



**ESCUELA SUPERIOR DE INGENIERÍA**

**INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS**

Desarrollo de aplicación Android para facilitar la superación de las barreras arquitectónicas para personas con movilidad reducida

Juan Manuel Hidalgo Navarro

24 de enero de 2018





## ESCUELA SUPERIOR DE INGENIERÍA

### GRADO EN INGENIERÍA INFORMÁTICA

Desarrollo de aplicación Android para la facilitación de accesibilidad de personas de movilidad reducida

- Departamento de Ingeniería Informática
- Director del proyecto: Pablo de la Torre Moreno
- Autor del proyecto: Juan Manuel Hidalgo Navarro

Cádiz, 24 de enero de 2018

Fdo: Juan Manuel Hidalgo Navarro



## **Agradecimientos**



## Resumen

Actualmente, resulta inconcebible un día a día sin su Smartphone. Debido al auge de estos dispositivos, cada vez más personas utilizan los múltiples servicios que ofrecen, facilitando tareas del día a día. Android es el sistema operativo para dispositivos móviles que ha experimentado mayor crecimiento en los últimos años, ocupando el 50 % de la población de Reino Unido.

Una aplicación Android es un software desarrollado para un dispositivo móvil o smartphone, el cual tienen la función de ayudar al usuario en la realización o facilitación de un trabajo concreto.

Este Trabajo de Fin de Grado, en adelante TFG, se centra en el desarrollo de una aplicación Android, la cual será un híbrido entre red social, página de valoraciones y aplicación de mapas, y tendrá como objetivo informar y facilitar la accesibilidad a personas que sufren de movilidad reducida y de esta manera puedan tener una rutina lo más normalizada posible.

Este proyecto, llamado con el acrónimo recursivo SACAPP (“SACAPP is an Accesible City App”), es una aplicación Android que informa de los lugares accesibles para personas discapacitadas y avisa a organizaciones registradas en la aplicación de las posibles visitas de estas personas discapacitadas para que así puedan llevar a cabo cualquier acción necesaria (como por ejemplo abrir la entrada para personas usuarios de sillas de ruedas o activar el elevador para que personas discapacitadas puedan superar unas escaleras), además de otras funcionalidades como permitir al usuario añadir lugares públicos que tengan algún interés hablando en temas de accesibilidad, de evaluar entidades públicas o privadas e incluso un chat donde permita hablar con otros usuarios de la aplicación.

Este proyecto nace del interés por conocer nuevas tecnologías y de adentrarse en el mundo tecnológico actual, donde las aplicaciones Android están a la orden del día, y las Smart Cities son proyectos muy innovadores que se están levantando con gran auge. Otro punto a tener en cuenta es la gran cantidad de información que una Smart City puede procesar, pudiendo considerar éstos como “Big Data”.

A parte de todo esto, otra motivación para realizar este proyecto es intentar evadir las barreras a las personas con discapacidad y poder ofrecerles un punto de información mediante el cual puedan facilitar su día a día.

**Palabras clave:** Android, Java, Android Studio, desarrollo, aplicación, app, node.js, javascript, mongodb, NOSQL, big data, smart city, accesibilidad, Firebase Cloud Messaging.





# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Alcance . . . . .	1
1.3. Objetivos del Sistema . . . . .	2
1.4. Glosario de Términos . . . . .	2
1.5. Estado del arte . . . . .	3
1.6. Organización del documento . . . . .	3
<b>2. Desarrollo del proyecto</b>	<b>5</b>
2.1. Metodología de desarrollo . . . . .	5
2.2. Planificación del proyecto . . . . .	6
2.2.1. Planificación de actividades . . . . .	6
2.2.2. Estimación temporal y calendario detallado . . . . .	8
2.2.3. Diagrama de Gantt . . . . .	9
2.3. Organización . . . . .	9
2.3.1. Roles y responsabilidades . . . . .	9
2.3.2. Recursos y herramientas . . . . .	9
2.4. Riesgos . . . . .	11
<b>3. Requisitos del Sistema</b>	<b>13</b>
3.1. Situación Actual . . . . .	13
3.1.1. Entorno Tecnológico . . . . .	13
3.1.2. Fortalezas y Debilidades . . . . .	14
3.2. Objetivos del Sistema . . . . .	14
3.3. Catálogo de requisitos . . . . .	15
3.3.1. Requisitos funcionales . . . . .	15
3.3.2. Requisitos no funcionales . . . . .	16
3.3.3. Reglas de negocio . . . . .	17
3.3.4. Requisitos de información . . . . .	17
3.4. Alternativas de Solución . . . . .	17
3.4.1. Android Studio . . . . .	18
3.4.2. Eclipse . . . . .	18
3.4.3. IntelliJ IDEA . . . . .	18
3.4.4. LibGdx . . . . .	18
3.5. Solución Propuesta . . . . .	18

<b>4. Análisis del Sistema</b>	<b>19</b>
4.1. Modelo Conceptual . . . . .	19
4.2. Modelo de Casos de Uso . . . . .	19
4.2.1. Actores . . . . .	20
4.2.2. Descripción de los casos de uso . . . . .	22
<b>5. Diseño del Sistema</b>	<b>29</b>
5.1. Arquitectura del Sistema . . . . .	29
5.1.1. Arquitectura Física . . . . .	29
5.1.2. Arquitectura Lógica . . . . .	29
5.2. Patrones de diseño . . . . .	31
5.3. Diseño de Interfaz de la aplicación . . . . .	32
5.3.1. Navegabilidad . . . . .	32
5.3.2. Diseño detallado de la interfaz . . . . .	36
<b>Bibliografía y referencias</b>	<b>37</b>
<b>Software usado</b>	<b>37</b>
<b>Instalación de L<sup>A</sup>T<sub>E</sub>X</b>	<b>40</b>
<b>GNU Free Documentation License</b>	<b>45</b>
1. APPLICABILITY AND DEFINITIONS . . . . .	45
2. VERBATIM COPYING . . . . .	46
3. COPYING IN QUANTITY . . . . .	46
4. MODIFICATIONS . . . . .	47
5. COMBINING DOCUMENTS . . . . .	48
6. COLLECTIONS OF DOCUMENTS . . . . .	49
7. AGGREGATION WITH INDEPENDENT WORKS . . . . .	49
8. TRANSLATION . . . . .	49
9. TERMINATION . . . . .	49
10. FUTURE REVISIONS OF THIS LICENSE . . . . .	50
11. RELICENSING . . . . .	50
ADDENDUM: How to use this License for your documents . . . . .	50

# Indice de figuras

2.1. Ciclo Scrum . . . . .	5
2.2. Diagrama de Gantt . . . . .	10
4.1. Diagrama conceptual de clases . . . . .	19
4.2. Diagrama de casos de uso . . . . .	21
5.1. Patrón de diseño Modelo-Vista-Vista de Modelo . . . . .	33
5.2. Acceso al registro de usuarios . . . . .	33
5.3. Acceso al registro de entidades . . . . .	34
5.4. Acceso al menú principal de usuarios . . . . .	34
5.5. Navegación por el menú principal de usuario . . . . .	35
5.6. Navegación dentro del Mapa . . . . .	35
5.7. Ventana principal de entidad . . . . .	36
5.8. Ventana de visualización de notificaciones de visitas . . . . .	36
5.9. Emacs + AucTEX . . . . .	41
5.10. Interfaz de Dia . . . . .	42



# Indice de tablas

2.1. Análisis previo y planificación . . . . .	6
2.2. Análisis de requisitos . . . . .	7
2.3. Diseño . . . . .	7
2.4. Implementación y pruebas . . . . .	8
2.5. Finalización . . . . .	8



# Capítulo 1

## Introducción

A continuación, se describe la motivación del presente proyecto y su alcance. También se incluye un glosario de términos y la organización del resto de la presente documentación.

### 1.1. Motivación

En la actualidad, las personas con discapacidad deben de superar muchas barreras, tanto arquitectónicas como sociales, a lo largo de su día a día. El hecho de tener que enfrentarse continuamente a estas barreras hace que no puedan llevar a cabo una vida cotidiana normalizada.

Una de las razones por las que esta aplicación ha sido desarrollada viene condicionada simplemente por la ayuda social que les podría aportar a estas personas que sufren de movilidad reducida, e intentar así incrementar su calidad de vida y facilitar el conocimiento de las instalaciones habilitadas a sus condiciones y capacidades.

Este proyecto también está motivado por la inquietud de conocer nuevas áreas de conocimiento dentro de nuestro sector, por eso mismo se han usado tecnologías que hoy en día se aplican en un gran porcentaje en el mundo laboral, y de esta manera, adquirir o complementar los conocimientos necesarios para poder darle un uso para poder usar estas herramientas en proyectos reales.

Los datos recogidos del año 2016 indican que Android aumentó su participación en un 2 %, por lo que actualmente representa un 86,1 % del mercado de smartphones, lo que nos indica que es una tecnología muy extendida en nuestra sociedad, y por consecuencia, conocer y controlar esta tecnología nos podría abrir muchas puertas en el mundo laboral. Según un estudio publicado en The Guardian, en Reino Unido la mitad de la población tiene un smartphone [1]

### 1.2. Alcance

El TFG es una aplicación Android orientado a un publico que sufre de discapacidades, en concreto, que sufren de movilidad reducida. Dentro de este público, se acogen también a personas mayores las cuales, a causa de la edad, han ido perdiendo capacidades motrices.

Teniendo en cuenta esto, la aplicación debe de tener una interfaz amigable y simple, ya que será usada por personas de todas las edades. Por supuesto, estos puntos no pueden hacer que su funcionalidad se

vea comprometida.

Todo este desarrollo está orientado a la ayuda en el día a día estas personas. El principal objetivo es que mediante la aplicación, puedan ir a cualquier lugar de la ciudad, sabiendo en todo momento de qué herramientas de accesibilidad disponen.

También se podría plantear la idea de hacer una aplicación accesible para usuarios con discapacidades auditivas o visuales. En éste segundo caso, la aplicación debería ser adaptada para que una persona con problemas de visión pueda usarla sin que su discapacidad se lo impida.

### 1.3. Objetivos del Sistema

Esta sección contiene la especificación de los objetivos o requisitos generales del sistema. Los objetivos principales son los siguientes y estarán especificados en las siguientes figuras:

- **OBJ-01** Ofrecer una aplicación móvil, donde con un simple vistazo se obtenga información útil con respecto a la entidad y la superación de barreras arquitectónicas.
- **OBJ-02** Diseñar menús intuitivos donde sea fácil para el usuario localizar su destino.
- **OBJ-03** Implementar un sistema de comentarios y valoraciones para que los propios usuarios puedan verificar una entidad.
- **OBJ-04** Incluir un sistema de registro de entidades públicas, como parques y monumentos, creado y valorado por los propios usuarios.
- **OBJ-05** Diseñar un sistema de aviso de visita a una entidad que será visitada, con un segundo aviso posterior cuando el usuario se encuentre cerca.

### 1.4. Glosario de Términos

Esta sección contiene una lista ordenada alfabéticamente de los principales términos, acrónimos y abreviaturas específicos del dominio del problema:

- **Accesibilidad:** Combinación de elementos constructivos y operativos que permiten a cualquier persona con discapacidad, entrar, desplazarse, salir, orientarse y comunicarse con el uso seguro, autónomo y cómodo en los espacios construidos, el mobiliario y equipo, el transporte, la información y las comunicaciones.
- **Barreras arquitectónicas:** aquellos obstáculos físicos que impiden que determinados grupos de población puedan llegar, acceder o moverse por un espacio urbano, un edificio o una parte de él.
- **Entidad:** Es el nombre que recibe la representación de los lugares los cuales los usuarios podrán registrar (solo en caso de ser público), valorar, o marcar para visitar. Se dividen en dos subconjuntos:
  - **Entidad privada:** Es una organización privada, que normalmente se corresponderá con una empresa. Estas entidades serán creadas como usuarios de la aplicación y gestionadas por un trabajador o persona encargada de dicha entidad. Las personas registradas en la aplicación podrán comentar y evaluar dichas entidades.



