Resumen

El proyecto consiste en el desarrollo de una aplicación nativa para Android para que los usuarios puedan compartir vehículo a diario para ir a trabajar. Con ello se busca conseguir un doble objetivo: que los usuarios puedan ahorrar dinero en sus desplazamientos diarios para ir a trabajar sin perder por ello la comodidad o rapidez de utilizar el vehículo particular y, segundo, reducir los atascos y la contaminación provocadas por el uso individual del vehículo. Para ello, la aplicación permitirá a los conductores crear viajes donde indicarán el lugar exacto y la hora a la que realizarán los desplazamientos, así como la tarifa que deben abonar los usuarios. Después, los usuarios pueden buscar aquellos viajes que se encuentren a una distancia máxima y unirse.

Descriptores

* Vehículo
* Carpooling
* Android

Índice

[1. INTRODUCCIÓN 1](#_Toc467520604)

[1.1 Presentación del documento 1](#_Toc467520605)

[1.2 Presentación del problema 1](#_Toc467520606)

[2. DEFINICIÓN Y OBJETIVOS DEL PROYECTO 2](#_Toc467520607)

[2.1 Definición de objetivos 2](#_Toc467520608)

[2.1.1 Objetivos 2](#_Toc467520609)

[2.1.2 Alcance 2](#_Toc467520610)

[2.2 Produco final 2](#_Toc467520611)

[2.3 Descripción de la realización 3](#_Toc467520612)

[2.3.1 Método de Desarrollo 3](#_Toc467520613)

[2.3.2 EDT 3](#_Toc467520614)

[2.3.3 Definición de tareas 3](#_Toc467520615)

[2.3.4 Productos Intermedios 3](#_Toc467520616)

[2.4 Organización 3](#_Toc467520617)

[2.4.1 Esquema organizativo 3](#_Toc467520618)

[2.4.2 Plan de Recursos Humanos 4](#_Toc467520619)

[2.5 Condiciones de ejecución 4](#_Toc467520620)

[2.5.1 Entorno de Trabajo 4](#_Toc467520621)

[2.5.2 Control de Cambios 5](#_Toc467520622)

[2.5.3 Recepción de los Productos 5](#_Toc467520623)

[2.6 Planificación 5](#_Toc467520624)

[2.6.1 Plan de Trabajo 5](#_Toc467520625)

[2.6.2 Diagrama de precedencias 5](#_Toc467520626)

[2.6.3 Diagrama de Gantt 5](#_Toc467520627)

[2.6.4 Reparto de Cargas de Trabajo por Recurso Humano del Proyecto 6](#_Toc467520628)

[2.6.5 Cargas de Trabajo para las Diferentes Tareas del Proyecto 6](#_Toc467520629)

[2.6.6 Presupuesto 6](#_Toc467520630)

[3. ESPECIFICACIÓN DE REQUISITOS 7](#_Toc467520631)

[3.1 Visión general 7](#_Toc467520632)

[3.2 EspEcificación de requisitos de la aplicación Android 7](#_Toc467520633)

[3.3 Casos de uso 7](#_Toc467520634)

[3.3.1 Registro 8](#_Toc467520635)

[3.3.2 Unirse a una ruta 8](#_Toc467520636)

[4. TECNOLOGÍAS UTILIZADAS 10](#_Toc467520637)

[4.1 Visión general 10](#_Toc467520638)

[5. ESPECIFICACIóN DEL DISEÑO 11](#_Toc467520639)

[5.1 Visión general 11](#_Toc467520640)

[5.2 Entorno de desarrollo 11](#_Toc467520641)

[5.2.1 Android Studio 11](#_Toc467520642)

[5.2.2 GitHub 11](#_Toc467520643)

[5.2.3 NotePad++ 11](#_Toc467520644)

[5.2.4 Visual Paradigms 12](#_Toc467520645)

[5.2.5 Microsoft Project 12](#_Toc467520646)

[5.3 Modelo de la base de datos 12](#_Toc467520647)

[5.4 Arquitectura de la aplicación 12](#_Toc467520648)

[6. IMPLEMENTACIÓN 13](#_Toc467520649)

[6.1 Visión general 13](#_Toc467520650)

[6.2 Implementación de LA APLICACIÓN 13](#_Toc467520651)

[6.3 14](#_Toc467520652)

[7. PRUEBAS 14](#_Toc467520653)

