



**Escuela Técnica Superior de Ingeniería Informática**

**POYECTO FIN DE CARERA**

# **DISEÑO Y DESARROLLO DE UN SISTEMA DE RESERVAS PARA UNA EMPRESA DE TAXIS**

Para optar a la titulación de

**Ingeniero Técnico en Informática de Gestión**

Presentado por

**Pablo Tenorio Garcia**

Dirigido por

**Sergio Saez Barona**

# **MEMORIA FINAL RESERVATAXI:**

**Diseño y desarrollo de un sistema  
de reservas para una empresa de Taxis**

# Índice de contenido

|  |    |
|--|----|
| 1.-INTRODUCCIÓN.....                                       | 5  |
| 1.1.-Objetivos del sistema.....                            | 7  |
| 1.2.-Alcance del sistema.....                              | 8  |
| 2.- ESPECIFICACIÓN DE REQUISITOS.....                      | 9  |
| 2.1.- Introducción.....                                    | 10 |
| 2.1.1.- Proposito.....                                     | 10 |
| 2.1.2.- Ámbito.....  | 10 |
| 2.1.3.-Definiciones, Acrónimos y Abreviaturas.....         | 10 |
| 2.1.2.- Referencias.....                                   | 11 |
| 2.2.- Descripción General.....                             | 11 |
| 2.2.1.- Perspectiva del producto.....                      | 11 |
| 2.2.2.- Funciones del producto.....                        | 11 |
| 2.2.3.- Características del usuario.....                   | 11 |
| 2.2.4.- Restricciones generales.....                       | 12 |
| 2.3.- Requisitos Especificos.....                          | 12 |
| 2.3.1.-Diagrama de contexto DFD.....                       | 12 |
| 2.2.-Requisitos del Sistema.....                           | 13 |
| 2.3.2-Requisitos de datos del sistema web de registro..... | 14 |
| 2.3.3.-Requisitos de datos del sistema API de reserva..... | 15 |
| 3.-ANALISIS.....   | 16 |
| 3.1.-Introducción.....                                     | 17 |
| 3.2.-Planificación temporal.....                           | 17 |
| 3.3.-Modelado de Procesos.....                             | 24 |
| 3.4.-Diagrama de contexto.....                             | 24 |
| 3.5.-Diagrama de primer nivel.....                         | 25 |
| 3.5.1.-Almacenes.....                                      | 27 |
| 3.6.-Diagrama de segundo nivel.....                        | 27 |
| 3.6.1.-Gestión de reservas.....                            | 28 |
| 3.7.-Detalle de los Casos de Uso del apartado web.....     | 30 |
| 3.8.-Casos de Uso del sistema API de reserva.....          | 32 |
| 3.9.-Modelado de datos.....                                | 35 |
| 3.9.1.-Descripción de Entidades.....                       | 36 |
| 3.9.2.-Descripción Relacionales.....                       | 37 |
| 3.9.3.-Reglas de negocio.....                              | 37 |
| 4.-DISEÑO.....   | 38 |
| 4.1.-Introducción.....                                     | 38 |
| 4.2.-Diseño de la base de datos.....                       | 40 |
| 4.2.1.-Diseño lógico.....                                  | 40 |
| 4.2.1.1.-Tablas del diseño lógico.....                     | 40 |
| 4.2.1.2.- Stored Procedure.....                            | 40 |
| 4.2.1.2.- Descripción de atributos .....                   | 40 |
| 4.2.2.-Diseño físico .....                                 | 42 |
| 4.2.2.1.-Integración de la Stored Procedure.....           | 43 |
| 4.3.-Diseño de las interfaces de usuario.....              | 45 |
| 4.3.1.- Interfaces de usuario WEB.....                     | 45 |
| 4.3.2.- Interfaces de usuario del sistema móvil.....       | 47 |