- **Entidad pública:** Es un lugar público, el cual no implica ningún tipo de instalaciones ni personal que lo gestione. Un ejemplo podría ser parques, plazas o localizaciones importantes dentro de una ciudad. Estas entidades serán creadas, comentadas y evaluadas por los usuarios de la aplicación.

## 1.5. Estado del arte

En esta sección trataremos de ver la situación de las tecnologías similares a la desarrollada en este TFG. Dado que en este proyecto se lleva a cabo el desarrollo de una aplicación android para la ayuda de personas con movilidad reducida, vamos a investigar que aplicaciones encontramos hoy en día referente a la misma temática.

- **DisCar:** Esta aplicación se creó con el objetivo de ayudar a encontrar aparcamiento habilitados para personas con movilidad reducida, para que así pudiesen llegar más cerca de su destino. Esta aplicación permite añadir más aparcamientos reservados al mapa, así como solicitarle al ayuntamiento el aumento necesario de aparcamientos en un lugar determinado.
- **Crosswalk:** Un gran problema que tienen las personas con movilidad reducida es cruzar varios carriles de una vez, ya que el tiempo es reducido y debe continuar hasta que atravesase todos los carriles, esto supone un peligro para su integridad física. Pero sin embargo, se ha implantado en algunas ciudades y para algunos ciudadanos la aplicación Crosswalk, que brinda a los peatones con movilidad reducida un tiempo de cruce adicional.
- **DisplaceTool:** Gracias a esta herramienta podrás localizar en tu ciudad todas las ubicaciones y sitios que puedes acceder fácilmente con silla de ruedas y no tengas ninguna barrera arquitectónica.

Viendo el panorama actual y conociendo la definición de nuestra aplicación, podríamos decir que **DisplaceTool** tiene una funcionalidad muy similar a la de nuestro proyecto. Pero nuestro proyecto ofrece una funcionalidad extra, que consiste en la posibilidad de visitar una empresa y en el aviso automático cuando el usuario está cerca.

## 1.6. Organización del documento

Esta sección contiene una descripción de los contenidos de la presente memoria. La primera parte de la memoria del TFG consta de una introducción y una planificación del proyecto.

Como primer capítulo y a modo de resumen, tenemos una breve descripción del contexto del proyecto, motivación de éste, así como el alcance previsto.

La planificación incluirá los plazos, los entregables, los recursos y el método de ingeniería software que se empleará.

Como segunda parte del documento, tendremos el documento de desarrollo siguiendo los metodos de ingeniería software asociados. Con apartado de requisitos, análisis, diseño, construcción y pruebas.

Para finalizar, tendremos una última sección con la recogida de conclusiones y manuales para el manejo de la aplicación.

## Capítulo 2

# Desarrollo del proyecto

En esta sección se describen los aspectos referentes a la gestión del proyecto. Estos aspectos abarcan la metodología, organización, costes, planificación, riesgos y aseguramiento de la calidad.

### 2.1. Metodología de desarrollo

A pesar de no tener una vida laboral muy larga, los conocimientos y experiencias adquiridos en otros proyectos me ha llevado a elegir la metodología de desarrollo Scrum.

Con esta metodología damos énfasis a la obtención de un prototipo funcional. Gracias a esto, podemos terminar de definir aquellas funcionalidades o características las cuales no estaban del todo definidas, o incluso evaluar si realmente dicha característica nos ayudaría o no a alcanzar nuestra funcionalidad final, o si en cambio esta debe de ser modificada.

Esta metodología proporciona a cada iteración un prototipo funcional de la aplicación, el cual es totalmente escalable y muy apto a la hora de incluir nuevas funcionalidades [2].

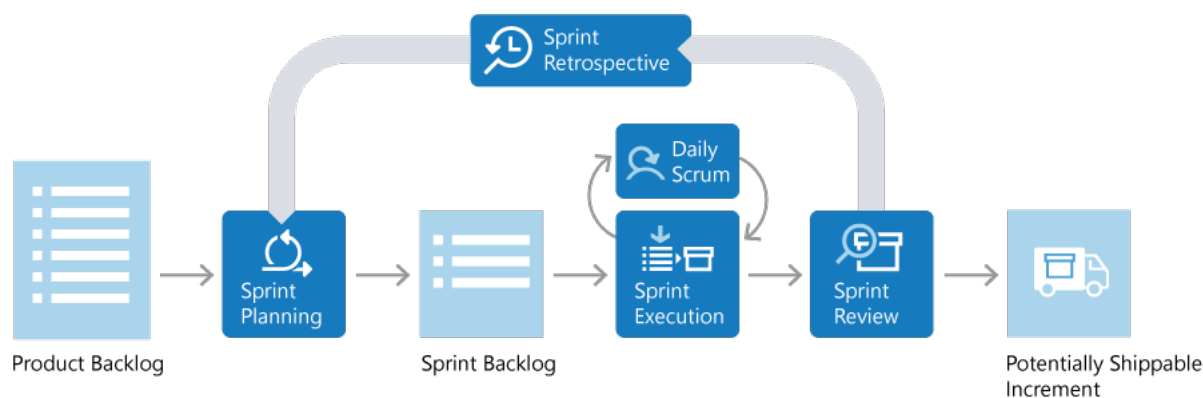


Figura 2.1: Ciclo Scrum

El proceso Scrum se basa en ciclos de desarrollos cortos en los que los requisitos pueden ser variables y el proceso está orientado al desarrollo de prototipos incrementales y funcionales.

En la parte inicial de nuestro ciclo Scrum, se definirá una lista de requisitos priorizados. Este proceso se denomina *Product Backlog*, y se describen los requisitos en un lenguaje no técnico.

Este es un caso especial, por lo que al ser un proyecto propio, el rol de *Product Owner* lo toma el desarrollador principal, el cual presenta al equipo de desarrollo el *Product Backlog* en una fase llamada *Sprint* y en esta presentación, el equipo determina que funcionalidades serán desarrolladas en este *Sprint* determinando sus fechas de entregas (las cuales son convenientes que no superen 4 semanas). Esta reunión se le llama el *Sprint Planning*.

Un *Sprint* queda definido por el equipo y debe de tener las siguientes características:

- Una descripción.
- Una fecha de finalización.
- Las funcionalidades que se van a desarrollar.
- La asignación de las funcionalidades a desarrollar.

Durante cada *Sprint*, cada día se debe de realizar un *Daily Scrum*, es decir, reuniones diarias las cuales cada integrante del equipo expone lo que realizó el día anterior, las tareas que va a realizar hoy y todos los problemas que ha tenido en el desarrollo en caso de que existiesen.

Al final de cada *Sprint*, el equipo presenta todas las funcionalidades realizadas y posteriormente se analizan para ver si algunas de estas funcionalidades pueden ser mejoradas o perfeccionadas.

Dada la naturaleza de este proyecto y ya que ha sido desarrollado por un solo integrante, se ha tenido que emular el proceso de desarrollo SCRUM se han adoptado diferentes roles y se han adaptado los ciclos de desarrollo a la metodología definida.

## 2.2. Planificación del proyecto

En los siguientes apartados se desarrolla la estimación temporal y definición de las fechas de entrega de cada apartado, además del desarrollo de la planificación detallada, utilizando un diagrama de Gantt.

### 2.2.1. Planificación de actividades

#### Análisis previo y planificación

En este punto se ha abordado la visión global del proyecto, comprendida en establecer los puntos iniciales necesarios para crear una planificación de tareas en función de las actividades a realizar.

Actividad	Descripción
Diseño conceptual de la aplicación	Diseño de las dinámicas y funcionamiento de la aplicación
Tecnologías a usar	Elección de herramientas para llevar a cabo el desarrollo
Análisis de riesgos	Enumeración de los posibles riesgos del proyecto
Planificación del proyecto	Fijación de plazos de entrega para cada funcionalidad del sistema
Documentación	Desarrollo del plan de trabajo

Tabla 2.1: Análisis previo y planificación

## Análisis de requisitos

Para abordar este tema haremos un estudio sobre las necesidades que se deben de cubrir para que la aplicación tenga el funcionamiento deseado, tanto desde el punto de vista funcional, como el punto de vista arquitectónico. Se ha diseñado un modelo troncal de las clases para posteriormente implementar en base a estas y los modelos que seguirá para su posterior almacenamiento en base de datos.

Actividad	Descripción
Requisitos funcionales	Definición de casos de uso
Diseño de estructura de la aplicación	Diagrama básico de clases
Diseño de la base de datos	Diseño del modelo conceptual
Diseño del servicio web	Plantear las funcionalidades y rutas del servicio web

Tabla 2.2: Análisis de requisitos

## Diseño

Como el proyecto a desarrollar es una aplicación Android, se ha realizado una la recopilación conceptual del diseño de todas las pantallas que intervienen en el, además con los diseños a aplicar (colores, fuentes, etc) y la lógica de la aplicación.

Actividad	Descripción
Definición de los tipos de usuarios	Diferenciación entre los tipos de usuarios que van a participar en la aplicación, así como los contexto en los que participan
Diseño conceptual	Un escenario de uso describe desde el punto de vista del usuario como será utilizado un producto determinado (es este caso una aplicación Android) en un escenario determinado.
Prototipo de las pantallas (mockups)	Se han creado recreaciones de las pantallas de las cuales la aplicación hará uso. Mockups y prototipos de alta fiabilidad
Redacción del documento	Creación del documento de diseño

Tabla 2.3: Diseño

## Implementación y pruebas

Hemos desarrollado la implementación de nuestra aplicación partiendo de todas las definiciones de los apartados anteriores y realizado las verificaciones oportunas para asegurar el funcionamiento de nuestra aplicación.

Actividad	Descripción
Implementación de las pantallas y la navegación entre ellas	Implementación de funcionalidad y diseño de las pantallas además de toda la conexión entre ellas.
Desarrollo de la API Rest	Implementación del servicio web que alimentará a la aplicación.
Implementación de la lógica	Desarrollo de la lógica que seguirá la aplicación
Versión incremental de la aplicación	En este punto se dispone de la primera versión iterativa de la aplicación.
Pruebas e incrementos en el Sprint	Incrementos en funcionalidades, pruebas y corrección de errores
Preparación final del proyecto	Pruebas de integración y despliegue

Tabla 2.4: Implementación y pruebas

**Finalización** El desarrollo de esta parte del proyecto consistirá en la fase de implantación y despliegue de la aplicación.

Actividad	Descripción
Redacción final de la memoria	Finalizar los apartados pendientes de ésta
Implantación del sistema en un entorno de producción	Montar la aplicación en un entorno ajeno al de desarrollo
Desplegar servicio web y la base de datos en un servidor de datos	Montar el servicio web en un servidor de acceso externo con acceso a la base de datos
Demostración de funcionamiento final	Montaje del video final y demo para presentación del proyecto
Entrega final	Entrega del proyecto completo y documentación

Tabla 2.5: Finalización

### 2.2.2. Estimación temporal y calendario detallado

El proyecto tiene una estimación temporal de 3 meses y medio aproximadamente, con una sola persona desarrollando éste mismo. Comienza en Octubre de 2017 y finaliza a finales de Enero de 2018.

En el siguiente apartado tenemos el desglose de las actividades que se deben realizar con una estimación temporal por días.