[7.1 Visión general 14](#_Toc467520654)

[7.2 Log 14](#_Toc467520655)

[7.3 Test 14](#_Toc467520656)

[7.4 Rendimiento de la base de datos 14](#_Toc467520657)

[8. MANUAL DE USUARIO 15](#_Toc467520658)

[8.1 Visión General 15](#_Toc467520659)

[9. CONCLUSIONES Y LÍNEAS FUTURAS 16](#_Toc467520660)

[9.1 Visión general 16](#_Toc467520661)

[9.2 Conclusiones 16](#_Toc467520662)

[9.3 Líneas futuras 16](#_Toc467520663)

[10. BIBLIOGRAFÍA 17](#_Toc467520664)

Índice de figuras

FIGURA 4.1: Modelo de desarrollo incremental......................................................................... 12

FIGURA 4.2: EDT ...................................................................................................................... 13

FIGURA 5.1: Organización ........................................................................................................ 30

FIGURA 7.1: Precedencias......................................................................................................... 33

FIGURA 7.2: Plan de Trabajo...................................................................................................... 33

FIGURA 7.3: Diagrama de Gantt................................................................................................. 34

FIGURA 7.4: Cargas de trabajo................................................................................................... 35

FIGURA 7.5: Tareas..................................................................................................................... 35

Índice de tablas

# INTRODUCCIÓN

## Presentación del documento

* **Definición y objetivos del proyecto:**
* **Especificación de requisitos:**
* **Tecnologías utilizadas:**
* **Especificación del diseño:**
* **Implementación:**
* **Pruebas.**
* **Manual de usuario:**
* **Conclusiones y líneas futuras:**

## Presentación del problema

## Aplicaciones Similares

**ANDROID**

* Blablacar (+ 10.000.000 descargas):
  + Solo permite contratar/publicar un viaje cada vez
  + No especifica el lugar de salida -> En grandes ciudades como Madrid puedo ser un inconveniente debido a las grandes distancias
* Amovens (+ 100.000 descargas):
  + Solo permite contratar/publicar un viaje cada vez

**WEB**

* <http://www.compartocoche.com/>
* <https://www.carpling.com/es/car/compartir-coche-al-trabajo/>
  + Ambas ofrecen compartir coche para ir al trabajo
  + Diseño y funcionalidad poco intuitiva
  + Solo web

# DEFINICIÓN Y OBJETIVOS DEL PROYECTO

## Visión general

## Definición de objetivos

### Objetivos

El objetivo de este proyecto es desarrollar una aplicación en Android que permita a las personas que se tienen que desplazar a diario a sus lugares de trabajo compartir vehículo. Para ello, se tratará de ofrecer al usuario aquellos viajes que mejor se adapten a sus necesidades, tratando de reducir las diferencias entre la experiencia de viajar en coche propio o compartir vehículo.

Los objetivos secundarios del proyecto serán conectarse a Google Maps para que el usuario pueda indicar con precisión tanto el punto de partida como el de llegada. Además, la aplicación debe ser intuitiva y fácil de usar y, también, multilenguaje (español e inglés de momento), con el objetivo de ser accesible al mayor número de usuarios posibles.

### Alcance

De acuerdo a los objetivos previamente señalados en el apartado anterior, el proyecto constará de los siguientes aspectos:

* Una aplicación Android nativa, intuitiva y fácil de usar por los usuarios.
* Permitir a los usuarios que se registren y logueen en la aplicación
* Permitir a los usuarios crear y unirse a rutas.
* Un sistema de búsqueda que permita al usuario encontrar aquellas rutas que mejor se adaptan a sus necesidades.
* Debe cambiar de idioma dependiendo de las preferencias del usuario.
* Conexión con Google Maps para obtener datos precisos sobre la localización de los usuarios.

## Produco final

El producto final del proyecto consiste en una aplicación Android, que permita a los usuarios crear o unirse a rutas para compartir vehículo diariamente para ir a trabajar.

Para ello, la aplicación permitirá registrarse en ella a las personas interesadas en su uso, a través de un nombre de usuario único, una contraseña y una dirección de correo electrónico para poder contactar en caso de necesidad.

El aplicativo contará con una base de datos que guarde toda la información de las rutas creadas por los usuarios y con un buscador que encuentre las rutas que mejor se adaptan a las necesidades de cada usuario.

Además, la base de datos se encontrará en un servidor externo especializado y se accederá a él a través de un servidor.

