>	<b>Alta</b>
<b>Estabilidad</b>	Estable	<b>Verificabilidad</b>	<b>Alta</b>
<b>Pre-requisito</b>	<b>Ninguna</b>		



## Especificación de Casos de Uso

Conforme a los requisitos de usuario definidos anteriormente (UR-XX ), podemos determinar los casos de usos (UC-XX ), mediante los cuales vemos las comunicaciones e interacciones con los usuarios de la App.



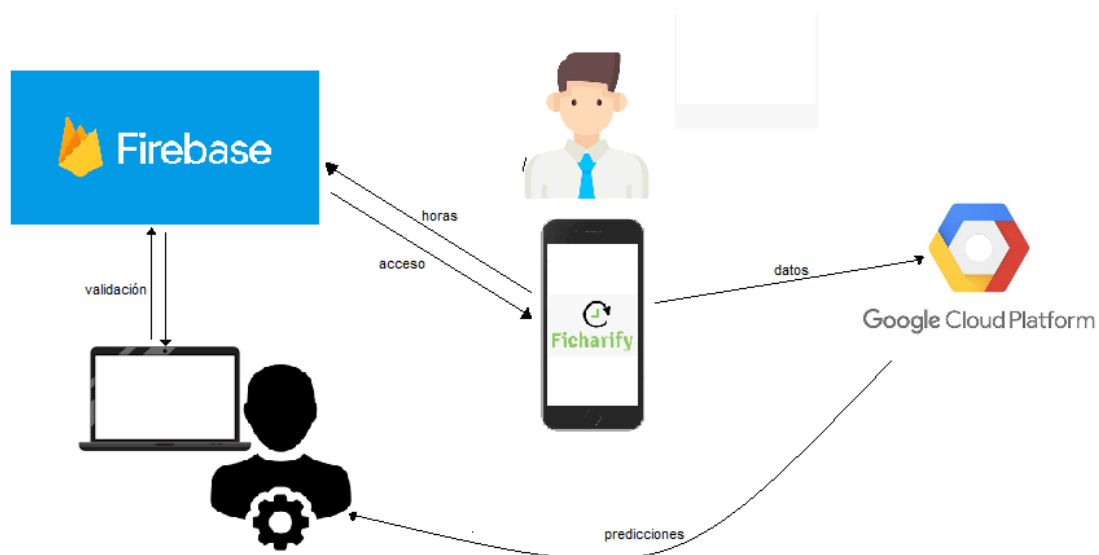


## Diseño del Sistema

Verificados los requisitos y casos de uso en esta etapa describiré el diseño del sistema, donde se integrará la aplicación, las bases de datos, su arquitectura software y el diseño de la misma.

Así mismo detallare los componentes del sistema y finalmente mostraré un prototipo que será presentado al cliente para su aprobación o modificación si hubiere cambios.

### Entorno del sistema



El proyecto consta de tres partes que se deben de comunicar para lograr un correcto funcionamiento:

#### a. Aplicación Web:

Permite al administrador del sistema validar los registros de horas generados, validar los periodos de inactividad, evaluar las predicciones, y consultar los datos de los empleados si fuese necesario

#### b. Aplicación de Móvil:

Permite a los trabajadores fichar, visualizar sus horas trabajadas en determinados periodos de tiempo y pedir periodos de vacaciones o bajas



**c. Base de Datos:**

Se almacena todo lo proveniente de las 2 fuentes

- Registros de horas
- Datos del empleado

Así como el resultado de las predicciones de las horas de entrada y de trabajo.

Además de guardar la información de los usuarios autorizados.

Arquitectura Software

En este apartado específico los componentes y la interacción entre los mismos. Ficharify consta de tres partes:

**a. La interfaz de interacción con los usuarios en:**

1. La aplicación móvil para empleados
2. La interfaz de interacción con los jefes y/o administradores en la aplicación web.

