22-3-2019 07-6-2019

Juanjo Martínez Sedano

Crynet Enforcement Local Logistics

Trabajo de Final de Grado Desarrollo de Aplicaciones Web

# Introducción

Esta es la documentación del Trabajo de Final de Grado de Desarrollo de Aplicaciones Multiplataforma (DAM), cuyo contenido consiste en este documento, el código fuente de la misma, y una serie de servicios web destinados a la lectura y envío de datos. La elaboración de todos estos apartados se ha realizado de forma individual por el alumno, Juanjo Martínez, y ha conllevado entorno a {CUANTAS} horas de trabajo.

También se adjunta la aplicación compilada, realizada por el alumno con ayuda de Severiano García en la realización y adaptación de alguno de los iconos.

Esta aplicación está inspirada en una empresa del videojuego *Crysis 2*, *CryNet*, intentando seguir sus principios de diseño en medida de lo posible. También toma inspiraciones de otras obras distópicas basadas en un futuro cercano.

## Licencia del proyecto

El proyecto se distribuye con una licencia Creative Commons (CC BY-ND 4.0).

Esta licencia permite la distribución del TFG en cualquier medio y formato, con cualquier finalidad, siempre y cuando se respete la autoría del autor de la obra (Juanjo Martínez), referenciándolo adecuadamente. También es importante mencionar que no se permite la distribución de copias derivadas de este trabajo, ni cambiar la licencia de este.

Si no se cumplieran estas condiciones, la licencia pierde toda validez junto a los derechos otorgados, por lo que no se podría compartir el trabajo bajo ninguna circunstancia.

Esta licencia no otorga ninguna garantía, y podría no dar todos los permisos necesarios para su uso, pudiendo estar limitada, por ejemplo, por otros derechos como publicidad, privacidad, o derechos morales.

Para toda la información sobre esta licencia, se invita al usuario a visitar la siguiente pagina web:

https://creativecommons.org/licenses/by-nd/4.0/

# Objetivos

La aplicación, *CryNet Enforcement Local Logistics App* o *CELL App* para abreviar, pretende cumplir una serie de necesidades impuestas por la empresa ficticia *Hargreave-Rasch Biomedical*, las cuales son:

* Acceso a Correo Electrónico
* Lista de tareas
* Sistema de notificaciones de actualización de políticas globales (GPU)
* Vista de perfil de empleados
* Mapa
* Reportes
* Gestión de almacén
* Transmisión de claves TOTP mediante NFC

Por encima de estos objetivos fijados por el alumno, prima la tarea de poner en practica todo lo estudiado durante el grado,