Por último, el usuario podrá modificar sus datos, a excepción del nombre de usuario, así como limitar el acceso a su perfil, a través de un menú de configuración.

## Descripción de la realización

### Método de Desarrollo

El método utilizado para desarrollar este proyecto es el iterativo e incremental. Este modelo divide el proyecto en varias partes y, en cada una de ellas realiza un desarrollo completo, desde el análisis hasta la pruebas e implementación: tras el cual, se obtiene una parte completamente funcional del proyecto. Este proceso se repite en varias ocasiones hasta lograr el producto final.

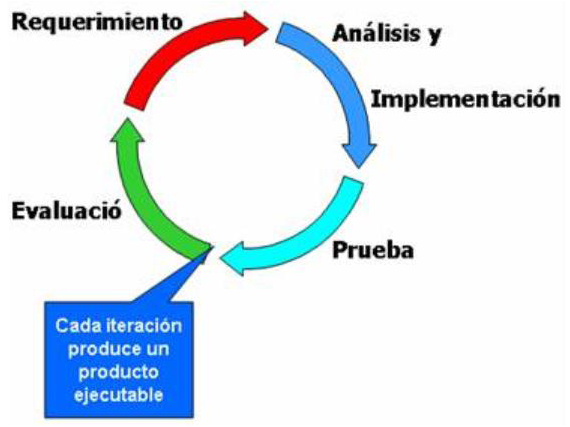


FIGURA 4.1 Modelo de desarrollo incremental

Este proceso de desarrollo intenta solucionar los defectos y problemas ocasionados por el modelo en cascada, que obligaba a completar una fase antes de comenzar con la siguiente y eso provocaba que no se adecuase a las necesidades específicas del desarrollo de software. Gracias a las iteraciones, este proceso permite encontrar y solucionar problemas de desarrollo en las fases iníciales y solventarlos del modo más efectivo posible. Además, tras cada desarrollo se obtiene un feedback y un aprendizaje, de cara a mejorar el proceso en las próximas iteraciones.

### EDT

### Definición de tareas

#### Planificación

* Lanzamiento del proyecto: aceptación del proyecto, análisis del mismo y selección del equipo de desarrollo.
* Análisis de herramientas para el desarrollo de la aplicación móvil: investigación de los diferentes sistemas operativos que existen actualmente en el mercado móvil, así como las diferentes tecnologías de desarrollo disponibles.
* Análisis de herramientas del servidor: investigación de las herramientas actuales para el desarrollo de servidores.
* Seguimiento: control del progreso del proyecto.

#### Modelo de datos

* Análisis de herramientas de bases de datos: investigar las diferentes tecnologías disponibles para la persistencia de los datos y escoger en base a necesidades futuras, como la escalabilidad.
* Identificación de las clases: análisis de las necesidades de la aplicación y diseño del sistema.

#### Diseño de la aplicación

* Desarrollo de la aplicación: diseño e implementación de la aplicación móvil.
* Desarrollo del servidor: diseño e implementación de la lógica del servidor.
* Desarrollo de la interfaz gráfica: diseño de la interfaz gráfica siguiendo los patrones de diseño existentes.
* Implementación: puesta en marcha del proyecto en fase beta.
* Pruebas: ejecución de unas pruebas bien definidas para comprobar el funcionamiento del software.

#### Despliegue y Cierre

* Memoria: realización del manual de usuario y de una memoria sobre la realización del proyecto.
* Informe sobre el desarrollo: reunión del equipo para registrar el conocimiento adquirido durante la realización del proyecto sobre los problemas surgidos y la manera de afrontarlos y superarlos, para generar un documento de guía para futuros casos.

### Productos Intermedios

* Estudio actual del mercado: documento con las diferentes aplicaciones que existen actualmente para compartir vehículo.
* Planificación: documento de planificación para el desarrollo del proyecto.
* Modelo de datos: diseño y creación de la base de datos que dará soporte a la aplicación.
* Diseño de la aplicación: desarrollo e implementación de la aplicación.
* Despliegue y cierre del proyecto: análisis y recogida de datos sobre los problemas del proyecto.

## Organización

### Esquema organizativo

* Director de proyecto: es el encargado de controlar y gestionar el buen funcionamiento y coordinación del grupo de trabajo, así como, orientar en el desarrollo de proyecto y definir las tareas que se deberán realizar. Por último, deberá aprobar las entregas realizadas en cada iteración.
* Equipo de trabajo: grupo multidisciplinar encargado del desarrollo e implementación del proyecto con las funcionalidades acordadas y siguiendo la planificación previa.