|   |    |
|---|----|
| 4.4.-Diseño de procesos.....                                | 51 |
| 4.4.1.-Diseño de procesos WEB.....                          | 51 |
| 4.4.1.-Diseño de procesos API y Software Móvil.....         | 53 |
| 5.- DETALLES DE IMPLEMENTACIÓN.....                         | 54 |
| 5.1.-Tecnologías empleadas.....                             | 55 |
| 5.2.-Arquitectura final del sistema.....                    | 56 |
| 5.5.-Presupuesto del proyecto.....                          | 58 |
| 5.6.-Implementación.....                                    | 60 |
| 5.6.1.-Modulo WEB.....                                      | 60 |
| 5.6.2.-Modulo API.....                                      | 60 |
| 5.6.3.-Modulo Aplicación móvil J2ME.....                    | 61 |
| 6.-PRUEBAS, PUESTA EN MARCHA .....                          | 62 |
| 6.1.- Pruebas.....  | 63 |
| 6.1.1.-Creación de usuario, envío de sms e instalación..... | 63 |
| 6.1.2.-Reserva de taxi.....                                 | 65 |
| 6.2.- Puesta en marcha.....                                 | 68 |
| 7.-CONCLUSION.....  | 69 |
| 7.1.- Conclusión.....                                       | 70 |
| 8.-BIBLIOGRAFIA.....  | 72 |

# **1.-INTRODUCCIÓN**

La propuesta técnica del proyecto constituye la base para el desarrollo del sistema informático. En dicha propuesta se describe que es lo que hará el sistema, cuál es el trabajo que se va a realizar para la elaboración del proyecto, cuándo se realizará dicho trabajo y que costes acarreará su realización.

El sistema propuesto se llama “ReservaTaxi: Diseño y desarrollo de un sistema de reservas para una empresa de taxis” y va a permitir a un usuario ya existente en una compañía de taxis, reservar un taxi desde su terminal móvil (teléfono), mediante la selección de sus 5 últimas direcciones de recogida.

El proyecto se ha realizado físicamente en las oficinas de Techideas, una empresa de ingeniería del software en el parque tecnológico BCNord, en Barcelona. Y se ha vendido a Interfacom, cliente asiduo de Techideas, encargado de taxímetros y sistemas de localización y control de flotas en Europa, usando su marca Taxitronic.

En este documento se incluyen los objetivos y el alcance del sistema que nos permitirán determinar cuáles son los principales puntos a cubrir por el sistema; los requisitos del sistema, que son la base para el desarrollo posterior del sistema; la planificación inicial del proyecto que nos permitirá especificar las actividades que se llevarán a cabo para el desarrollo del proyecto, su duración y su prioridad; la estimación de costes que nos dará una aproximación del coste económico final que tendrá el proyecto y el diagrama de contexto del sistema.

En un primer paso se analizará la especificación de requisitos, donde se describirá el comportamiento del sistema, incluyendo un conjunto de casos de uso que describirán todas las interacciones que tendrán los usuarios con el software.

Posteriormente, se realizará un análisis del sistema a desarrollar, donde se mostrarán los procesos y los datos que interaccionan con el sistema.

El diseño será el siguiente aspecto que se trata. Aquí se conectará la visión lógica desarrollada en el análisis con la actividad propia de la programación.

Posteriormente se mostrarán los detalles de la implementación seguido de las pruebas, puesta en marcha y la conclusión.

## **1.1.-Objetivos del sistema**

El objetivo del sistema informático es proporcionar la funcionalidad necesaria para que el usuario final de una compañía de taxis, sea capaz de reservar un taxi en una de las últimas direcciones desde donde lo ha hecho últimamente, o si no encontrara una solución, llamar a la operadora sin tener que recordar el número de teléfono de su radiotaxi favorito.

Esta reserva se realiza a través de Internet, facilitando el proceso, evitando esperas y obteniendo un resultado instantáneo (ya sea positivo o negativo).

Este mismo sistema, puede ser la base para dar un paso mas y realizar la reserva de taxis usando un mapa con geocodificación, pero en este caso, desgraciadamente, no daremos una solución tan compleja y completa.