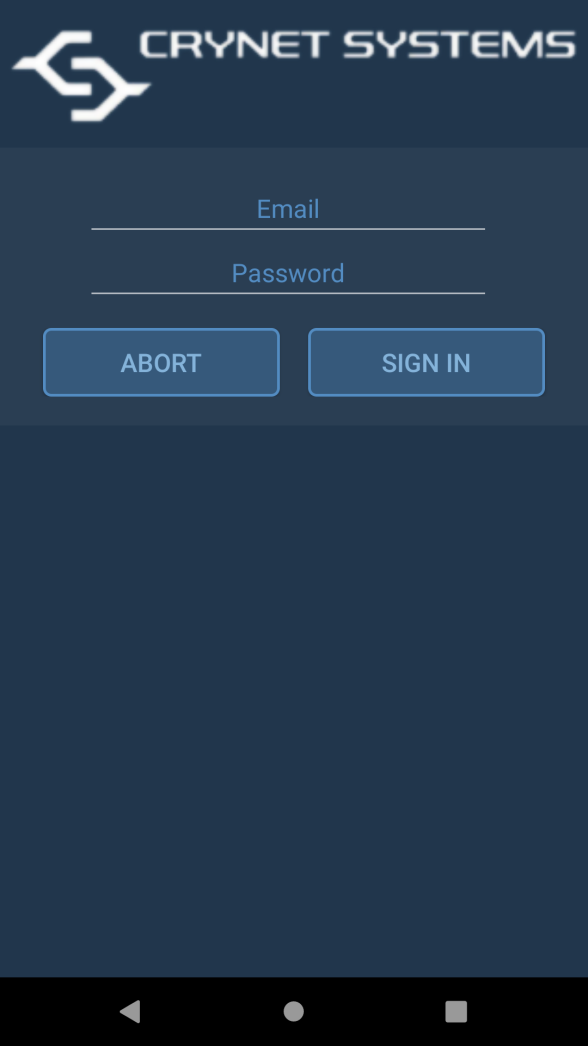
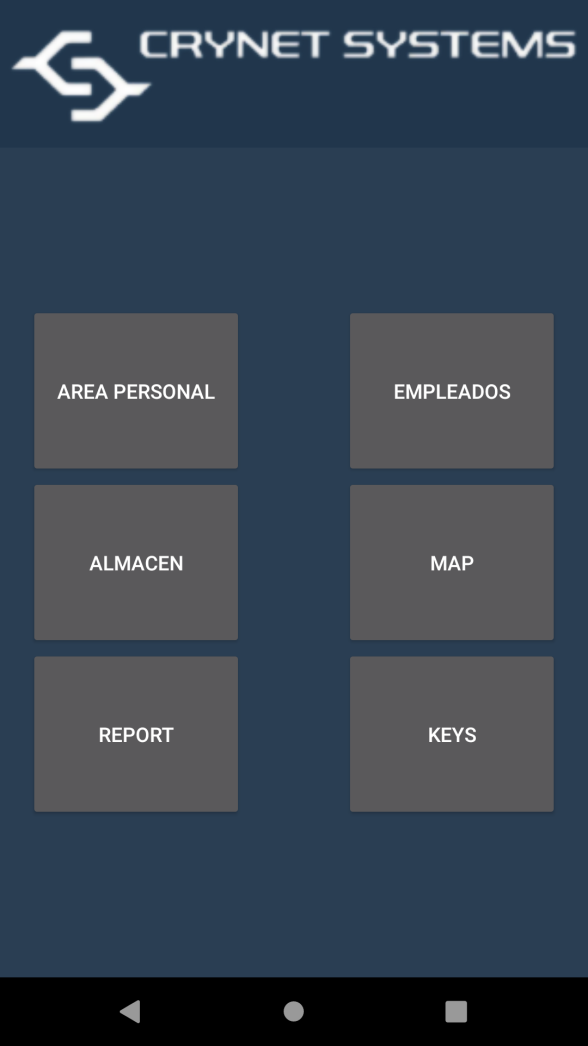
# Fases del proyecto

## Diseño:

El diseño siempre ha girado en mente a una pantalla de login, que al completar llevaría a un menú principal con las diferentes funciones de la app.

La configuración de colores y la tipografía se eligieron tras observar diferentes diseños de *Crynet Systems*, y componen casi en su totalidad la pantalla de login y la principal .

Los iconos fueron recortados y modificados con diferentes combinaciones de colores para poder aparecer bien en la app.

### C:\Users\Administrador\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_1559678115.pngArea personal

Tras completar el estas vistas, el trabajo se centró en el área personal, que por aquel entonces era denominada simplemente Emails. Esta ventana fue la primera introducción importante al diseño, y la que más se distancia con respecto a las demás.



### Detalle del correo

Tras completar la anterior pantalla, se decidió que la mejor forma de mostrar los datos de los correos en detalle sería con una ventana emergente, parecida al diseño que utiliza google en Gmail, lo que terminó resultando en la vista de la derecha.

### C:\Users\Administrador\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_1559678177.pngEmpleado

La vista empleado es posiblemente la que más cambios experimentó durante la fase de diseño. Las diferentes ideas pasaron por una vista similar a una lista, con un “ítem” para cada empleado, similar a una lista de contactos, una vista “carnet” con la información de un solo empleado, o una vista en tabla de los diferentes empleados.

El resultado final se basa principalmente en la distribución del carnet, pero con un diseño “normal” de aplicación. El problema que esta vista presenta es que solo se puede ver un empleado, por lo que sería necesaria otra vista que te muestre todos, o una forma de buscar el empleado que se quiere seleccionar.