2.3. Organización

* Reuniones de Seguimiento: se celebrarán cada dos semanas al mes y en las que se analizará, basándose en los Informes de Seguimiento, la marcha del proyecto, detectando los posibles problemas que puedan surgir en su evolución y planteando alternativas de solución a los mismos. La celebración de dichas reuniones es competencia del Director del Proyecto.
* Hitos de Validación: tras la primera reunión de Lanzamiento del Proyecto, el director validará la finalización de cada una de las fases del proyecto en la medida que se cumplimenten todas y cada una de sus actividades. El director dispondrá de cinco días laborables para proporcionar un informe de validación, con la conformidad o no del producto. Este objetivo es fácilmente asumible ya que la implantación es progresiva y se realiza con la directa participación de los usuarios implicados. La no-formalización de este informe de validación en el plazo indicado dará por validada la fase.
* Recepción: se considera como recepción al último hito de validación, y se corresponde con el final del período de seguimiento de la explotación.

### Plan de Recursos Humanos

El equipo de trabajo idóneo estaría formado por los siguientes perfiles:

* Director del proyecto: es el encargado de organizar el proyecto, coordinar al equipo de desarrollo y controlar que se cumpla la planificación.
* Administrador de BD: su función es la de diseñar, programar y mantener la base de datos del proyecto.
* Programador Android: su función es el diseño y desarrollo de la aplicación móvil.
* Diseñador gráfico: es el encargado del diseño y creación de las interfaces graficas sencillas y fáciles para el usuario.
* Programador del servidor: su función es el diseño y desarrollo del servidor de la aplicación.

## Condiciones de ejecución

### Entorno de Trabajo

El lugar de trabajo será el hogar del director del proyecto.

El horario será variable debido a la situación actual del desarrollador del proyecto, estudiante; dependiendo de las tareas que tenga que realizar para otras asignaturas. Eso provocará que se tenga que trabajar tanto en días laborales, como fin de semanas y festivos.

Los medios informáticos de los que se disponen son los siguientes:

* HARDWARE
* Ordenador portátil Sony Vaio
* Smartphone Sony Xperia.
* SOFTWARE
* Android Studio
* Microsoft Project

El director del proyecto es responsable de la seguridad de los módulos de software, datos y documentación presentes en sus locales, incluso durante la fase de desarrollo.

### Control de Cambios

Todas las peticiones formuladas durante la ejecución del proyecto, que impliquen cambios en las especificaciones, diseños o desarrollos ya realizados, o en el entorno de desarrollo que se describe en el siguiente procedimiento:

1. Comunicación formal al director del proyecto sobre las mordicaciones solicitadas.
2. Valoración por el equipo del proyecto de las repercusiones, tanto técnicas como de planificación
3. Presentación de una oferta valorada.
4. Aprobación o rechazo de la oferta.
5. En caso afirmativo, modificación del plan de trabajo y del presupuesto.

### Recepción de los Productos

Para la recepción de cada producto software, que constituyan todo o parte del conjunto a desarrollar, se definirán previamente las pruebas que se consideren necesarias. Si se obtienen unos resultados satisfactorios en dichas pruebas, los productos se entregarán al director del proyecto para su implementación. En caso negativo, se comunicarán dichos fallos al equipo de trabajo en un plazo máximo de cinco días laborables para su análisis y corrección.

Los documentos tales como estudios, diseños, especificaciones funcionales, documentación, análisis, etc. deberán ser entregados y aprobados por el director del proyecto.

El director dispondrá de cinco días laborables para comunicar formalmente su aceptación o rechazo del módulo. Si en el plazo antes citado no se recibe respuesta alguna, los módulos se considerarán recibidos.

## Planificación

En este apartado se detallan las tareas y la planificación del proyecto.

### Plan de Trabajo

En la siguiente tabla, se muestran las tareas a realizar especificadas anteriormente, el recurso asignado a cada tarea y las fechas de comienzo y fin.