- Análisis previo y planificación: 21 días
- Análisis de requisitos: 13 días
- Implementación y pruebas: 53 días
- Finalización: 17 días

Teniendo en cuenta esto, sería en total de 104 días de trabajo, y teniendo en cuenta que el alumno tiene una jornada laboral de 8 horas diarias y que solo podría dedicarle unas 5 horas diarias, el proyecto pasaría a tener unas 520 horas de desarrollo.

### 2.2.3. Diagrama de Gantt

A continuación, en la figura 2.2, se adjunta el diagrama de Gantt correspondiente a la planificación del proyecto.

## 2.3. Organización

### 2.3.1. Roles y responsabilidades

Debido a la naturaleza de este proyecto, todos los roles implicados en un proyecto software han sido representados por una misma persona, para así interpretar diferentes roles en los distintos momentos del desarrollo para poder lograr un producto de calidad y siguiendo los procesos de ingeniería, lo cual ofrece un valor añadido al proyecto.

#### Director del proyecto

La responsabilidad fundamental de este rol será coordinar y resolver todos los conflictos que puedan aparecer en un equipo de trabajo, siendo además el responsable de todos los planes de proyecto, y será el encargado de organizar, gestionar y cumplir el plan de proyecto. También supervisará a los miembros de un equipo.

**Desarrollador** La responsabilidad de este rol será, realizar la implementación del sistema. Además, realizará la integración del sistema y su liberación. Una vez el sistema esté liberado, será el encargado de ofrecer al usuario final una serie de pautas para que pueda utilizar el sistema a satisfacción. Será el encargado de liberar los correspondientes manuales de uso (usuario y administrador) del producto software. Además, en nuestro caso particular, serán definidos como encargados de llevar a cabo la formación al cliente.

#### Analista

La responsabilidad de este rol será la de realizar la definición y análisis de requisitos y el diseño del sistema a desarrollar según el proyecto dado.

Es el encargado de convertir estos requisitos a requisitos funcionales y no funcionales. Además diseñará el plan de pruebas del proyecto.

#### Tester

La responsabilidad de este rol recae en implementar el plan de pruebas asignado al proyecto y realizar todas las pruebas según la especificación a satisfacer.

#### Diseñador gráfico

La responsabilidad de este rol será la de acompañar al desarrollador encargado de la implementación del sistema realizando el diseño gráfico del sistema. Ofrecen un punto de diseño artístico y de gran calidad visual para los sistemas a desarrollar.

### 2.3.2. Recursos y herramientas

Se hará uso de sistema operativo Linux, en este caso la distribución Ubuntu 16.10 LTS como sistema operativo principal.

La estación de trabajo será un portátil MSI CX62. Además, para las pruebas, se usará un SmartPhone Motorola Moto G2, un Samsung Galaxy s7 Edge y una emulación de un Nexus 5. Como servidor de

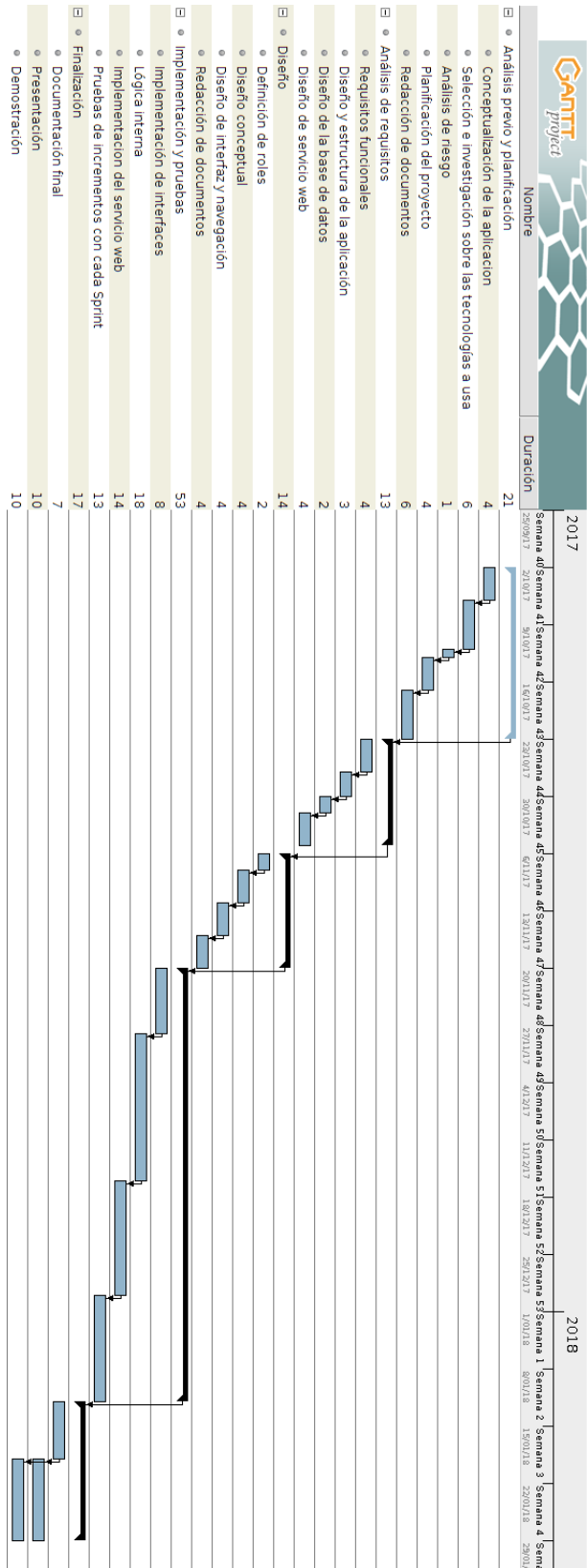


Figura 2.2: Diagrama de Gantt



datos, se usará el mismo portátil el cual desplegará la API Rest y la base de datos.

Se dispondrá de un servidor que aloje el software desarrollado con el repositorio GIT, en este caso se hará uso de la plataforma GitHub como servidor principal de control de versiones.

Para la realización del proyecto se han utilizado las siguientes herramientas:

- Diseño:
  - Gimp 2.8
  - Ninja Mockups
- Desarrollo
  - Android Studio 2.3.3
  - Android SDK
  - Sublimetext
  - Node.js
  - MongoDB
  - Firebase Cloud Messaging
  - Control de versiones Git mediante GitHub
- Redacción de documentos:
  - $\text{\LaTeX}$

## 2.4. Riesgos

En esta parte del documento enumeraremos los riesgos que conlleva el proyecto en orden de prioridad:

1. Se parte de una experiencia nula en todas las tecnologías usadas en el proyecto ni en desarrollo de aplicaciones móviles. Esto ha sido así debido al deseo de aprender y dominar tecnologías que tienen bastante difusión hoy en día. No obstante, existe la posibilidad de dedicar más tiempo el proceso de desarrollo del proyecto para aprender y consultar documentación para lograr los objetivos establecidos.
2. Existe un riesgo a nivel artístico, ya que el desarrollo de una interfaz de usuario amigable y con colores agradables y bien combinados es una tarea complicada de la cual se encargan diseñadores gráficos. A pesar de esto, es un reto interesante, ya que en un futuro es importante saber desenvolverse por estos ámbitos para casos en la vida laboral real.
3. La lógica interna de la aplicación es otro de los riesgos a tener en cuenta, ya que se intenta desarrollar una aplicación que se actualice y aplique datos en tiempo real.



## Capítulo 3

# Requisitos del Sistema

En esta sección se detalla la situación que origina el desarrollo o mejora de un sistema informático. Luego presentaremos los hitos u objetivos del sistema tanto como sus requisitos. Finalmente terminaremos con las alternativas tecnológicas del proyecto y el análisis de la brecha de requisitos planteados, junto a su solución base seleccionada.

### 3.1. Situación Actual

Partimos de cero, en el desarrollo de un nuevo proyecto, por lo tanto más que situación actual del sistema, podremos hablar de la situación actual del mercado de aplicaciones Android de ayuda para personas con movilidad reducida, o incluso generalizando, para personas que sufran de cualquier tipo de discapacidad.

Una aplicación Android es un software creado para el entretenimiento o para ayudar y/o facilitar información al usuario en su día a día. Dicho esto, podríamos identificar dos categorías dentro de las aplicaciones para smartphones, entretenimiento y herramientas de ayuda. Nuestra aplicación entraría en el conjunto de aplicaciones de ayuda al usuario.

En este caso nos centraremos en el tipo de aplicaciones Android destinadas para la facilitación del día a día de los usuarios. En nuestro caso, siendo una aplicación para personas de movilidad reducida acotamos mucho el rango del publico, pero este es un entorno en el cual podemos ofrecer una gran ayuda al usuario.

#### 3.1.1. Entorno Tecnológico

Android es un sistema operativo con núcleo monolítico basado en el kernel de Linux version 2.6 que fue lanzado al mercado el 23 de septiembre de 2008. Diseñado principalmente para dispositivos móviles con pantalla táctil, como pueden ser Smartphones o tabletas. Esta disponible gratuitamente para un uso comercial o no comercial. Soporta distintas estructuras de procesador, entre ellas ARM, x86, MIPS y IBM POWER, no obstante la arquitectura utilizada mayoritariamente son los procesadores ARM. Esta publicado bajo la licencia Apache 2.0 y GNU GPL2. [3]

Desde su lanzamiento, compitió con distintas plataformas ya existentes en aquel momento, iPhone OS conocido actualmente como iOS, Blackberry OS o Windows Phone 7. A pesar de esto, su crecimiento fue exponencial a nivel mundial, popularizandose entre todos los sistemas operativos disponibles para smartphones. A nivel mundial alcanzó una cuota de mercado del 50.9 %, esto represento más del doble

que el sistema operativo iOS de Apple.

Para realizar la implementación de la aplicación que soportara este sistema operativo, se ha decidido usar el entorno de desarrollo proporcionado por Google para ello, Android Studio.

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA . Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android

### 3.1.2. Fortalezas y Debilidades

Esta sección está dedicada a evaluar los puntos positivos y negativos del negocio actual de la organización para la que se va a desarrollar el sistema software.

En la actualidad, existe un amplio abanico de aplicaciones Android pero esto no resta que sea una gran oportunidad de abarcar un contexto poco explorado con la facilidad de distribución a las personas que nos proporciona Android.

## 3.2. Objetivos del Sistema

Esta sección contiene la especificación de los objetivos o requisitos generales del sistema. Los objetivos principales son los siguientes y estarán especificados en las siguientes figuras:

- **OBJ-01** Ofrecer una aplicación móvil, donde con un simple vistazo se obtenga información útil con respecto a la entidad y la superación de barreras arquitectónicas.
- **OBJ-02** Diseñar menús intuitivos donde sea fácil para el usuario localizar su destino.
- **OBJ-03** Implementar un sistema de comentarios y valoraciones para que los propios usuarios puedan verificar una entidad.
- **OBJ-04** Incluir un sistema de registro de entidades públicas, como parques y monumentos, creado y valorado por los propios usuarios.
- **OBJ-05** Diseñar un sistema de aviso de visita a una entidad que será visitada, con un segundo aviso posterior cuando el usuario se encuentre cerca.