### Keypad

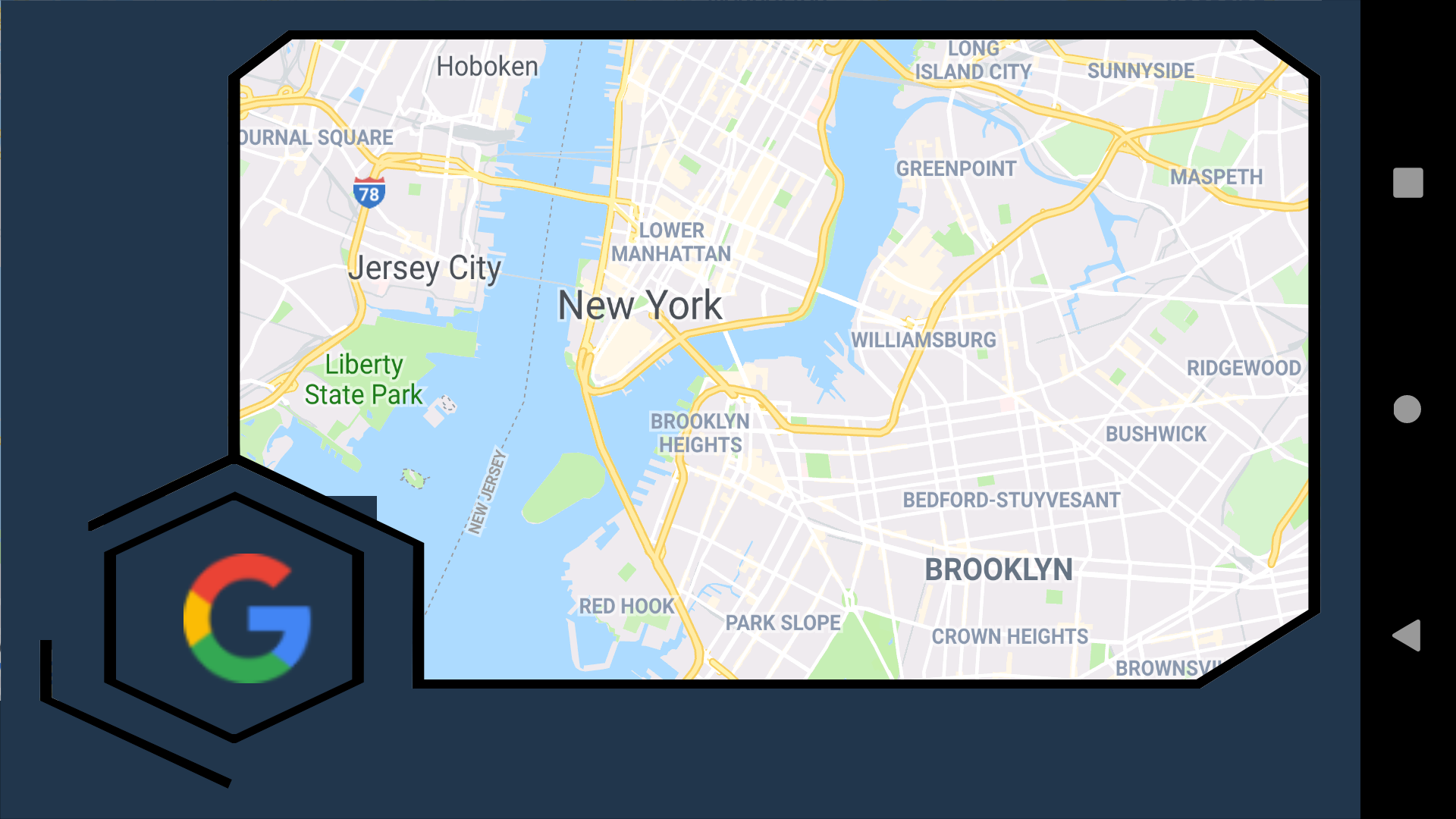
Esta vista es posiblemente la más simple de las que se dan por completamente terminadas en el apartado grafico, el único problema que presento fue decidir cuantos botones se debían poner en cada línea. Por lo demás utiliza los colores y fuente de la app, y no tiene mucho más.

### C:\Users\Administrador\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_1559678139.pngReport

Vista bastante simple, algo similar a la de correo en detalle, busca poder avisar de que algo no esta sucediendo como debería. En un principio se pretendía añadir mas campos a completar para el usuario, pero se terminaron descartando a favor de una interfaz más simple, que no requiere tanta atención para manejar, pero que transmite la misma cantidad de información o más.

### Mapa

El mapa busca cumplir un layout muy concreto, basado en el minimapa original del juego en el que se basa la aplicación. Sin embargo y por problemas de complejidad, este diseño no se adapta correctamente en distintos dispositivos



### Almacén

La vista del almacén fue planificada, pero no se llegó a plasmar en Android. Su diseño sería una vista con distintos ítems que mostrarían el resumen de cada objeto del inventario, con una implementación similar a como se ven las notificaciones en Android.