## **1.2.-Alcance del sistema**

En cuánto al alcance organizacional, el sistema mantendrá interacción con los usuarios finales, que usarán el software en el terminal móvil y con el administrador del sistema en la empresa de taxis, que podrá comprobar estadísticas, comprobar datos de usuarios, etc...

Por lo que se refiere al alcance informático, el sistema es prácticamente automático. El usuario final usará su teléfono para pedir primero la lista de las últimas direcciones de recogida y posteriormente seleccionar dicha dirección.

En cuanto al alcance funcional, el sistema constará de tres partes con funciones diversas:

1. Web de aprovisionamiento, alojada en el servidor ReservaTaxi, donde el usuario se registrará y descargará el programa para el móvil.
2. Software del terminal móvil, desde donde el usuario podrá reservar taxis.
3. API (Application Programming Interface), que comunicará nuestro servidor con el servidor central de la empresa para obtener la lista de direcciones y reservar el taxi.

## **2.- ESPECIFICACIÓN DE REQUISITOS**



## **2.1.- Introducción**

Esta especificación tiene como objetivo analizar y documentar las necesidades funcionales que deberán ser soportadas por el sistema a desarrollar. Para ello, se identificarán los requisitos que ha de satisfacer el nuevo sistema mediante entrevistas, el estudio de los problemas de las unidades afectadas y sus necesidades actuales. Además de identificar los requisitos se deberán establecer prioridades, lo cual proporciona un punto de referencia para validar el sistema final que compruebe que se ajusta a las necesidades del usuario.

### **2.1.1.- Proposito**

El propósito es definir cuales son los requerimientos que debe tener un programa que gestione la reserva de taxis, para usuarios vip de una empresa, en este caso, RT033. Con este sistema se liberan muchos puestos de operadora que estaban pendientes de ofrecer los destinos típicos de los clientes vip.

### **2.1.2.- Ámbito**

El producto que vamos a describir es un programa que facilitará la petición de taxis a los usuarios vip de esta empresa. Se llama RESERVATAXI.

Este producto debe ser capaz de adaptarse a la mayoría de terminales móviles (smartphones).

El servidor debe estar permanentemente conectado al servidor central de la empresa de taxis y con conexión también permanente a internet para que los usuarios puedan hacer uso de sus servicios.

### **2.1.3.-Definiciones, Acrónimos y Abreviaturas**

Reserva: acción de pedir un taxi

Servidor local: máquina donde se encuentra nuestro software

Servidor cliente: máquina donde se encuentran todas las bases de datos de los taxis, su localización, las peticiones, etc

API: interfaz de programación de aplicaciones (Application Programming Interface). En nuestro caso, es la encargada de recibir todas las peticiones desde los terminales móviles, y comunicarse con el servidor del cliente.

## **2.1.4.- Referencias**

ASTM E1340-96, Standard Guide for Rapid Prototyping of Computerized Systems.  
IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.  
IEEE Std 730-1998, IEEE Standard for Software Quality Assurance Plans.  
IEEE Std 730.1-1995, IEEE Guide for Software Quality Assurance Planning.  
IEEE Std 828-1998, IEEE Standard for Software Configuration Management Plans.  
IEEE Std 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software.  
IEEE Std 982.2-1988, IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software.  
IEEE Std 1002-1987 (Reaff 1992), IEEE Standard Taxonomy for Software Engineering Standards.  
IEEE Std 1012-1998, IEEE Standard for Software Verification and Validation.  
IEEE Std 1012a-1998, IEEE Standard for Software Verification and Validation: Content Map to IEEE/EIA 12207.1-1997.  
IEEE Std 1016-1998, IEEE Recommended Practice for Software Design Descriptions.  
IEEE Std 1028-1997, IEEE Standard for Software Reviews.  
IEEE Std 1042-1987 (Reaff 1993), IEEE Guide to Software Configuration Management.  
IEEE P1058/D2.1, Draft Standard for Software Project Management Plans, dated 5 August 1998.  
IEEE Std 1058a-1998, IEEE Standard for Software Project Management Plans: Content Map to IEEE/EIA 12207.1-1997.  
IEEE Std 1074-1997, IEEE Standard for Developing Software Life Cycle Processes.  
IEEE Std 1233, 1998 Edition, IEEE Guide for Developing System Requirements