<b>OBJ-01</b>	Ofrecer una aplicación móvil, donde con un simple vistazo se obtenga información útil con respecto a la entidad y la superación de barreras arquitectónicas.
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripcion</b>	La aplicación debe de ser portable, es decir, que pueda ser usada en un dispositivo móvil con conexión y localización. Debe de ser intuitiva y de interfaz amigable y debe poder mostrar todos los datos necesarios para que el usuario supere las barreras arquitectónicas que se pueda encontrar
<b>Prioridad</b>	Alta

<b>OBJ-02</b>	Diseñar menús intuitivos donde sea fácil para el usuario localizar su destino
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripcion</b>	Una funcionalidad vital de esta aplicación es que el usuario pueda situar en un mapa las entidades a las que desea ir. Además deberá conocer su dirección y poder acceder directamente a la herramienta Google Map para poder llegar hasta su destino.
<b>Prioridad</b>	Alta

<b>OBJ-03</b>	Implementar un sistema de comentarios y valoraciones para que los propios usuarios puedan verificar una entidad
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripcion</b>	Es importante que los usuarios puedan dejar un comentario y valorar el servicio de las entidades o la veracidad sobre sus recursos accesibles para advertir sobre la información expuesta a próximos usuarios.
<b>Prioridad</b>	Media

<b>OBJ-04</b>	Incluir un sistema de registro de entidades públicas, como parques y monumentos, creado y valorado por los propios usuarios
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripcion</b>	A diferencia de las entidades tipo 'empresas', nuestro sistema abarca otro tipo de entidades, como las entidades públicas, concepto que hace referencia a parques, monumentos o zonas de interés que no están gestionadas directamente por nadie. Para poder incluir en nuestro sistema este tipo de entidades, pondremos a disposición del usuario herramientas necesarias para que pueda realizar un registro de estos datos. Dicha entidad también tendrá recursos accesibles y podrá ser comentada y valorada.
<b>Prioridad</b>	Media

<b>OBJ-05</b>	Diseñar un sistema de aviso de visita a una entidad que será visitada, con un segundo aviso posterior cuando el usuario se encuentre cerca.
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripcion</b>	Uno de los funcionamiento fundamentales de la aplicación, es el aviso automático de cuando el usuario está llegando a la entidad. Para ello hará falta implementar un sistema, en el cual el usuario previamente marque para visitar una entidad, y cuando este cerca, el sistema mande una notificación a la entidad para que pueda ir iniciando los servicios necesarios para poder recibir a esta persona
<b>Prioridad</b>	Alta

### 3.3. Catálogo de requisitos

Esta sección contiene la descripción del conjunto de requisitos específicos del sistema a desarrollar.

#### 3.3.1. Requisitos funcionales

Descripción completa de la funcionalidad que debe ofrecer el sistema.

- **RQ-01** Los usuarios podrán darse de alta indicando el motivo de su movilidad reducida.
- **RQ-02** La entidad se dará de alta indicando los recursos accesibles y su localización.

- **RQ-03** El usuario podrá consultar un mapa donde se muestran las entidades.
- **RQ-04** El usuario podrá registrar comentarios y valoraciones de las entidades.
- **RQ-05** El usuario podrá consultar comentarios de otros usuarios
- **RQ-06** El usuario podrá registrar visitas a una entidad
- **RQ-07** El usuario podrá eliminar las visitas registradas
- **RQ-08** El usuario/entidad podrá editar sus datos.
- **RQ-09** Notificar cuando un usuario visitante está cerca
- **RQ-10** El usuario podrá agregar una entidad pública
- **RQ-11** La entidad podrá consultar sus comentarios.
- **RQ-12** La entidad podrá consultar sus próximas visitas.

### **3.3.2. Requisitos no funcionales**

Descripción de otros requisitos (relacionados con la calidad del software) que el sistema deberá satisfacer.

- **NFR-01** El software debe de proporcionar la funcionalidad necesaria para satisfacer las necesidades propuestas.
- **NFR-02** El software debe de tener tiempos de respuesta bajos y no sobrecargar los recursos del sistema.
- **NFR-03** El producto debe de tener facilidad para ser entendido, aprendido y usado.
- **NFR-04** El producto debe de responder bien ante unas condiciones determinadas y durante un tiempo determinado.
- **NFR-05** La aplicación debe de ser capaz de proteger la información y los datos y prevenir accesos y/o modificaciones no autorizados.
- **NFR-06** El software debe de tener gran capacidad para ser modificado o ampliado sin que su estructura cambie.
- **NFR-07** Nuestro sistema debe de poder correr en entornos hardware diferente.
- **NFR-08** Debe de tener capacidad para que dos o mas sistemas o componentes puedan intercambiar información y/o llevar sus funciones requeridas.
- **NFR-09** Debe de ser compatible con sistema Android cuya versión sea igual o superior a la 5.0 (API 21).
- **NFR-10** El sistema deberá estar organizado para minimizar los errores del usuario.

### 3.3.3. Reglas de negocio

Definiremos en esta sección las reglas o políticas que se han decidido implantar para desarrollar el sistema.

- **RN-01** El usuario podrá comentar en una entidad cuantas veces desee.
- **RN-02** El usuario podrá marcar para visitar una misma empresa solo una vez al día

### 3.3.4. Requisitos de información

En esta sección se describen los requisitos de gestión de información (datos) que el sistema debe gestionar. El sistema debe gestionar los datos referentes a las entidades y al registro de visitas de los usuarios.

Para ello debemos gestionar los siguientes requisitos de información:

IRQ-01	Usuario		
Atributo	Tipo	Obligatorio	Único
Nombre	Cadena texto	Si	No
Apellidos	Cadena texto	Si	No
Email	Cadena texto	Si	Si
Contraseña	Cadena texto	Si	Si
Tipo reducción de movilidad	Cadena texto	Si	No

IRQ-02	Entidad		
Atributo	Tipo	Obligatorio	Único
Nombre de la Entidad	Cadena texto	Si	No
Página web	Cadena texto	Si	No
Email	Cadena texto	Si	Si
Contraseña	Cadena texto	Si	Si
Recursos accesibles	Conjunto de Cadena texto	Si	No

- **IRQ-01** Se debe almacenar los datos del usuario.
- **IRQ-02** Debemos de guardar los datos de la entidad junto a las accesibilidades de las que ésta dispone.
- **IRQ-03** Se deben almacenar las solicitudes de visitas de los usuarios a las entidades.
- **IRQ-04** Se debe almacenar los comentarios de los usuarios junto a su valoración

## 3.4. Alternativas de Solución

En esta sección, se ofrece un estudio del arte de las diferentes alternativas tecnológicas que permitan satisfacer los requerimientos del sistema, para luego seleccionar la herramienta o conjunto de herramientas que utilizaremos como base para el software a desarrollar.

Existen diferentes tecnologías disponibles para desarrollar aplicaciones Android hoy en día.

### **3.4.1. Android Studio**

Android Studio es un entorno de desarrollo integrado (IDE), basado en IntelliJ IDEA de la compañía JetBrains, que proporciona varias mejoras con respecto al plugin ADT (Android Developer Tools) para Eclipse. Android Studio utiliza una licencia de software libre Apache 2.0, está programado en Java y es multiplataforma.

### **3.4.2. Eclipse**

Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

### **3.4.3. IntelliJ IDEA**

IntelliJ IDEA es un ambiente de desarrollo integrado(IDE) para el desarrollo de programas informáticos

### **3.4.4. LibGdx**

Libgdx es un framework java de desarrollo de juegos libre que provee una API unificada que trabaja sobre una serie de plataformas soportadas.

El framework provee de un entorno desarrollo para un prototipado rápido e iteraciones rápidas. En lugar de desplegar en Android/iOS/Javascript después de cada cambio en el código, se pueden ejecutar y depurar en tu propio escritorio de forma nativa. Libgdx intenta no ser una única solución. No fuerza un diseño específico.

## **3.5. Solución Propuesta**

Se ha optado por la selección de Android Studio, ya que para desarrollar una aplicación para sistema Android, este es el IDE de desarrollo oficial y nos proporciona muchas ayudas y facilidades que no nos proporcionan el resto de soluciones, como por ejemplo las facilidades que brinda a la hora de crear los diferentes recursos o clases, las mejoras en la creación de vistas, el emulador que nos proporciona, etc.

Esta opción permite un desarrollo en Java que es uno de los lenguajes enseñados en la carrera y uno de los lenguajes de programación más extendidos.



## Capítulo 4

# Análisis del Sistema

Esta sección cubre el análisis del sistema de información a desarrollar, haciendo uso del lenguaje de modelado UML.

### 4.1. Modelo Conceptual

A partir de los requisitos de información, se desarrollará un diagrama conceptual de clases UML.

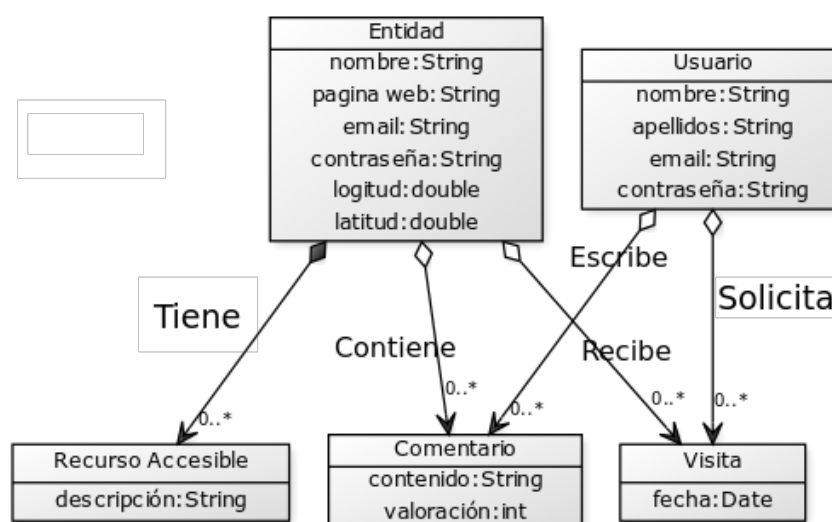


Figura 4.1: Diagrama conceptual de clases

A partir del diagrama de clases que se muestra en la figura 4.1, se comenzará el diseño de la aplicación para su posterior implementación.

### 4.2. Modelo de Casos de Uso

A partir de los requisitos funcionales descritos anteriormente, se emplearán los casos de uso como mecanismo para representar las interacciones entre los actores y el sistema bajo estudio. Para cada caso de uso deberá indicarse los actores implicados, las precondiciones y postcondiciones, los pasos que conforman el escenario principal y el conjunto de posibles escenarios alternativos.

Comenzaremos entonces por el diagrama de casos de uso que describe las relaciones entre los actores, el sistema y los casos de uso de éste.