## Implementación y configuración

El trabajo de implementación del proyecto se puede separar en dos categorías, app y servicios web, cada una con sus complejidades propias.

### Login

#### Aplicación

El login utiliza varios métodos para checkear los credenciales antes de crear una tarea asíncrona que prueba los datos contra el servicio web. Si los datos recibidos del servicio son validos, los almacena en memoria y pasa a la siguiente vista.

#### Servicio web

Este servicio web busca en la tabla empleados por aquellos cuyo correo y contraseña coincidan con los datos especificados. Si no se encuentra ningún resultado devuelve un código de error 401(Unauthorized), con el mensaje “Usuario no válido”. En caso de encontrar un usuario cuyos datos coincidan, devuelve un token de acceso a usar por la app en todas las consultas online.

### Keypad

#### Aplicación

Esta actividad utiliza una serie de métodos para actualizar el teclado según se van pulsando teclas, aunque el valor almacenado al pulsar estas es independiente de la letra como símbolo general, la misma letra puede tener diferentes valores en función de varios valores. Estos valores no son aleatorios, sino que buscan el que se pueda transmitir la clave “encriptada”, no se puede pasar de valores a representación ni viceversa si no se dispone del algoritmo

### Empleado

#### Aplicación

La actividad de empleado llama al servicio web para obtener los datos de los empleados, utilizando el token de sesión como argumento. Una vez ha recibido los datos los pinta sobre los distintos campos de texto de la actividad

#### Servicio web

Se dispone de multiples servicios web para obtener datos de uno, varios, o todos los empleados con/o sus departamentos, que se leen desde una base de datos sql. Como todas los servicios de la app, se requiere de un token para llamar a estos servicios

### Area personal

#### Aplicación

Esta es posiblemente la actividad más compleja de toda la app; combina 5 listas (Recyclerview) que contienen datos tanto locales como obtenidos desde internet. El layout personalizado de todo esto funciona gracias a varios listeners de clicks en los diferentes apartados de la app, y los ítems de las listas tienen a su vez listeners personalizados para llevar a cabo la tarea que requieran. Todo esto termina resultando en 11 clases, sin contar las usadas para leer los datos de forma asíncrona.

#### Servicio web

La vista de área personal utiliza 3 servicios web que leen datos de 4 tablas diferentes, tratando adecuadamente los datos y devolviendo todo lo necesario de forma ordenada y limpia, en un único JSON con todos los datos.

### Configuración del servidor

El servidor requiere de Composer, Silex, y Redbean correctamente configurados en el directorio del vhost.

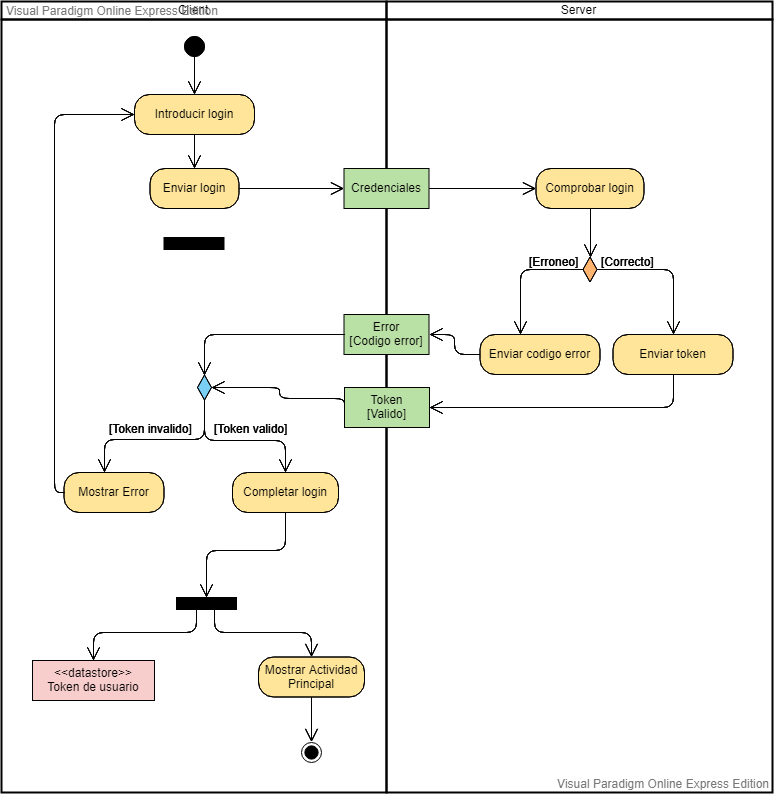
Además, el fichero funciones.php contiene una serie de funciones necesarias para que funcionen los servicios, entre las que se encuentra tratado de tokens y credenciales siempre que se vaya a ejecutar una consulta a un servicio web.