### Diagrama de precedencias

En el siguiente gráfico, se muestra el orden en el que se realizarán las tareas

### Diagrama de Gantt

En este apartado se muestra el diagrama de Gantt

### Reparto de Cargas de Trabajo por Recurso Humano del Proyecto

### Cargas de Trabajo para las Diferentes Tareas del Proyecto

### Presupuesto

Presupuesto total del proyecto, incluyendo el coste de los recursos humanos y las necesidades de hardware y software para el desarrollo de la aplicación.

# ESPECIFICACIÓN DE REQUISITOS

## Visión general

En este apartado se detallan los requisitos que el proyecto desarrollado debe satisfacer y que definen el funcionamiento de todo el software del mismo. Además, se exponen los casos de usos de la aplicación y se muestran los diagramas de secuencia de los más complejos. Este apartado se divide en los siguientes bloques:

* Especificación de requisitos de la aplicación Android:requisitos que debe cumplir la aplicación web.
* Casos de uso

## EspEcificación de requisitos de la aplicación Android

Los requisitos funcionales de la aplicación son:

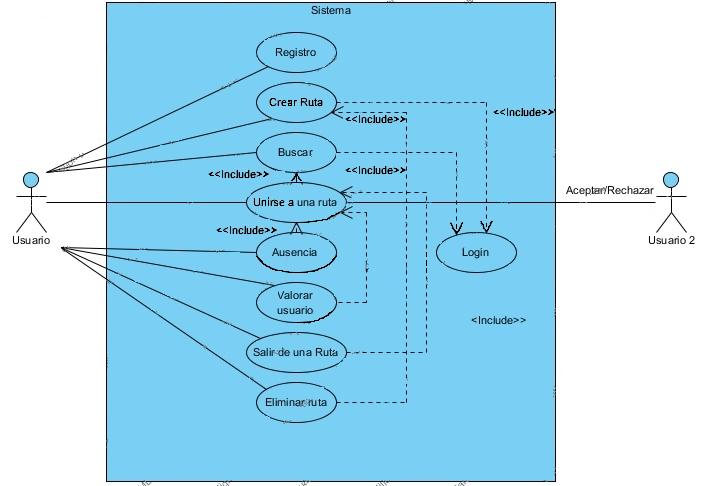
1. La aplicación debe permitir a los usuarios que se registren y que inicien sesión
2. La aplicación de permitir a los usuarios crear viajes
3. La aplicación debe permitir al usuario buscar los viajes que más le pueden ser útiles, en base a diferentes criterios
4. La aplicación debe permitir a los usuarios unirse a un viaje
5. La aplicación debe permitir al usuario conductor aceptar o rechazar a un usuario
6. La aplicación debe permitir a los usuarios avisar si no van a acudir al viaje
7. La aplicación debe permitir indicar que un viaje se ha realizado con éxito o, en caso de que exista algún problema, notificarlo
8. La aplicación debe permitir valorar al conductor
9. La aplicación debe permitir salir de un viaje
10. La aplicación debe permitir eliminar un viaje

Los requisitos no funcionales de la aplicación son:

1. La aplicación debe ser intuitiva y fácil de usar

## Casos de uso

A continuación, se muestran los principales casos de uso de la aplicación:

****

### Registro

Este caso de uso representa el registro inicial y el terminar registro por parte del usuario en la aplicación.

* **Nombre:** Registro + Terminar registro
* **Propósito:** El objetivo de este caso de uso es completar todos los pasos del registro del usuario en la aplicación.
* **Actores:** El usuario
* **Poscondición:** el usuario estará registrado en la aplicación.
* **Escenario principal:**
  1. El usuario abre la aplicación.
  2. El usuario pulsa el botón ‘Registro’.
  3. El usuario rellena el formulario.
* **Escenarios alternativos:**

1. Se repite el paso hasta el formulario es correcto.

### Unirse a una ruta

Este caso de uso representa al usuario uniéndose a una ruta ya creada

* **Nombre:** Unirse a una ruta
* **Propósito:** El objetivo de este caso de uso es permitir al usuario unirse a una ruta ya existente.
* **Actores:** El usuario
* **Precondición:** el usuario está logeado en la aplicación y la ruta está creada.
* **Pos condición:** el usuario queda pendiente de ser aceptado en una ruta.
* **Escenario principal:**
  1. El usuario pulsa el botón ‘Buscar ruta’.
  2. El usuario rellena los campos disponibles y pulsa el botón ‘Buscar’.
  3. Si existen rutas disponibles, se muestran al usuario.
  4. El usuario selecciona una ruta.
  5. El usuario pulsa el botón ‘Unirme’.
* **Escenarios alternativos:**

1. Se repite el paso hasta el formulario es correcto.
2. Si no existen rutas, se vuelve a rellenar formulario.

# TECNOLOGÍAS UTILIZADAS

## Visión general

En este capítulo se exponen las tecnologías utilizadas para el desarrollo de la aplicación.