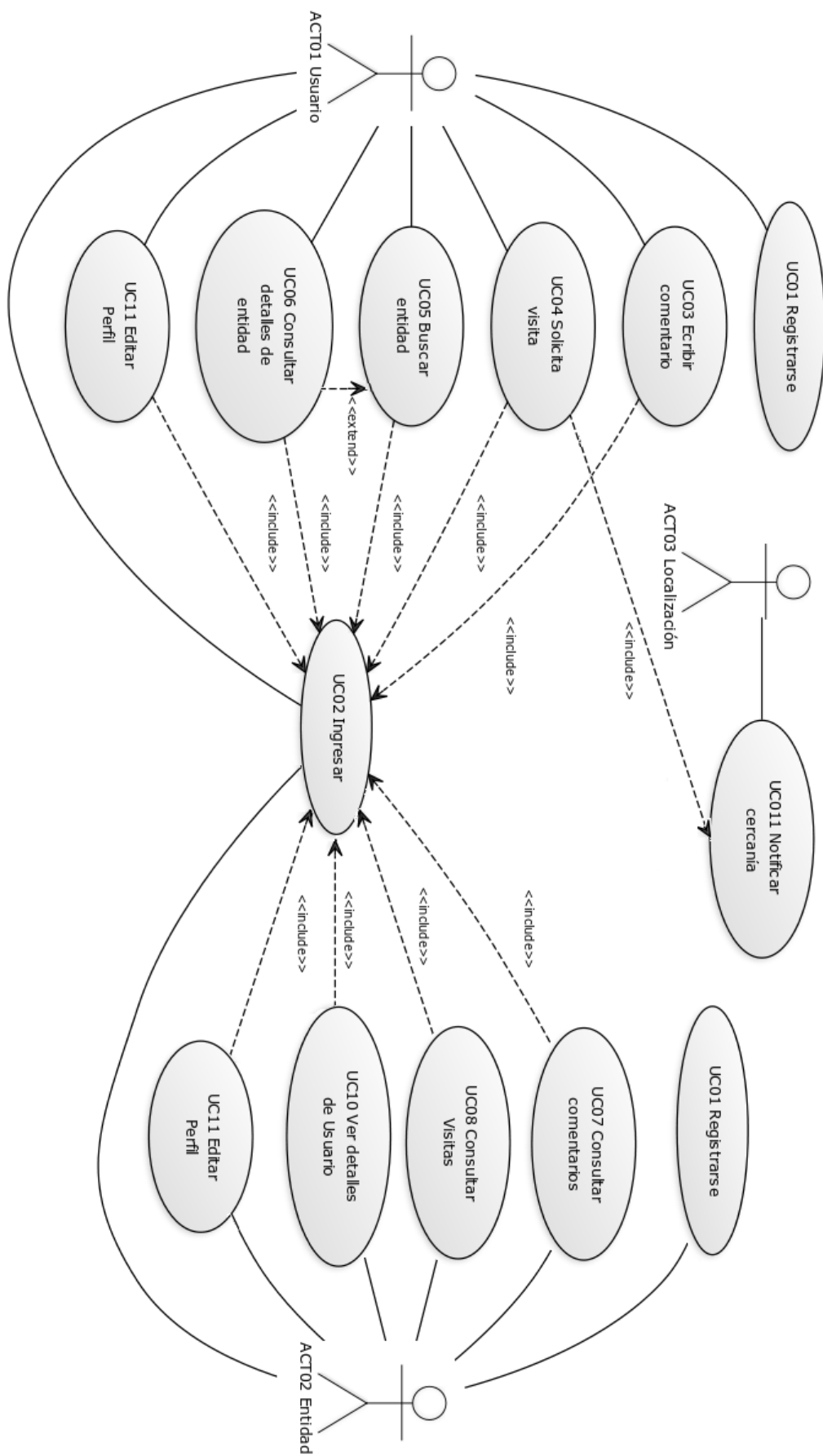
#### 4.2.1. Actores

<b>ACT-01</b>	<b>Usuario</b>
<b>Descripción</b>	Este rol lo llevará a cabo la persona que se registre en la aplicación

<b>ACT-02</b>	<b>Entidad</b>
<b>Descripción</b>	Este rol lo llevará a cabo las empresas o organizaciones que se registren en el sistema.

<b>ACT-03</b>	<b>Localización</b>
<b>Descripción</b>	Este rol lo llevará a cabo el sistema. Consiste en la geolocalización del usuario y la entidad para poder realizar las operaciones de visita.

Y con estos actores, tendríamos este modelo de casos de uso:



21  
Figura 4.2: Diagrama de casos de uso

En la siguiente tabla se expone un resumen de los casos de uso del sistema.

Código	Nombre	Actor
UC-01	Registrarse	ACT-01 Usuario o ACT-02 Entidad
UC-02	Ingresar	ACT-01 Usuario o ACT-02 Entidad
UC-03	Escribir comentario	ACT-01 Usuario
UC-04	Solicitar visita	ACT-01 Usuario
UC-05	Buscar Entidad	ACT-01 Usuario
UC-06	Consultar detalles de entidad	ACT-01 Usuario
UC-07	Consultar comentarios	ACT-02 Entidad
UC-08	Consultar visitas	ACT-02 Entidad
UC-09	Notificar cercanía	ACT-03 Localización
UC-10	Ver detalles de los usuarios	ACT-02 Entidad
UC-11	Editar perfil	ACT-01 Usuario o ACT-02 Entidad
UC-12	Añadir entidad pública	ACT-01 Usuario

#### 4.2.2. Descripción de los casos de uso

A continuación se hará una descripción más detallada de cada caso de uso

<b>Identificado</b>	<b>UC-01</b>
<b>Nombre</b>	Registrarse
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripción</b>	Acción para registrar a un usuario o entidad en el sistema, para que así pueda acceder a sus servicios
<b>Actor</b>	Usuario, Entidad
<b>Precondiciones</b>	-
<b>Postcondiciones</b>	El usuario/entidad se registra en la aplicación con un email y una contraseña
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El usuario/entidad inicia la aplicación</li> <li>2. El sistema carga la ventana de login</li> <li>3. El usuario/entidad pulsa el botón de registro</li> <li>4. El sistema carga la ventana de registro</li> <li>5. El usuario/entidad rellena los datos</li> <li>6. El sistema valida que el email no esté ya registrado y confirma la contraseña</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>6a. El sistema indica que el email ya ha sido registrado</li> <li>6b. El sistema comprueba que no todos los campos han sido rellenos y lanza un aviso</li> <li>6c. El sistema comprueba que la contraseña no coincide con su confirmación y lanza un aviso</li> </ol>
<b>Inclusiones</b>	-
<b>Extensiones</b>	

<b>Identificado</b>	<b>UC-02</b>
<b>Nombre</b>	Ingresar
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripción</b>	Acción para registrar a un usuario o entidad en el sistema, para que así pueda acceder a sus servicios
<b>Actor</b>	Usuario, Entidad
<b>Precondiciones</b>	El usuario/entidad debe de estar registrado
<b>Postcondiciones</b>	El usuario/entidad accederá a los servicios de la aplicación
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El usuario/entidad inicia la aplicación</li> <li>2. El sistema carga la ventana de login</li> <li>3. El usuario/entidad introduce su email y contraseña y pulsa el botón de acceso</li> <li>4. El sistema comprueba que el email y contraseña sean válidos</li> <li>5. El sistema muestra la ventana principal de la aplicación</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>4b. El sistema comprueba que el email y contraseña no son correctos y manda un mensaje al usuario/entidad</li> </ol>
<b>Inclusiones</b>	-
<b>Extensiones</b>	-

<b>Identificado</b>	<b>UC-03</b>
<b>Nombre</b>	Escribir comentario
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripción</b>	El usuario deja un comentario a la entidad con un contenido y una valoración
<b>Actor</b>	Usuario
<b>Precondiciones</b>	El usuario debe de estar registrado y debe acceder a la aplicación
<b>Postcondiciones</b>	El usuario deja un comentario a la entidad
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El usuario busca la entidad en la ventana principal de la aplicación</li> <li>2. El usuario pulsa la entidad para ver detalles</li> <li>3. El sistema muestra la ventana de detalles de la entidad</li> <li>4. El usuario pulsa el botón de comentario</li> <li>5. El sistema muestra un dialogo para introducir contenido y valoración</li> <li>6. El usuario introduce los datos.</li> <li>7. El sistema registra el comentario</li> </ol>
<b>Flujo alternativo</b>	-
<b>Inclusiones</b>	-
<b>Extensiones</b>	-

<b>Identificado</b>	<b>UC-04</b>
<b>Nombre</b>	Solicitar visita
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripción</b>	El usuario marca una entidad como futura visita
<b>Actor</b>	Usuario
<b>Precondiciones</b>	El usuario debe de estar registrado y debe acceder a la aplicación
<b>Postcondiciones</b>	El usuario marca la entidad para visitar, para que así el sistema pueda avisar a la entidad cuando el usuario se encuentre cerca
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El usuario busca la entidad en la ventana principal de la aplicación</li> <li>2. El usuario pulsa la entidad para ver detalles</li> <li>3. El sistema muestra la ventana de detalles de la entidad</li> <li>4. El usuario pulsa el botón de marcar para visitar</li> </ol>
<b>Flujo alternativo</b>	-
<b>Inclusiones</b>	-
<b>Extensiones</b>	-

<b>Identificado</b>	<b>UC-05</b>
<b>Nombre</b>	Buscar entidad
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripción</b>	El usuario busca una entidad en concreto
<b>Actor</b>	Usuario
<b>Precondiciones</b>	El usuario debe de estar registrado y debe acceder a la aplicación
<b>Postcondiciones</b>	El usuario encuentra la entidad
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El usuario inicia sesión en la aplicación</li> <li>2. El sistema muestra un mapa con todas las entidades</li> <li>3. El usuario introduce un nombre de entidad</li> <li>4. El sistema muestra todas las entidades que coinciden con ese nombre</li> <li>5. El usuario encuentra la entidad</li> </ol>
<b>Flujo alternativo</b>	4b. Ninguna entidad coincide con ese nombre
<b>Inclusiones</b>	-
<b>Extensiones</b>	-

<b>Identificado</b>	<b>UC-06</b>
<b>Nombre</b>	Consultar detalles de la entidad
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripción</b>	El usuario consulta los datos de una entidad
<b>Actor</b>	Usuario
<b>Precondiciones</b>	El usuario debe de estar registrado y debe acceder a la aplicación
<b>Postcondiciones</b>	El usuario consulta los detalles de la entidad
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El usuario inicia sesión en la aplicación</li> <li>2. El sistema muestra un mapa con todas las entidades</li> <li>3. El usuario selecciona la entidad deseada</li> <li>4. El sistema abre una ventana con todos los datos, comentarios y recursos accesibles de la entidad</li> </ol>
<b>Flujo alternativo</b>	-
<b>Inclusiones</b>	-
<b>Extensiones</b>	-

<b>Identificado</b>	<b>UC-07</b>
<b>Nombre</b>	Consultar comentarios
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripción</b>	La entidad consulta los comentarios que le han dejado los usuarios
<b>Actor</b>	Entidad
<b>Precondiciones</b>	La entidad debe de estar registrada y debe acceder a la aplicación
<b>Postcondiciones</b>	La entidad consulta los comentarios
<b>Flujo principal</b>	1. La entidad inicia sesión en la aplicación 2. El sistema muestra la ventana inicial de entidades con todos los datos de ésta, incluido los comentarios de los usuarios
<b>Flujo alternativo</b>	-
<b>Inclusiones</b>	-
<b>Extensiones</b>	-

<b>Identificado</b>	<b>UC-08</b>
<b>Nombre</b>	Consultar visitas
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripción</b>	La entidad consulta las próximas visitas
<b>Actor</b>	Entidad
<b>Precondiciones</b>	La entidad debe de estar registrada y debe acceder a la aplicación
<b>Postcondiciones</b>	La entidad consulta los comentarios
<b>Flujo principal</b>	1. La entidad inicia sesión en la aplicación 2. El sistema muestra la ventana inicial de entidades con todos los datos de ésta, incluido los comentarios de los usuarios 3. La entidad pulsa el botón de notificaciones de visitas. 4. El sistema muestra un listado de próximas notificaciones de visitas
<b>Flujo alternativo</b>	-
<b>Inclusiones</b>	-
<b>Extensiones</b>	-

<b>Identificado</b>	<b>UC-09</b>
<b>Nombre</b>	Notificar cercanía
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripción</b>	El sistema notifica a la entidad la cercanía de un usuario que va a visitar la empresa
<b>Actor</b>	Localización
<b>Precondiciones</b>	El usuario debe de haber marcado a la entidad para visitar
<b>Postcondiciones</b>	La entidad recibe una notificación de cercanía del usuario
<b>Flujo principal</b>	1. El usuario marca una entidad para visitar 2. El sistema manda una notificación a la entidad cuando el usuario se encuentre cerca
<b>Flujo alternativo</b>	-
<b>Inclusiones</b>	-
<b>Extensiones</b>	-