**b. Los datos almacenados en la base de datos:**

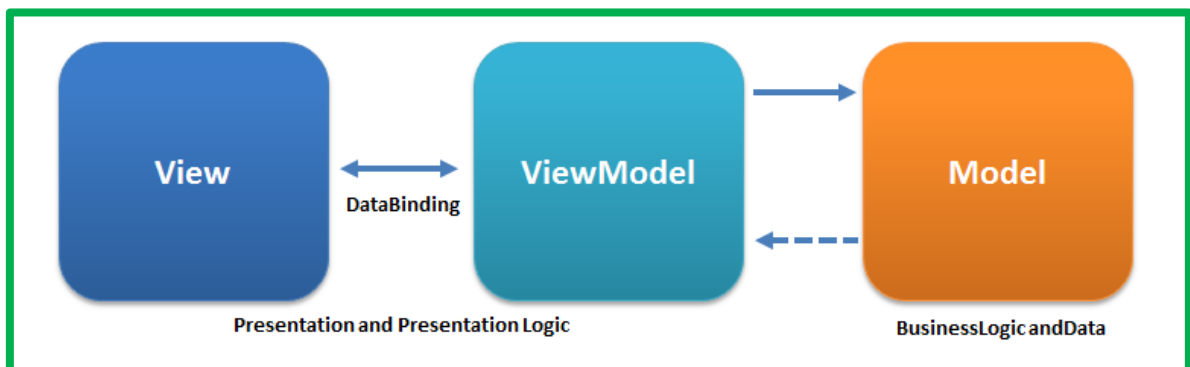
1. Registros de horas, claves de acceso, predicciones, etc

**c. Lógica de unión entre la parte a y b.**

Patrón de diseño

Por las características mi de sistema y por las restricciones propias del software a usar, [Ionic](#) y Angular que de forma nativa ya implementan el sistema MVVM y por encajar en mi sistema he decidido usar el modelo MVVM

[Model-View-ViewModel \(MVVM\).](#)



A continuación se describen las diferentes partes del:



### Model:

El modelo contiene los datos almacenados con los que trabaja la aplicación y permite interactuar con ellos y representarlos. Es independiente de los otros elementos.

### View:

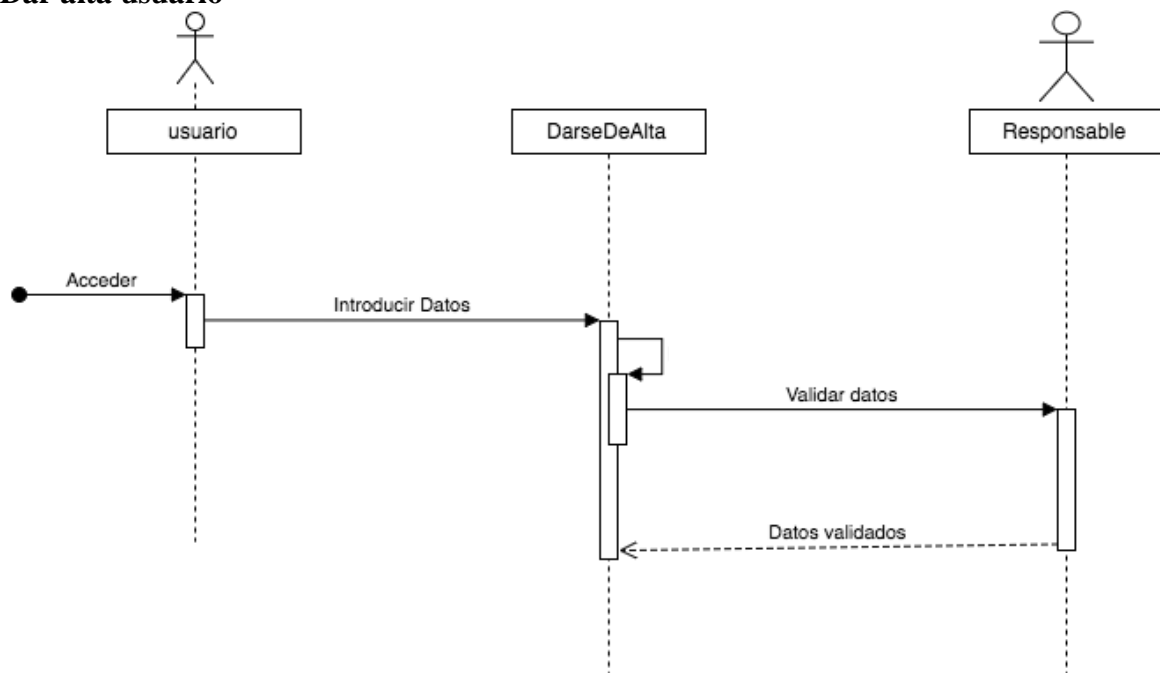
Son todos los elementos que aparecen en la interfaz gráfica de usuario tanto en la aplicación móvil como en la aplicación web tales como botones, graficas, navegación.