## ***2.2.- Descripción General***

### **2.2.1.- Perspectiva del producto**

La aplicación RESERVATAXI debe aumentar la eficacia en la gestión de la reserva de taxis, facilitar al usuario su uso e incluso ahorrar en gastos.

## **2.2.2.- Funciones del producto**

Las funciones que debe realizar el producto las podemos calificar en varios bloques:

### A) Creación de usuarios

- A1) Registro de usuario
- A2) Descarga de software para el terminal apropiado
- A2) Instalación del software

### B) Reserva de un taxi

- B1) Generar una lista con las direcciones habituales
- B2) Crear una reserva del taxi en la dirección deseada.

## **2.2.3.- Características del usuario**

A pesar de que cualquier persona se puede descargar el software en su terminal, solo funcionara con aquellos que sean clientes vip de la empresa de taxis. Además, siendo vip, se necesita de un terminal smartphone con conexión a Internet, ya sea por 3G o por wifi, aunque preferiblemente por 3g.

## **2.2.4.- Restricciones generales**

Dada la amplitud de smartphones del mercado, se ha logrado abarcar la mayoría, faltando solo uno, el iPhone. Pero dando servicio a terminales con symbian, android y blackberry, gracias al J2ME Polish

## **2.3.- Requisitos Especificos**

### **2.3.1.-Diagrama de contexto DFD**

A continuación se presenta el diagrama que nos permitirá visualizar el subsistema propuesto con las diferentes entidades externas con los que interactuará.



Antes de describir el contenido del diagrama, resulta conveniente tener en cuenta que todos los nombres de los flujos de datos (flechas) hacen referencia a transacciones de información entre el subsistema y las entidades externas con las que se comunica; aunque el nombre de cada uno de estos flujos no lo indiquen en algunos casos.

Cada uno de los distintos flujos de datos están separados por comas bajo una misma flecha; en realidad cada flujo debería de estar representado por una flecha, pero para facilitar la comprensión del diagrama y no cargarlo demasiado, se han juntado aquellas que van de una misma entidad a otra en un mismo sentido.

La principal entidad externa que interactúa con el subsistema son los **clientes**. Los clientes serán los usuarios finales del sistema, sus interacciones con el subsistema son las siguientes:

- **Registrarse como nuevo cliente del subsistema:** Antes de realizar cualquier tipo de operación habrá que realizar un registro en el subsistema. Este proceso consiste simplemente en introducir el número de teléfono del cliente y la única finalidad es la descarga del software. Usando el número de teléfono introducido, el servidor enviará un mensaje de texto con un link de descarga y un código de activación. Cuando el usuario introduce el link de descarga, se hace una llamada a la API, esta detecta que tipo de terminal hace la llamada y envía el paquete de instalación oportuno. Si no existe una versión compatible avisará con un mensaje de error.
- **Obtener lista de direcciones favoritas y número de teléfono:** Una vez el software ha sido instalado correctamente, automáticamente al iniciar el programa, se hace una llamada a la API para conocer la lista de direcciones favoritas y el número de teléfono de la empresa deseada.

- **Reservar un taxi:** Seleccionando una dirección desde el menú de la aplicación se enviará la petición a nuestro servidor.
  
- **Obtener lista de direcciones favoritas y número de teléfono:** Al iniciar el software en el terminal, se hace una llamada a la API para conocer la lista de direcciones asignada a ese usuario. La API recolecta la información enviada, la procesa y la reenvía al servidor de taxis, obteniendo como respuesta esa lista, creada a su criterio.
  