<b>Identificado</b>	<b>UC-10</b>
<b>Nombre</b>	Consultar detalles de usuario
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripción</b>	La entidad consulta detalles de los usuarios
<b>Actor</b>	Entidad
<b>Precondiciones</b>	La entidad debe de estar registrada y debe acceder a la aplicación
<b>Postcondiciones</b>	La entidad consulta los datos de los usuarios que van a verla
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. La entidad inicia sesión en la aplicación</li> <li>2. El sistema muestra la ventana inicial de entidades con todos los datos de ésta, incluido los comentarios de los usuarios</li> <li>3. La entidad pulsa el botón de notificaciones de visitas.</li> <li>4. El sistema muestra un listado de próximas notificaciones de visitas</li> <li>5. La entidad pulsa en una entrada</li> <li>6. El sistema le muestra una ventana con los datos del usuario.</li> </ol>
<b>Flujo alternativo</b>	-
<b>Inclusiones</b>	-
<b>Extensiones</b>	-

<b>Identificado</b>	<b>UC-11</b>
<b>Nombre</b>	Editar perfil
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripción</b>	El usuario/entidad editan los datos de su perfil
<b>Actor</b>	Usuario, Entidad
<b>Precondiciones</b>	El usuario/entidad debe de estar registrada y debe acceder a la aplicación
<b>Postcondiciones</b>	Los datos de usuario/ entidad quedan modificados
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El usuario/entidad inicia sesión en la aplicación</li> <li>2. El sistema muestra la ventana inicial</li> <li>3. El usuario/entidad pulsar el botón de edición de perfil</li> <li>4. El sistema muestra una ventana de edición.</li> <li>5. El usuario/entidad modifican los datos</li> <li>6. El sistema almacena los cambios</li> </ol>
<b>Flujo alternativo</b>	-
<b>Inclusiones</b>	-
<b>Extensiones</b>	-



<b>Identificado</b>	<b>UC-12</b>
<b>Nombre</b>	Registrar entidad pública
<b>Autor</b>	Juan Manuel Hidalgo Navarro
<b>Descripción</b>	El usuario introduce datos de una entidad pública
<b>Actor</b>	Usuario
<b>Precondiciones</b>	El usuario debe de estar registrada y debe acceder a la aplicación
<b>Postcondiciones</b>	Se añade al sistema de entidades una nueva entidad pública no gestionada
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El usuario/entidad inicia sesión en la aplicación</li> <li>2. El sistema muestra la ventana inicial</li> <li>3. El usuario realiza una pulsacion larga en la zona del mapa donde quiere introducir la nueva entidad pública</li> <li>4. El sistema muestra una ventana de edición.</li> <li>5. El usuario/entidad modifican los datos</li> <li>6. El sistema almacena los cambios</li> </ol>
<b>Flujo alternativo</b>	-
<b>Inclusiones</b>	-
<b>Extensiones</b>	-



## Capítulo 5

# Diseño del Sistema

En esta sección se recoge la arquitectura general del sistema de información, la parametrización del software base (opcional), el diseño físico de datos, el diseño detallado de componentes software y el diseño detallado de la interfaz de usuario.

### 5.1. Arquitectura del Sistema

En esta sección se define la arquitectura general del sistema de información, especificando la infraestructura tecnológica necesaria para dar soporte al software y la estructura de los componentes que lo forman.

#### 5.1.1. Arquitectura Física

Describiremos los principales elementos hardware que forman la arquitectura física de nuestro sistema, recogiendo por un lado los componentes del entorno de producción y los componentes de cliente.

El sistema cuenta con un Back-End, por lo que necesitaremos de un servidor que aloje el servicio web y la base de datos. Como requisito simplemente debe de ser un servidor con sistema operativo Linux (o una distribución de éste), Windows o Macintosh OS

#### 5.1.2. Arquitectura Lógica

La arquitectura lógica del sistema está formada por los elementos software (servicios, aplicaciones, librerías, frameworks, etc.) que componen el software base, más el software desarrollado para cumplir los requisitos de la aplicación de escritorio. A continuación se va a proceder a explicar los diferentes elementos software usados en el sistema.

#### Sistema de control de versiones Git

Git es el software de control de versiones que se ha elegido para el desarrollo del proyecto. Fue desarrollado por Linus Torvalds (creador del sistema operativo Linux) y a día de hoy se usa en grandes proyectos, por ejemplo el propio núcleo de Linux. Como es costumbre en el desarrollo de éste proyecto, Git es software libre distribuido bajo la licencia GPL, lo que lo hace una alternativa libre de usar.

Git permite tener el código generado en la implementación del sistema en un repositorio y poder acceder a él remotamente desde cualquier PC, así como poder restaurar versiones anteriores en caso que la actual

deba ser reemplazada.

El repositorio elegido para el sistema es GitHub, ya que actualmente es uno de los que mejores prestaciones ofrece. Las herramientas más importantes de las que dispone son las siguientes:

- Una wiki para el mantenimiento de las distintas versiones de las páginas.
- Un sistema de seguimiento de problemas, que permiten a los miembros del equipo detallar un problema software o una sugerencia que deseen hacer.
- Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero y debatir sobre determinados cambios realizados en un commit específico.
- Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

## **Android Studio**

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Cuando se presentó reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android.

Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0.

Entre sus características, estas son las más interesantes para elegirlo como IDE principal de desarrollo del proyecto:

- Renderizado en tiempo real
- Consola de desarrollador
- Soporte para construcción basada en Gradle
- Refactorización específica de Android
- Un editor de diseño enriquecido que permite grandes posibilidades en la configuración de la interfaz de usuario
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones, y otros problemas
- Un dispositivo virtual de Android que se utiliza para ejecutar y probar aplicaciones

## **Gradle**

Android Studio usa Gradle, un paquete de herramientas de compilación avanzadas, para automatizar y administrar el proceso de compilación, y al mismo tiempo te permite definir configuraciones de compilación personalizadas y flexibles.

Cada configuración de compilación puede definir un conjunto de código y recursos propios, y reutilizar las partes comunes a todas las versiones de tu app. El complemento de Android para Gradle funciona con el paquete de herramientas de compilación para proporcionar procesos y ajustes configurables específicos para la compilación y prueba de aplicaciones de Android.

## **Node.js**

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto.

El problema número uno que resuelve Node.js es proporcionar una manera fácil para construir programas de red escalables. Resuelve este problema cambiando la forma en que se realiza una conexión con el servidor. En lugar de generar un nuevo hilo de OS para cada conexión (y de asignarle la memoria acompañante), cada conexión dispara una ejecución de evento dentro del proceso del motor de Node.

Node ejecuta V8 JavaScript, siendo V8 el motor JavaScript subyacente que Google usa con su navegador Chrome.

## **MongoDB**

Es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.

En lugar de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos similares a JSON con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

## **Firebase Cloud Messaging**

Antiguamente conocido como Google Cloud Messaging (GCM), Firebase Cloud Messaging (FCM) es una plataforma para mensajes y notificaciones para Android, iOS, y aplicaciones web que actualmente puede ser usada de forma gratuita.

Con FCM, puedes notificar a una app cliente que un correo electrónico nuevo o que otros datos están disponibles para la sincronización. Puedes enviar mensajes de notificación para volver a atraer a más usuarios y aumentar su retención. Para los casos de uso de mensajería instantánea, un mensaje puede transferir una carga de hasta 4 KB a una app cliente.

Sus funciones principales son:

- Envía mensajes de notificación que se muestran a tu usuario. También puedes enviar mensajes de datos y determinar completamente lo que ocurre en el código de tu aplicación.
- Distribuye mensajes a tu app cliente en cualquiera de las siguientes tres formas: a dispositivos individuales, a grupos de dispositivos o a dispositivos suscritos a temas.
- Envía mensajes de confirmación, de chat y de otros tipos desde los dispositivos a tu servidor a través del canal de conexión confiable de FCM que consume poca batería.

## **5.2. Patrones de diseño**

El diseño de aplicaciones es complejo y es la etapa más importante del desarrollo y que más impacto tiene, no sólo sobre el producto final, sino también sobre su vida futura, mediante el mantenimiento del mismo.

En el diseño de la aplicación es donde se definen las estructuras y entidades que se van a encargar de resolver el problema planteado. Dependiendo de lo bien que se definan estas estructuras, influirá significativamente en el éxito, fracaso o mantenimiento del sistema.

Los patrones de diseño son soluciones para problemas típicos y recurrentes que nos podemos encontrar a la hora de desarrollar una aplicación. Son fruto de experiencias anteriores aplicados a diferentes problemas, probadas en determinados contextos para alcanzar un mejor diseño.

Existen numerosos patrones para solucionar distintos problemas de diseño, no obstante, en este documento solo se explicaran los patrones más importantes usados en el sistema:

- **Patrón Interface:** Usado para solucionar los problemas generados cuando estamos en una situación donde varios objetos tienen que un comportamiento similar, que de alguna forma están enlazados, pero queremos que se traten de forma distinta. Para ello aplicamos el patrón interface que ayuda a definir un comportamiento independiente de donde vaya a ser utilizado.
- **Patrón Singleton:** Permite restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Al estar internamente autoreferenciada, en lenguajes como Java, el recolector de basura no actúa.
- **Patrón Modelo-Vista-Modelo de Vista (MVVM):** MVVM es un patrón arquitectónico. Es una especialización del patrón de modelo de presentación introducido por Martin Fowler. También está relacionado con el patrón modelo-vista-controlador (MVC) y el patrón modelo-vista-presentador (MVP). Cada componente representa lo siguiente:
  - Los **modelos** representan el estado y las operaciones de los objetos que tu aplicación manipula.
  - Las **vistas** contienen elementos de la interfaz de usuario, e incluyen todo el código que implementa la experiencia del usuario de la aplicación. Una vista define la estructura, el diseño y la apariencia de lo que el usuario ve en pantalla.
  - Las vistas contienen elementos de la interfaz de usuario, e incluyen todo el código que implementa la experiencia del usuario de la aplicación. Una vista define la estructura, el diseño y la apariencia de lo que el usuario ve en pantalla.

## 5.3. Diseño de Interfaz de la aplicación

En esta sección se incluye un prototipo de baja fidelidad o mockup de la interfaz de usuario del sistema.

### 5.3.1. Navegabilidad

En las siguientes figuras se puede observar tanto los prototipos de las diferentes pantallas de la aplicación como la navegación entre éstas.