### ViewModel

Es el mediador entre la capa View y Model. Recibe peticiones de la View, que provienen del usuario, y realizar los cambios oportunos en el Model. View y ViewModel se comunican mediante un binder y se realiza por enlace de datos.

### Díagramas de secuencia UML.

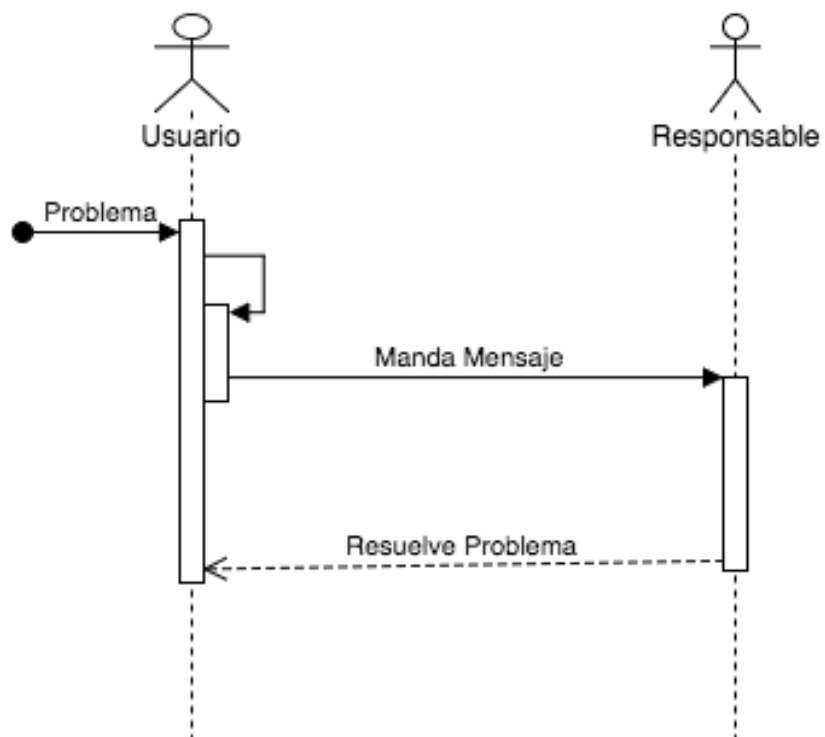
#### Dar alta usuario



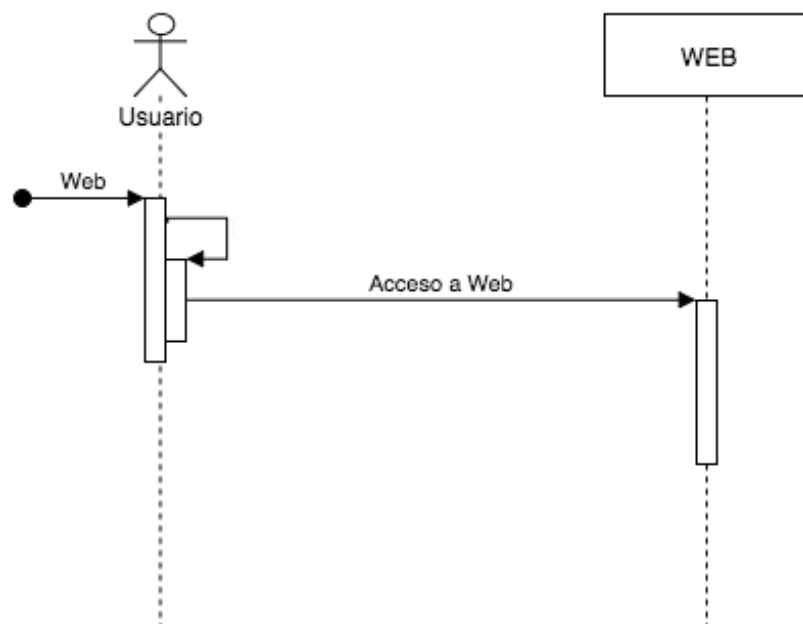




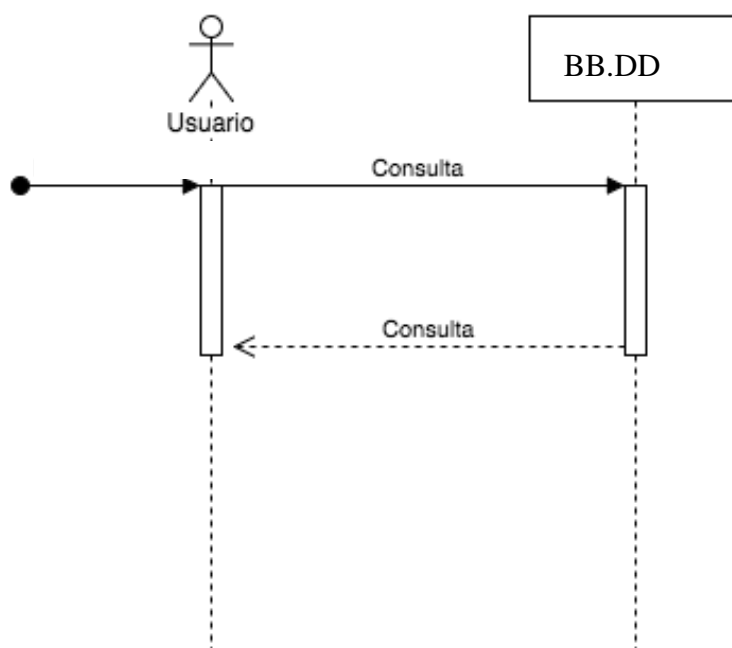
**Solicitar credenciales al responsable / administrador de Ficharify.**



**Acceder a la aplicación WEB**



### Consultas credenciales





## Especificación del diseño de componentes

Se describen los componentes que forman la aplicación en formato tablas.

**Nombre:** Indica el objetivo del componente.

**Tipo:** Indicará el nivel al que se encuentra el componente.

**Función:** Especifica detalladamente la funcionalidad del componente.

**Relacionados:** Son los componentes con los que estará enlazado el componente actual de

**Subordinados:** Especificará los componentes en los que se dividirá el componente actual.

## Componente del Sistema

COM1	
Nombre	Base de Datos Externa
Tipo	Módulo
Función	Este componente tiene como función poder acceder a los datos almacenados para obtención de datos, comprobación de credenciales, persistencia de los datos de los registros
Relacionados	Ninguno
Subordinados	COM2

COM2	
Nombre	Acceso al Sistema
Tipo	Módulo