## Android

## Java

## Node.js

## JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes. Todos los navegadores modernos interpretan el código

JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

## MongoDB

## JSON

## Firebase

## Google Maps

# ESPECIFICACIóN DEL DISEÑO

## Visión general

El objetivo de este capítulo es describir la labor de diseño realizada para desarrollar el proyecto, así como las herramientas y entornos que han sido necesarios para su realización. Para ello, contiene los siguientes apartados:

* Entorno de desarrollo
* Arquitectura de la aplicación
* Modelo de la Base de Datos

## Entorno de desarrollo

### Android Studio

### GitHub

Git es un sistema de control distribuido con énfasis en la velocidad. Este sistema de control de versiones es muy popular en proyectos open source, y su autor original fue Linus Torvald. Al igual que la mayoría de sistemas de control de revisiones distribuidos, cada directorio de trabajo Git es un repositorio con la historia completa y las plena capacidades de un sistema completo de seguimiento de versiones, independientemente del acceso a la red o de la existencia de un servidor central.

Su principal ventaja es su gran rendimiento, ya que, al tratarse de un sistema distribuido, las búsquedas son mucho más eficaces lo que supone una gran rapidez para detectar diferencias entre archivos. Además, su sistema de ramas o branches es muy flexible, pudiendo crear ramas locales independientes de un repositorio centralizado.

GitHub es un servicio de alojamiento repositorio Git basado en la web GitHub, que ofrece todas las funcionalidades de Git sobre el control de revisión distribuida y la gestión de código fuente, así como añade sus propias características. A diferencia de Git, GitHub ofrece una interfaz gráfica, tanto en la web como en el escritorio, con integración para móviles. También proporciona control de acceso y varias características de colaboración como wikis, gestión de tareas y seguimiento de errores y peticiones para cada proyecto.

### NotePad++

Notepad++ es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. Incluye opciones más avanzadas que pueden ser útiles para usuarios avanzados como desarrolladores y programadores. Se distribuye bajo los términos de la Licencia Pública General de GNU. Soporta resaltado de sintaxis y plegado de código de más de 50 lenguajes de programación, scripting, y marcado. Trata de detectar automáticamente el lenguaje que utiliza un fichero determinado, utilizando una lista modificable de extensiones de archivo. El programa también soporta autocompletado para un subconjunto de la API de algunos lenguajes de programación.

### Sublime Text

Sublime Text es un editor de texto y código, y está cargado de funcionalidades útiles y cómodas desde el punto de la usabilidad y eficiencia. Las características más importantes del editor son:

* Da soporte nativo para 43 lenguajes de programación y texto plano.
* Auto completado y marcado de llaves
* Resalta las expresiones propias de la sintaxis del lenguaje, facilitando su lectura.
* Permite la edición simultánea, lo que facilita los cambios de nombre de una variable

### Visual Paradigms

Visual Paradigm for UML es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. Además del soporte para el modelado, proporciona la generación de informes y es capaz de generar código automático.

En este proyecto se ha utilizado para generar el modelo de la base de datos, la arquitectura de la aplicación y los casos de uso.

### Microsoft Project

Microsoft Project (o MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

El software Microsoft Office Project en todas sus versiones (la versión 2013 es la más reciente a febrero de 2013) es útil para la gestión de proyectos, aplicando procedimientos descritos en el PMBoK del Project Management Institute.

## Modelo de la base de datos

## Arquitectura de la aplicación

# IMPLEMENTACIÓN

## Visión general

Este capítulo contiene observaciones sobre cómo se han implementado las tecnologías utilizadas en el desarrollo del proyecto.

## Implementación de LA APLICACIÓN

## 

# PRUEBAS

## Visión general

## Log

## Test

## Rendimiento de la base de datos

# MANUAL DE USUARIO

## Visión General

# CONCLUSIONES Y LÍNEAS FUTURAS

## Visión general

## Conclusiones

## Líneas futuras

# BIBLIOGRAFÍA

1. Formato cita web: “Titulo/Nombre de la Web”, http://www.google.es, (consultado el 7/10/12).