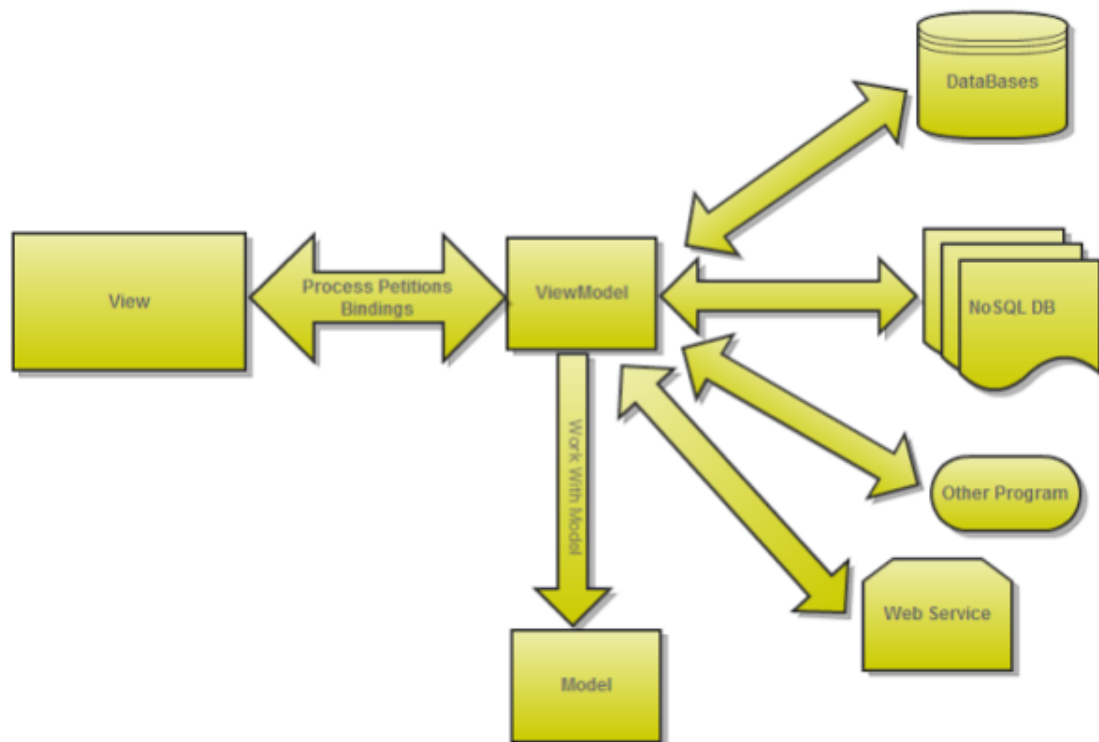


Figura 5.1: Patrón de diseño Modelo-Vista-Vista de Modelo

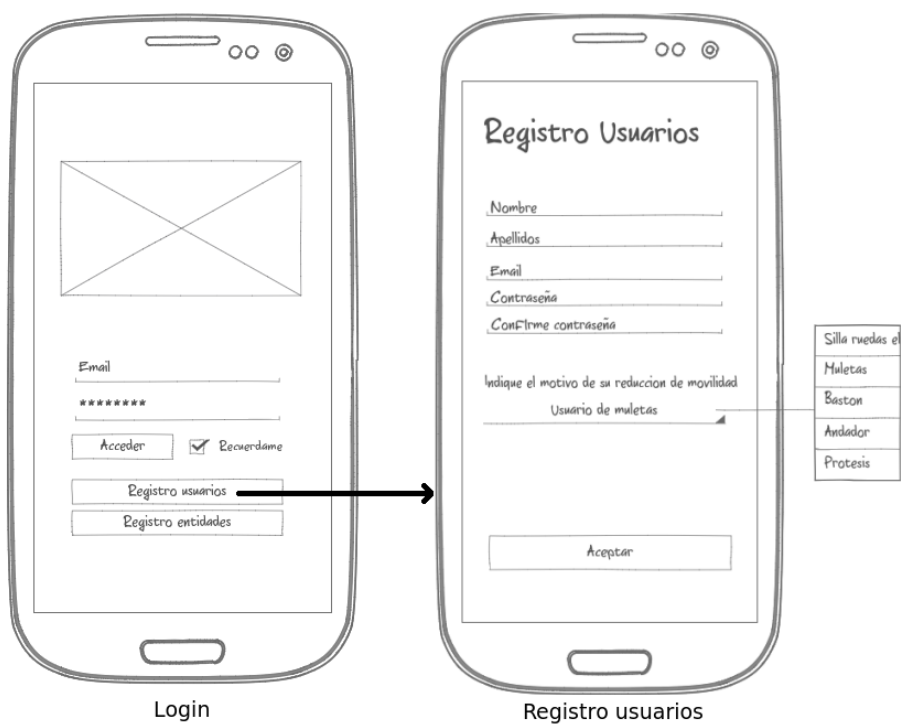


Figura 5.2: Acceso al registro de usuarios

En la figura 5.2 podemos ver el flujo principal de registro de usuarios.

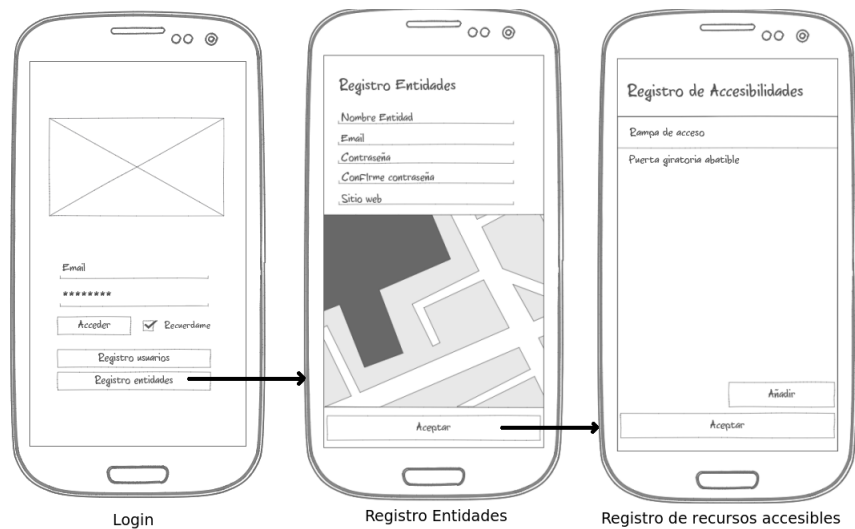


Figura 5.3: Acceso al registro de entidades

En la figura 5.3 se nos muestra las ventanas de registro de entidades, seguida de la ventana de registro de recursos accesibles.

Una vez registrado, las ventanas de menú principal de la aplicación son distintas según entremos con un usuario o con una entidad.

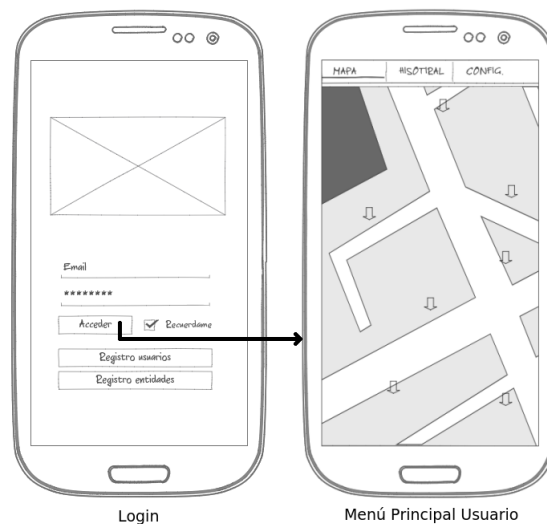


Figura 5.4: Acceso al menú principal de usuarios

Como podemos ver en 5.4, al iniciar sesión con un usuario, se nos abriría el menú principal del mapa con las entidades mostrándose en el mismo, y con tres pestañas arriba, que nos dan la capacidad de navegar por el resto de ventanas de la aplicación, como podemos ver en 5.5





Figura 5.5: Navegación por el menú principal de usuario

Dentro del menú principal, en la sección del mapa, tenemos varias posibilidades de navegación. Podremos abrir la información de una entidad pulsando en su marcador, podremos ir a la ventana de detalles de la misma pulsando en el dialogo que aparece tras pulsar el marcador o podremos incluir una entidad pública haciendo una pulsación larga en el mapa. Podremos verlo en la figura 5.6

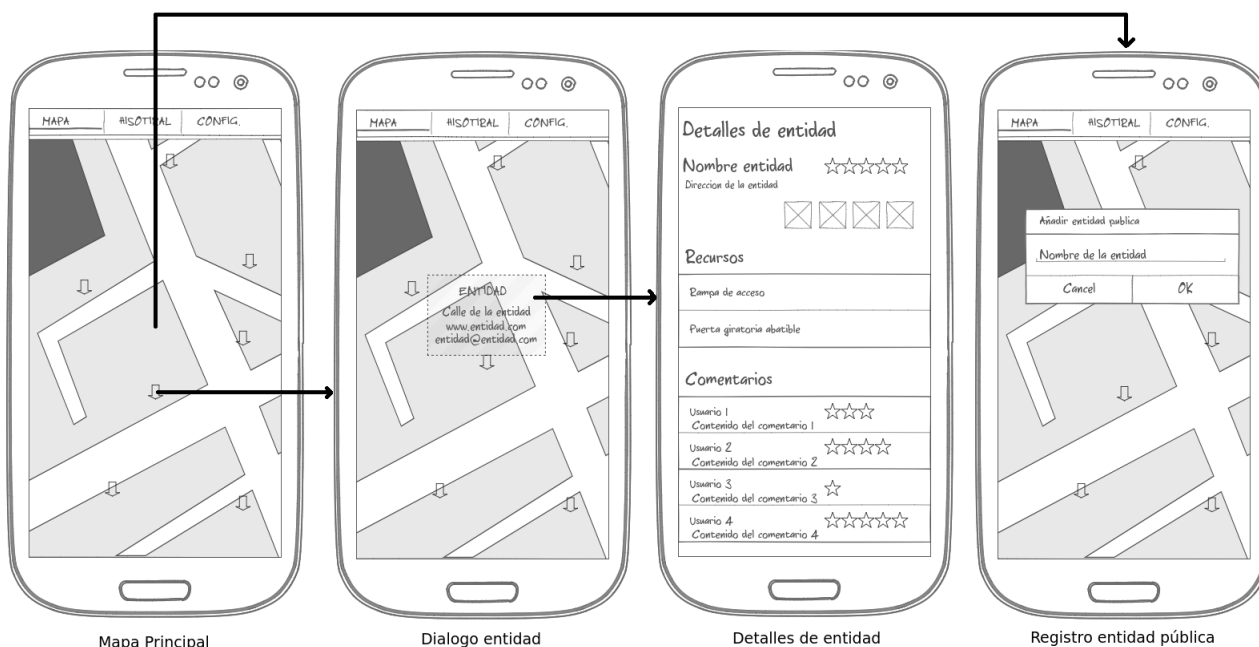


Figura 5.6: Navegación dentro del Mapa

Si ingresamos en la aplicación como una entidad, la ventana principal que se mostraría sería la siguiente:

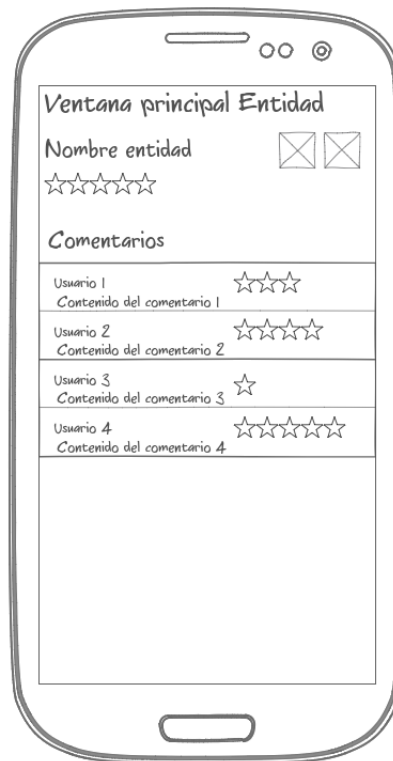


Figura 5.7: Ventana principal de entidad

Y por último, la navegación disponible que tendríamos en la entidad sería el menú de notificaciones de visitas, que podemos ver en la figura ?? (Falta por diseñar, Pablo, la incluiré más adelante)

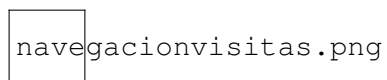


Figura 5.8: Ventana de visualización de notificaciones de visitas

### 5.3.2. Diseño detallado de la interfaz

En este apartado, veremos con detalle el diseño de cada una de las interfaces de la aplicación:

# Bibliografía

- [1] Charles Arthur. Half the uk population owns a smartphone. *The Guardian*, 2011.
- [2] Tiago Silva Da Silva, Angela Martin, Frank Maurer, and Milene Silveira. User-centered design and agile methods: a systematic review. In *Agile Conference (AGILE)*, 2011, pages 77–86. IEEE, 2011.
- [3] Android Developers. What is android, 2011.