Función	Este componente tiene como finalidad ofrecer una interfaz de acceso a los diferentes usuarios de la aplicación.
Relacionados	COM1
Subordinados	Ninguno

COM3	
Nombre	Interfaz
Tipo	Módulo
Función	Este componente tiene como función la renderización de la aplicación tras la realización de diferentes acciones.



Relacionados	COM1, COM2
Subordinados	Ninguno

COM4	
Nombre	Comunicación
Tipo	Módulo
Función	Este componente tiene como función proporcionar comunicación entre los usuarios y el SDK de autentificación
Relacionados	Ninguno
Subordinados	Ninguno

COM5	
Nombre	Internet
Tipo	Módulo
Función	Este componente tiene como función la comunicación de la aplicación con el otro proyecto y con diferentes aspectos de la aplicación.
Relacionados	COM3, COM4
Subordinados	Ninguno

COM6	
Nombre	Visualización
Tipo	Módulo
Función	Este componente tiene como función mostrar la información requerida por los usuarios obtenidas de la Base de Datos.
Relacionados	COM1
Subordinados	Ninguno



## Implementación

### Formato y tipos de datos

Se trabajará con 2 tipos de datos:

- a. Datos del empleado
  - Correo electrónico
  - Contraseña
  - Identificador (número aleatorio)
- b. Datos de rendimiento
  - Hora de entrada
  - Hora de salida
  - Horas trabajadas
  - Hora estimada de llegada
  - Días de baja

### SEGURIDAD Y AUTENTIFICACION DE LOS DATOS

Uno de los requisitos de usuario, mas importantes era mantener la seguridad de la información, así como la autenticación y login en nuestra aplicación.