- **Reservar un taxi:** Se manda una petición al servidor central de la empresa de taxi seleccionada y esta nos devuelve el estado de la petición, que será reenviada al terminal del cliente.

## 2.2.-Requisitos del Sistema

A continuación se presentarán mediante unas sencillas tablas explicativas la información con la que se va a tratar en el subsistema.

En cada una de las tablas aparecerá el código del requisito de datos, el nombre del requisito, versión, los requisitos asociados con los cuales tienen relación; la descripción del requisito, los datos que contendrá, su importancia, la cual puede ser; imprescindible, interesante, superfluo o por determinar, y la procedencia de los datos del requisito.

### 2.3.2-Requisitos de datos del sistema web de registro

|                      |  |
|----------------------|--|
| RD-W01               | Usuarios   |
| Versión              | 1.0  |
| Requisitos asociados | Radio Taxis  |
| Descripción          | El sistema almacenará los datos básicos del usuario.   |
| Datos específicos    | <ul style="list-style-type: none"><li>● Id</li><li>● Numero de Teléfono</li><li>● Radio Taxi</li><li>● Fecha de Registro</li></ul> |
| Procedencia          | Externa (del usuario que se registra)<br>Interna (Auto generada por el subsistema)   |

|                      |   |
|----------------------|---|
| RD-W02               | Radio Taxis   |
| Versión              | 1.0   |
| Requisitos asociados |   |
| Descripción          | Diferentes Radio Taxis de la zona   |
| Datos específicos    | <ul style="list-style-type: none"><li>● Radio Taxi ID</li><li>● Radio Taxi Empresa</li><li>● Número de Teléfono</li></ul> |
| Procedencia          | Interna (Datos fijos)   |

|                      |   |
|----------------------|---|
| RD-W03               | Descargas   |
| Versión              | 1.0   |
| Requisitos asociados | Usuarios, Radio Taxis   |
| Descripción          | Datos para estadísticas y activación de servicios   |
| Datos específicos    | <ul style="list-style-type: none"> <li>● Usuario</li> <li>● Link</li> <li>● Fecha Descarga</li> <li>● Modelo Teléfono</li> <li>● Radio Taxi ID</li> <li>● Radio Taxi Empresa</li> <li>● Numero Teléfono Empresa RT</li> </ul> |
| Procedencia          | Interna (Datos fijos)   |

### 2.3.3.-Requisitos de datos del sistema API de reserva

|                      |  |
|----------------------|--|
| RD-A01               | Identificador Calle  |
| Versión              | 1.0  |
| Requisitos asociados |  |
| Descripción          | Entero que identifica la calle en el servidor de taxis central |
| Datos específicos    | <ul style="list-style-type: none"> <li>● ID Calle</li> </ul>   |
| Procedencia          | Interna (Datos fijos)  |

|                      |   |
|----------------------|---|
| RD-A02               | Numero de Calle   |
| Versión              | 1.0   |
| Requisitos asociados |   |
| Descripción          | Entero que identifica el numero de la calle en el servidor de taxis central |
| Datos específicos    | <ul style="list-style-type: none"> <li>● Numero Calle</li> </ul>            |
| Procedencia          | Interna (Datos fijos)   |

# **3.-ANALISIS**



### 3.1.-Introducción

Una vez conocida la propuesta técnica del proyecto ReservaTaxis, en éste apartado del proyecto vamos a centrarnos más en los detalles del sistema mediante un análisis exhaustivo; tanto en los procesos como en los datos que van a aparecer e interaccionar en el subsistema.

Antes de mostrar los modelados, se ofrece una vista de la planificación temporal del proyecto.

Para explicar el análisis del subsistema de un modo ordenado, se empezará por mostrar y comentar los distintos procesos que interactuarán tanto entre el sistema como fuera de él, los almacenes, y los flujos de datos que permiten la comunicación entre los componentes; a este apartado se le conoce como el **modelado de procesos**. El contenido de este apartado se mostrará de forma ordenada, de menor a mayor detalle.