# Software utilizado

Es usual en un PFC referenciar que software has usado para la realización del mismo. Aprovecharé este apartado para que conozcas alguna herramienta que puede serte de ayuda para realizar tus documentos en  $\text{\LaTeX}$

## Emacs + AucTeX

Emacs es uno de los programas de edición más usados por desarrolladores de software, ya que es bastante versátil admitiendo gran cantidad de “plugins” o extensiones que permiten ampliar aun más sus funcionalidades.

Uno de estos plugins es AucTeX [?], el cual incluye rutas para ciertos comandos, resaltado de sintaxis, previsualización del documento, menú matemático en el cual podemos acceder e insertar la gran mayoría de los símbolos matemáticos, para no tener que memorizarlos. Podemos ver un ejemplo de Emacs + AucTeX en la figura 5.9

Por ejemplo, para cerrar un entorno `\begin()`, con su respectivo `\end()`, utilizaremos el atajo `C-c M-]`, para añadir un `\item`, tenemos el atajo `C-c C-j`, y así unos cuantos, que una vez que nos habituamos a ellos, son bastante cómodos.

Además, es bastante configurable, con indentado automático, corrector ortográfico y demás. El fichero adjunto a este documento, *conf\_emacs* incluye una configuración con varias de estas opciones.

## Doxygen

Realmente, *Doxygen* [?] no es una herramienta que vayamos a utilizar para realizar documentos  $\text{\LaTeX}$  directamente. Sin embargo, para la documentación de código si es bastante útil.

Esta herramienta realiza una documentación automática de código fuente. Es decir, para nuestro PFC, podemos utilizar para generar la documentación de las APIs de nuestras librerías y demás. Puede generar esta documentación en varios formatos, y entre ellos,  $\text{\LaTeX}$ , de forma que podemos utilizar ese código generado en nuestra memoria de forma automática.

## GNU Make

*GNU Make* es el programa de recompilación y de control de dependencias por excelencia. Se puede utilizar para compilar proyectos software en diversos códigos, o como en el caso de este documento, para compilar documentos  $\text{\LaTeX}$  con diversas opciones.

Para más información [?]

## Dia

*Dia* es un editor de gráficos vectoriales el cual incluye distintas plantillas para distintos tipos de gráficos, como pueden ser UML, ERe, diagramas de flujo, esquemas Cisco de red y un larguísimo etcétera. Podemos ver el interfaz en la figura [5.10](#)

Estos diagramas podemos exportarlos a diversos formatos de imagen (.png, .eps, ...) o a formato .tex, como vimos anteriormente.

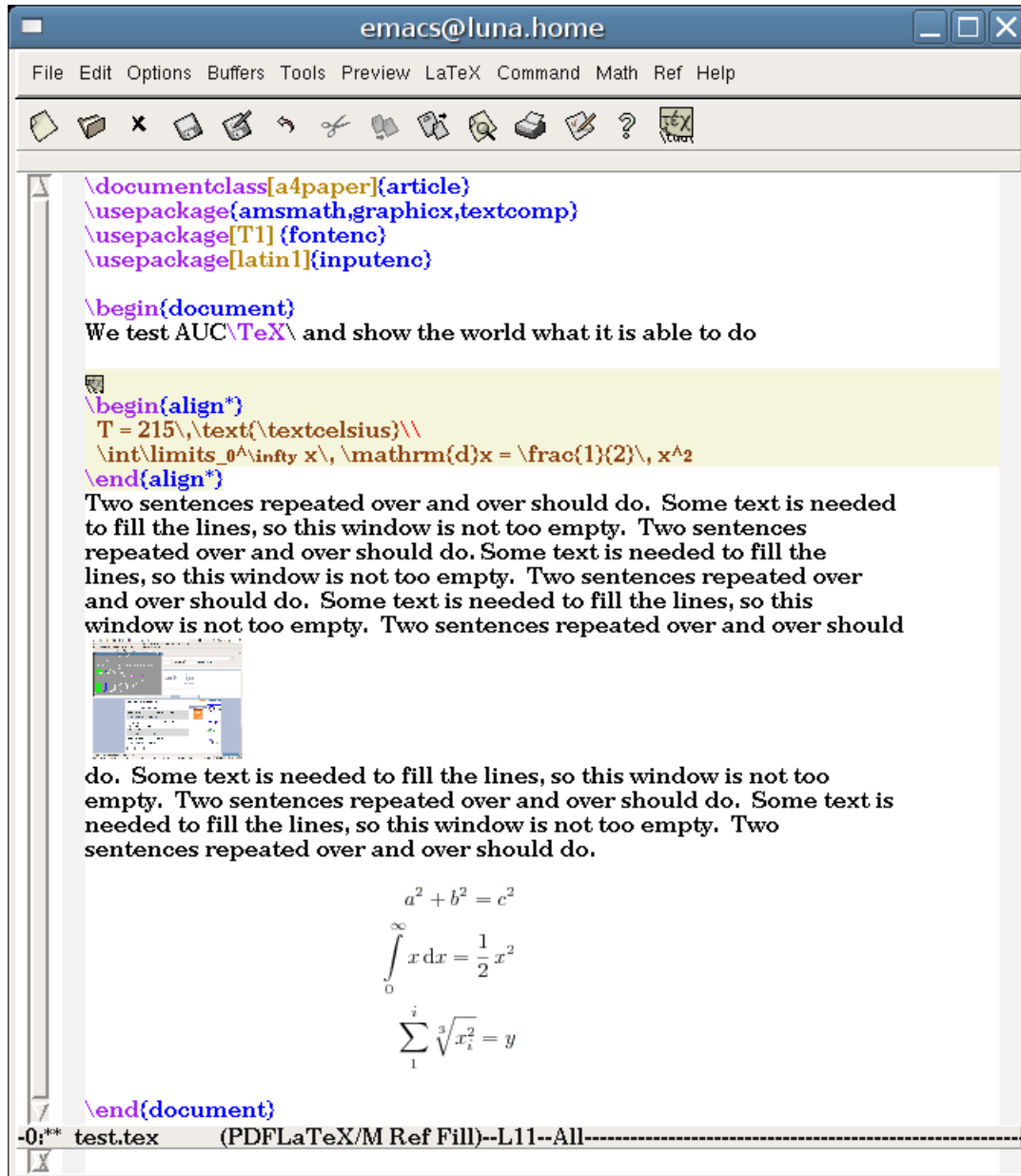


Figura 5.9: Emacs + Auc $\TeX$

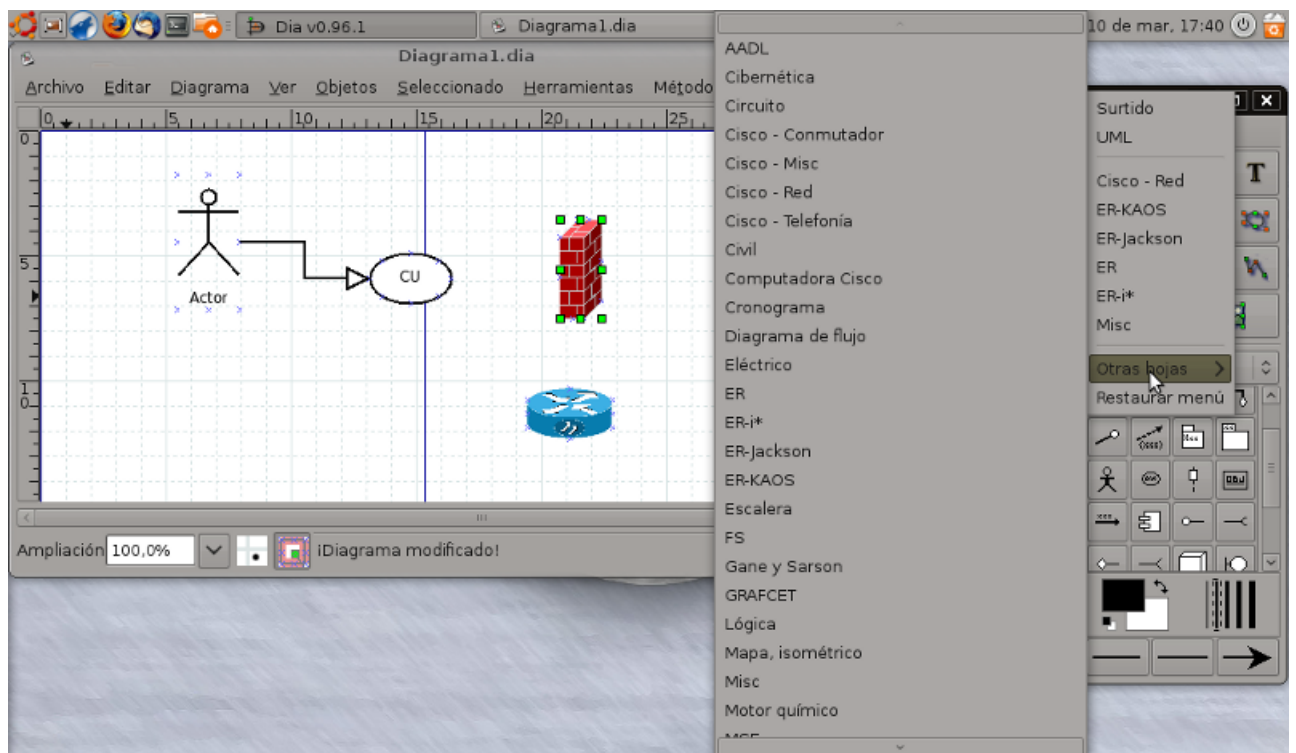


Figura 5.10: Interfaz de Dia



# Instalación de L<sup>A</sup>T<sub>E</sub>X

Veamos que tenemos que hacer para instalar L<sup>A</sup>T<sub>E</sub>X con todas sus capacidades en un sistema basado en Debian, como Ubuntu. Primero hay que tener en cuenta que L<sup>A</sup>T<sub>E</sub>X es relativamente pesado con respecto a otros compiladores.

Nosotros vamos a utilizar la distribución de L<sup>A</sup>T<sub>E</sub>X incluida en los repositorios de Ubuntu llamada *texlive*. Si la buscas en tu gestor de paquetes, encontrarás infinidad de paquetes aparte del principal. Existen otras distribuciones como Te<sub>X</sub>

Si instalas solo los básicos, es decir instalas *texlive* y los programas necesarios para él, no podrás compilar este documento, ya que faltarían paquetes tales como *supertabular* y varios. Por eso, si no tienes problema de espacio en el disco duro te recomiendo que instales el paquete *texlive-full*, que instala **todos** los paquetes de *texlive*, incluyendo documentación en todos los idiomas disponibles. Si buscas no tener problemas de dependencias, este es tu método.

```
sudo apt-get install texlive-full
```

En caso de querer ser un poco más concreto, en principio puedes trabajar con la más básica (*texlive* y sus dependencias) y en función de los paquetes que te vayan faltando, los instalas.



# GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”). To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## **5. COMBINING DOCUMENTS**

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

## **6. COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## **8. TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## **9. TERMINATION**

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## **10. FUTURE REVISIONS OF THIS LICENSE**

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## **11. RELICENSING**

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

## **ADDENDUM: How to use this License for your documents**



To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.