## Pruebas

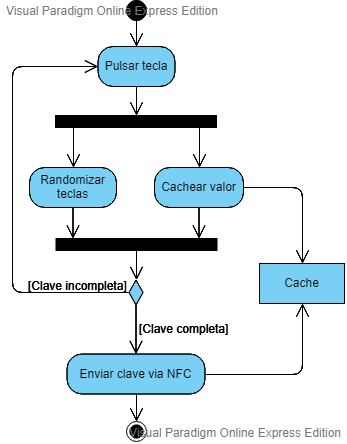
# Diagramas UML

### Diagramas de actividades

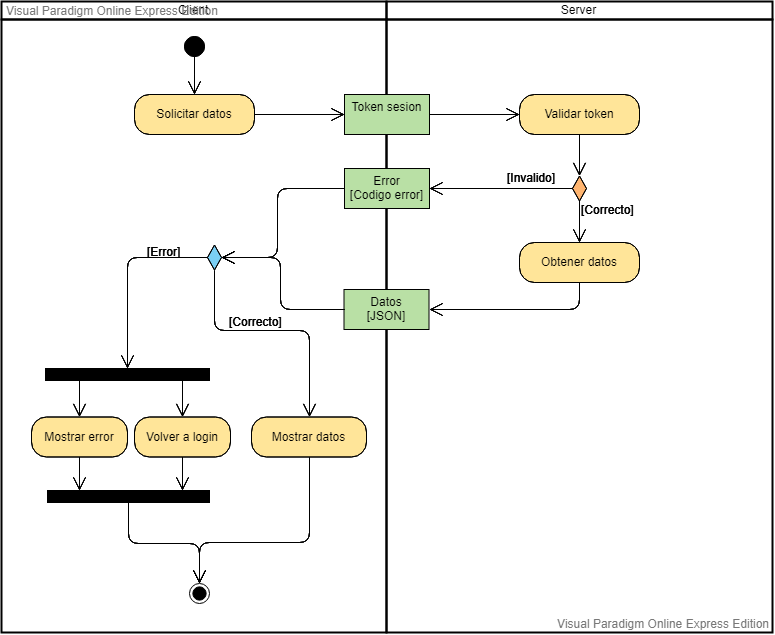
#### Iniciar sesion



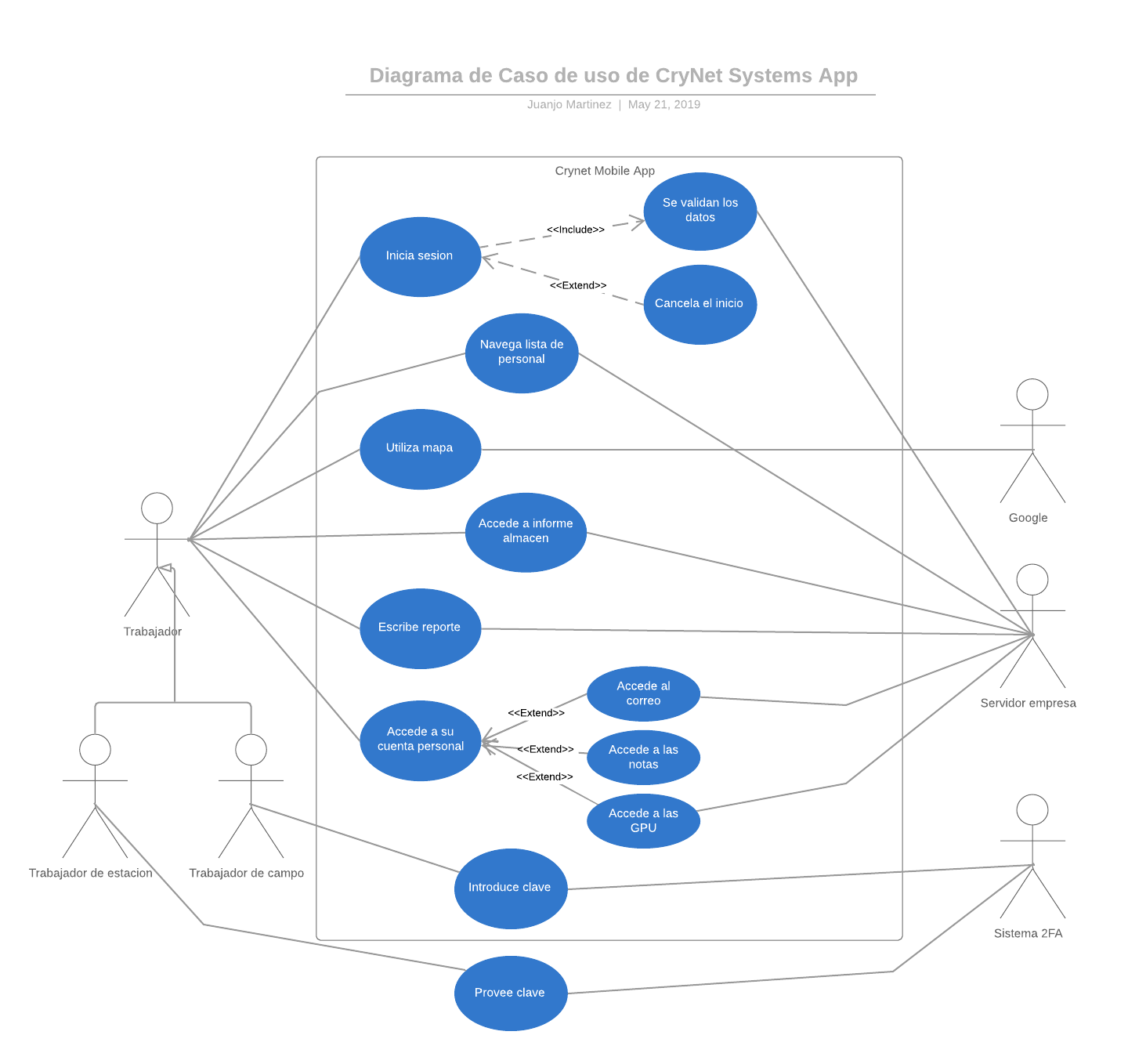