Una vez conocido el modelado de procesos se presentará el análisis de los datos que tendrán relevancia en el subsistema, agrupados en entidades, y las relaciones que existen entre ellos. A este apartado se le conoce como **modelado de datos**.

Una vez conocido de antemano el contenido del análisis del proyecto se procederá a presentar el modelado de los procesos mediante diagramas y documentación relacionada.

### 3.2.-Planificación temporal

Para disponer de un control temporal en el desarrollo del proyecto, se han detallado las tareas, recursos e hitos a conseguir en el transcurso del tiempo. La presente tabla muestra al desarrollador del proyecto en sus distintos roles, y el coste por hora que cobrará en el desarrollo.

| Nombre        | Grupo       | E-Mail             | Coste |
|---------------|-------------|--------------------|-------|
| Pablo Tenorio | Analista    | pabtegar@ei.upv.es | 40    |
| Pablo Tenorio | Programador | pabtegar@ei.upv.es | 30    |

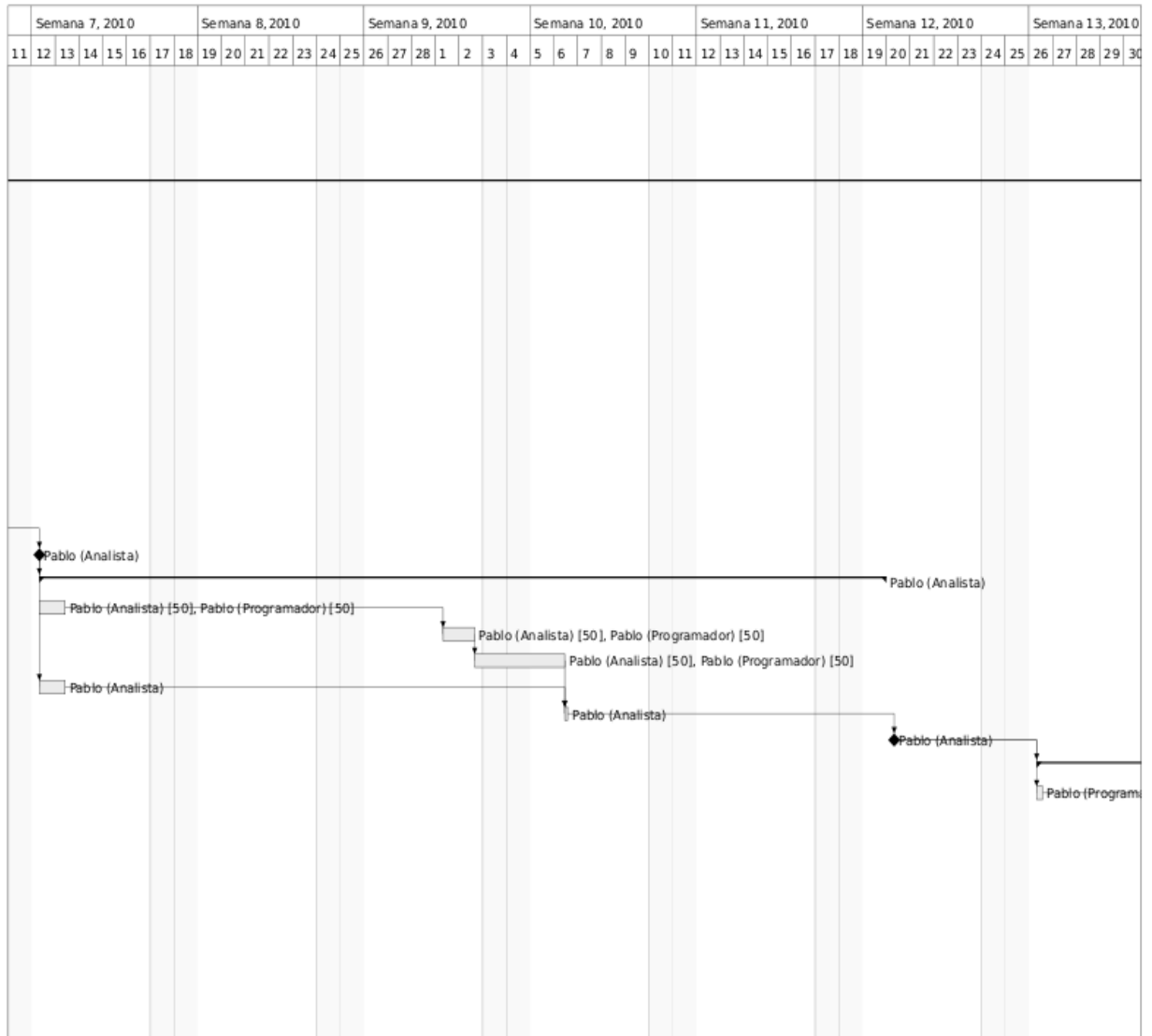
La siguiente tabla a presentar muestra cada una de las tareas a realizar en el desarrollo del proyecto, unas de las columnas con información más relevante es la de "Coste" de cada uno de los procesos, los cuales se tienen que tener en cuenta con la tabla anterior.