Para cumplir este objetivo usamos la herramienta de autenticación de Google [Firebase Authentication](#) porque nos permite una integración en la aplicación sencilla y rápida, resaltar que estamos usando la infraestructura, respaldo y experiencia de Google en esta delicada tarea lo que aporta seguridad al proyecto, además de entre otras, las siguientes funcionalidades :

1. Autenticación directa
2. Integración con Facebook, Google, Twitter, GitHub.
3. Autenticación con número de teléfono

Dado que los datos deben ser anónimos solo pedimos una dirección de correo electrónico y una contraseña que lo asociamos a un **id** que se genera de forma aleatoria.



Vista de la tabla de usuarios y sus ids

Authentication					WEB SET
USERS					
SIGN-IN METHOD					
TEMPLATES					
USAGE					
Search by email address, phone number, or user UID					ADD USER
Identifier	Providers	Created	Signed In	User UID ↑	
luis@e.com	✉	Jan 11, 2018	Jan 11, 2018	2eTjl0E2EAgfS1d9KRvp0f3W.	
esteban@gmail.com	✉	Dec 15, 2017	Jan 6, 2018	2oLP1U9xBifOAC0rUoo2NI2Y	
p@gmail.com	✉	Dec 9, 2017	Dec 9, 2017	CbpdnySjRcXUfhOm75Zzeuf..	
gerente@dorsa.es	✉	Dec 3, 2017	Jan 4, 2018	CmcecJsDZwUpTcmtOyapsjU	
cristian@gmail.com	✉	Dec 15, 2017	Jan 4, 2018	JTDsm0oZt40Sk7zCWxVzfA..	

## Pruebas

Describiremos mediante varias tablas e ilustraciones las distintas pruebas que hemos realizado para verificar el correcto funcionamiento de la aplicación móvil y de la aplicación web así determinar si cumplimos con los objetivos expuestos en la planificación. Todas las pruebas fueron satisfactorias, destacar que en la fase de desarrollo se comprobaba toda funcionalidad antes de pasar a la siguiente tarea. Se encontraron mucho errores, pero fueron corregidos.

Pruebas de interfaz, BB.DD, Api, Servidor

ID	Ámbito	Pruebas	Descripción
----	--------	---------	-------------



01	Interfaz API	Login Verificar medico	Cuando accede por primera vez a la aplicación el médico debe poder acceder si tienen las credenciales correctas caso contrario no
02	BB.DD	Acceder a la aplicación	El acceso a la aplicación solo está disponible para los usuarios autorizados en la BB.DD
03	Aplicación	Registro horas trabajo	Se debe poder iniciar el registro de las horas y pararlo
04	Interfaz	Botones de la interfaz	Todos los botones deben funcionar al ser presionados
05	API	Envío de datos a Google Cloud	Los datos recogidos deben ser enviados a Google para las predicciones
06	BBDD	Conexión a Firebase	La aplicación debe poder consultar la base de datos y obtener los tokens de autorización y conexión.
07	BB.DD	Persistencia de la información en la BBDD	Todos los datos generados en la aplicación deben guardarse en la BB.DD



08	SDK Firebase	Persistencia temporal de los datos	Si no hay conexión a internet el SDK de Firebase deberá almacenar los datos hasta la siguiente conexión.
09	Interfaz	Se debe visualizar la landing page	Comprobación de que todos los links funcionen y que sea responsive
10	Dominio ficharify.com	Debe funcionar el dominio www.ficharify.com	Las peticiones al dominio deben funcionar para poder acceder
11	Aplicación	Editar datos del usuario	Los datos del empleado deben actualizarse satisfactoriamente





## Pruebas de compatibilidad

Dado que la aplicación ha sido construida en [Ionic Framework](#), por lo tanto la aplicación es híbrida, es decir no es nativa y la documentación de la tienda de aplicaciones Play Store de Google recomienda verificar la compatibilidad.

## Conclusiones

En el desarrollo de este proyecto, han intervenido muchas tecnologías nuevas para nosotros, y han sido complicadas de aprender. Hemos empleado mucho esfuerzo, cientos de horas, además todo lo aprendido en la universidad que ha jugado un papel fundamental; el esfuerzo ha valido la pena, el proyecto cumple las expectativas y los objetivos planteados al inicio del mismo.

En lo que respecta al proyecto desde el punto de vista técnico, he podido comprobar que el mercado cada día demanda aplicaciones totalmente funcionales en el menor tiempo posible por lo que la elección de la plataforma de desarrollo Google Firebase en unión al framework de desarrollo Ionic han sido las correctas.