#### Funcionamiento



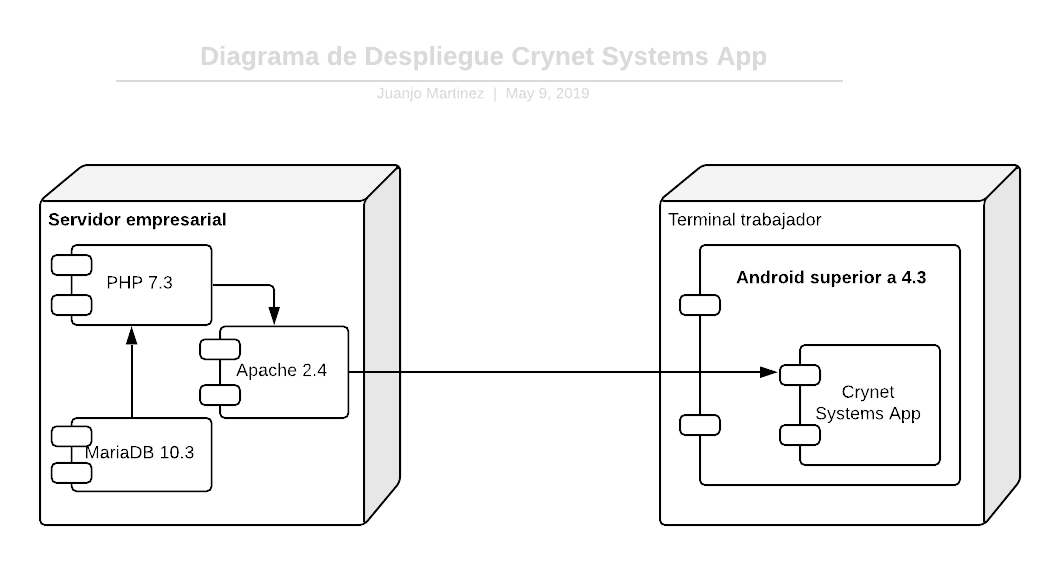
#### Obtener datos del servidor



### Diagrama de casos de uso

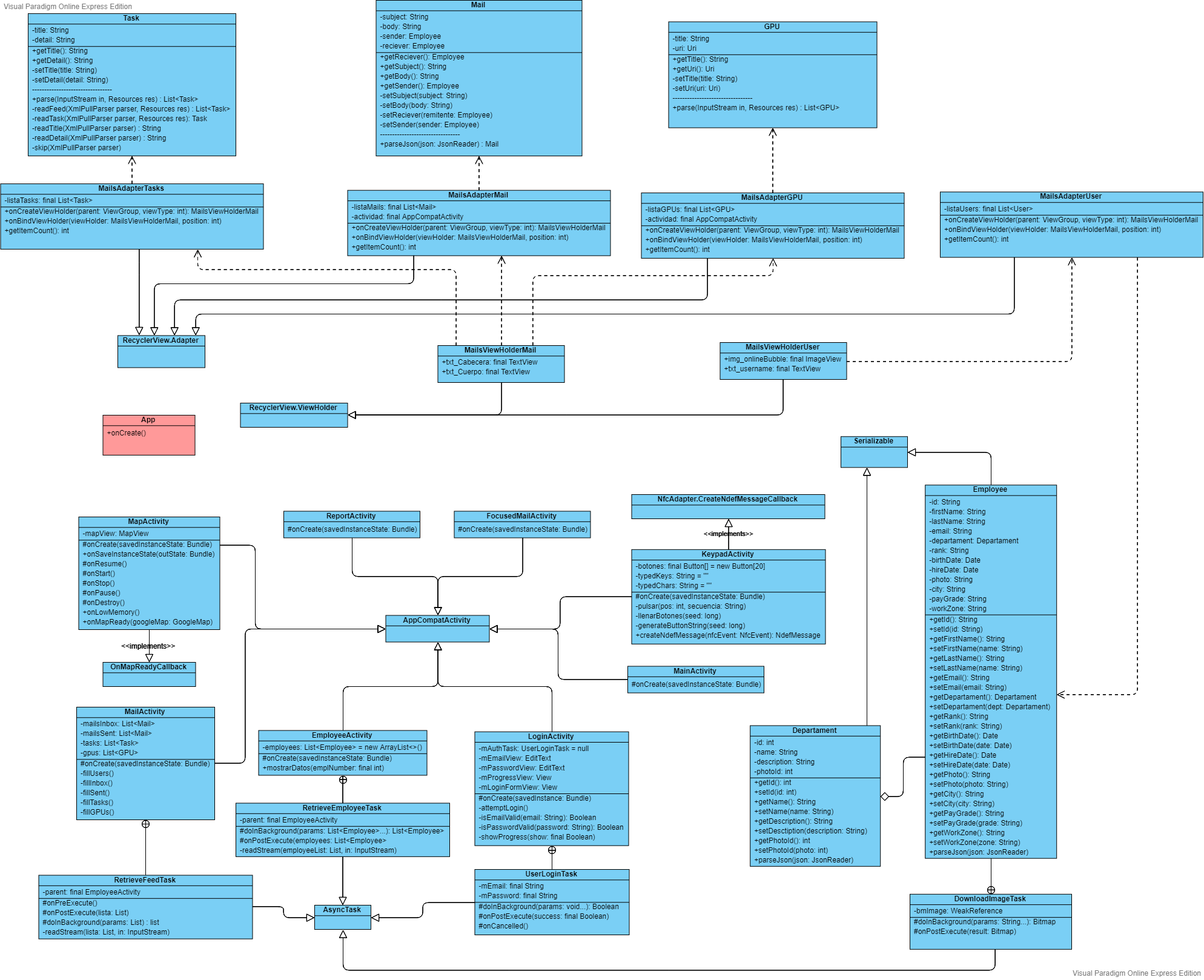


### Diagrama de despliegue



### Diagrama Entidad-Relacion

### 



# Ampliación y posibles mejoras

De todos esos requisitos, se ha omitido por falta de tiempo la gestión de almacén, y la vista de mapa y empleados se ha visto recortada, por lo que quedaría pendiente de una posible ampliación.