| WBS   | Nombre  | Inicio        | Fin           | Trabajo       | Duración       | Desperdicio    | Coste          |
|-------|---|---------------|---------------|---------------|----------------|----------------|----------------|
| 1     | <b>Proyecto informático</b>                             | <b>nov 27</b> | <b>nov 27</b> | <b>1d</b>     | <b>7h</b>      | <b>144d 1h</b> | <b>320</b>     |
| 1.1   | <b>Estudio inicial del proyecto</b>                     | <b>nov 27</b> | <b>nov 27</b> | <b>1d</b>     | <b>7h</b>      | <b>144d 1h</b> | <b>320</b>     |
| 1.1.1 | Lectura y comprensión de la propuesta                   | nov 27        | nov 27        | 3h            | 3h             | 123d 5h        | 120            |
| 1.1.2 | Búsqueda inicial de objetivos y alcance                 | nov 27        | nov 27        | 5h            | 5h             | 123d 2h        | 200            |
| 2     | <b>Desarrollo del proyecto</b>                          | <b>nov 27</b> | <b>jun 18</b> | <b>27d 3h</b> | <b>144d 1h</b> |                | <b>7775,61</b> |
| 2.1   | <b>Análisis de requisitos</b>                           | <b>nov 27</b> | <b>dic 4</b>  | <b>4d 6h</b>  | <b>4d 5h</b>   | <b>139d 3h</b> | <b>1520,61</b> |
| 2.1.1 | Definición del alcance y de los objetivos               | nov 27        | nov 29        | 1d 2h         | 1d 7h          | 122d 3h        | 400,4          |
| 2.1.2 | Documentación de los requisitos de datos                | nov 29        | nov 30        | 7h            | 7h             | 121d 4h        | 280            |
| 2.1.3 | Documentación de los requisitos funcionales             | nov 29        | nov 30        | 1d 2h         | 1d 2h          | 121d 2h        | 400            |
| 2.1.4 | Diagrama de contexto                                    | nov 30        | dic 1         | 5h            | 7h             | 121d 3h        | 200,2          |
| 2.1.5 | Elaboración de la planificación temporal y los recursos | nov 29        | nov 30        | 5h            | 5h             | 122d 6h        | 200            |
| 2.1.6 | Revisión con el tutor y mejoras de los requisitos       | dic 1         | dic 4         | 1h            | 7h             | 123d 3h        | 40,01          |
| 2.1.7 | H1: Fin del análisis de requisitos                      | dic 4         | dic 3         | N/D           | N/D            | 124d           | 0              |
| 2.2   | <b>Análisis del sistema</b>                             | <b>dic 18</b> | <b>feb 11</b> | <b>4d</b>     | <b>40d</b>     | <b>90d</b>     | <b>1280</b>    |
| 2.2.1 | Creación de diagramas de flujo de datos                 | dic 18        | dic 19        | 1d 7h         | 1d 7h          | 113d 3h        | 600            |
| 2.2.2 | Creación del modelo conceptual                          | ene 8         | ene 9         | 1d 2h         | 1d 2h          | 98d 7h         | 400            |
| 2.2.3 | Documentación del análisis                              | ene 22        | ene 22        | 5h            | 5h             | 90d 4h         | 200            |
| 2.2.4 | Revisión con el tutor y mejoras del análisis            | feb 1         | feb 1         | 2h            | 2h             | 82d 6h         | 80             |