Más concretamente, hay tres grandes apartados que necesitan una ampliación:

* Vista empleados: Sería conveniente añadir una ventana intermedia, que filtre todos los empleados y los ordene por departamento, y permita buscar fácilmente.
* Vista mapa: Es necesario rediseñar toda esta actividad, pero debido a la complejidad de esta tarea (se requiere un layout personalizado), ha quedado excluida de los objetivos de cara a la presentación. Se mantiene como “demo” de lo que podría ser, pero no es funcional en su estado actual.
* Vista almacén: Uno de los objetivos propuestos era la creación de una actividad para la gestión de varios almacenes de la compañía, pero debido a la falta de experiencia en planificación de proyectos, y a la falta de tiempo en general, al final no se pudo implementar. Quedan como vista de cómo podría realizarse los esquemas hechos en papel de esta actividad.

Asimismo, hay otras mejoras que se podrían aplicar al proyecto:

* Utilización de un servidor de correo real en el back end: La implementación actual utiliza una base de datos SQL para almacenar y servir los correos, con los problemas que esto conlleva.
* Generación de teclas en la vista Keypad: La falta de conocimientos avanzados en matemáticas orientadas a probabilidades y funciones complejas ha limitado enormemente esta vista. Una revisión con ayuda de alguien curtido en estos campos ayudaría enormemente en la creación de un algoritmo mucho más fiable.
* Publicar la aplicación en google play, o al menos, firmarla con un certificado valido reconocido por Google: Posiblemente la ampliación más simple, el problema es meramente económico, la licencia para obtener las claves de firma supone un desembolso económico.
* Diferentes *temas*: Una de las ideas iniciales era que en función de una serie de parámetros, la aplicación pudiera cambiar de “tema”, afectando a la configuración de colores en todas las pantallas, pero esta idea finalmente fue descartada. Sin embargo, seguiría siendo una buena mejora que añadir.

# Conclusión

Este proyecto ha sido el culmen de todo lo aprendido en estos dos años de grado, tanto en el centro escolar como en el centro de trabajo, y pese a sus posibles fallos que posiblemente se vean expuestos en la presentación por el jurado ,es una aplicación de la cual me siento muy orgulloso.

Ha *absorbido* una cantidad de tiempo que mucha gente ni se plantearía el plasmar en un proyecto cuando este tiene unos requisitos en tiempo mucho más bajos 80 horas a la hora de escribir esto, aun faltando algunos apartados por terminar de pulir, y con la presentación aún pendiente de preparar.

Me ha permitido ver con otros ojos trabajos de otras personas que, cuando no se tiene la experiencia de trabajar en un proyecto de estas características, podría parecer algo normal, casi trivial. Creo que esa forma de ver algunas tareas ha afectado muy negativamente a las previsiones de tiempo en varias pantallas, encontrándome en el caso más llamativo habiendo gastado 4 veces el tiempo planeado, en la tarea de preparar los servicios web.

Pero por encima de todo me ha permitido plasmar en un proyecto lo que soy capaz de hacer con suficiente tiempo y ganas, y me ha ayudado a cercar un poco más mis expectativas laborales de cada a medio-largo plazo.

Por ultimo me veo en la necesidad de agradecer a todas las personas que se han visto involucradas de forma directa o indirecta en el proyecto:

* Profesores, de este ciclo y de los anteriores, que me han dado los conocimientos teóricos y prácticos para poder completar este proyecto, así como la ambición de siempre querer añadir algo nuevo, de siempre poder mejorar algo más, de siempre poder ver el problema desde un ángulo diferente.
* Amigos y compañeros, que siempre se han visto dispuestos a echar una mano en las fases de desarrollo donde han podido, con el como enfocar algunos aspectos de código, o con las pruebas en un montón de dispositivos físicos diferentes, de una forma mucho mas realista que con el uso de emuladores.
* Familia, fuente de apoyo y ayuda incluso mayor que la de amigos y compañeros, que siempre ha estado a mi lado para lo bueno y lo no tan bueno.

Gracias a todos.

# Bibliografía

El Lenguaje Unificado de Modelado (Grady Booch, James Rumbaugh, Ivar Jacobson)  
Addison-Wesley – Pearson (2ª edición)

Android Docs: Documentación para desarrolladores de apps  
<https://developer.android.com/docs>

Material Design: Build beautiful products, faster.  
<https://material.io/>

Stackoverflow: Learn, Share, Build

<https://stackoverflow.com/>

Redbean: The Power ORM  
<https://redbeanphp.com/index.php>

Silex: The PHP micro-framework based on the Symfony Components  
<https://silex.symfony.com/doc/2.0/>

Fandom: La fuente de confianza de fans en entretenimiento  
<https://www.fandom.com/>