| 2.2.5 | H2: Fin del análisis                                    | feb 12        | feb 11        | N/D           | N/D            | 76d            | 0              |
| 2.3   | <b>Diseño del sistema</b>                               | <b>feb 12</b> | <b>mar 20</b> | <b>6d 1h</b>  | <b>26d</b>     | <b>64d</b>     | <b>1775</b>    |
| 2.3.1 | Diseño de los procesos y módulos del sistema            | feb 12        | feb 13        | 1d 2h         | 1d 2h          | 77d 4h         | 350            |
| 2.3.2 | Diseño y creación de las interfaces                     | mar 1         | mar 2         | 1d 7h         | 1d 7h          | 65d 3h         | 525            |
| 2.3.3 | Diseño de la base de datos                              | mar 2         | mar 6         | 1d 4h         | 1d 4h          | 65d 7h         | 420            |
| 2.3.4 | Documentación del diseño                                | feb 12        | feb 13        | 1d 2h         | 1d 2h          | 80d 6h         | 400            |
| 2.3.5 | Revisión con el tutor y mejora del diseño               | mar 6         | mar 6         | 2h            | 2h             | 65d 5h         | 80             |
| 2.3.6 | H3: Fin del diseño del sistema                          | mar 20        | mar 20        | N/D           | N/D            | 56d            | 0              |
| 2.4   | <b>Construcción y puesta en marcha</b>                  | <b>mar 26</b> | <b>jun 4</b>  | <b>10d 3h</b> | <b>50d</b>     | <b>10d</b>     | <b>2520</b>    |
| 2.4.1 | Puesta en marcha de la base de datos                    | mar 26        | mar 26        | 5h            | 5h             | 52d 1h         | 150            |
| 2.4.2 | Programación de los procesos                            | abr 2         | abr 6         | 5d            | 5d             | 45d 4h         | 1200           |
| 2.4.3 | Pruebas de integridad entre módulos                     | abr 6         | abr 10        | 2d 4h         | 2d 4h          | 47d 3h         | 600            |
| 2.4.4 | Documentación de las pruebas                            | abr 6         | abr 10        | 1d 7h         | 1d 7h          | 47d 5h         | 450            |
| 2.4.5 | Revisión con el tutor y mejora de implementación        | may 1         | may 1         | 3h            | 3h             | 33d 3h         | 120            |
| 2.4.6 | H4: Fin de la construcción                              | jun 4         | jun 4         | N/D           | N/D            | 9d 6h          | 0              |
| 2.5   | <b>Redacción de la memoria final</b>                    | <b>jun 4</b>  | <b>jun 18</b> | <b>2d 1h</b>  | <b>10d</b>     |                | <b>680</b>     |
| 2.5.1 | Estructuración de documentos de cada clase              | jun 4         | jun 4         | 5h            | 5h             | 9d 3h          | 200            |
| 2.5.2 | Redacción   | jun 4         | jun 5         | 1d 2h         | 1d 2h          | 8d 1h          | 400            |
| 2.5.3 | Entrega y exposición                                    | jun 5         | jun 6         | 2h            | 2h             | 7d 7h          | 80             |
| 2.5.4 | H5: Fin del proyecto                                    | jun 18        | jun 18        | N/D           | N/D            |                | 0              |

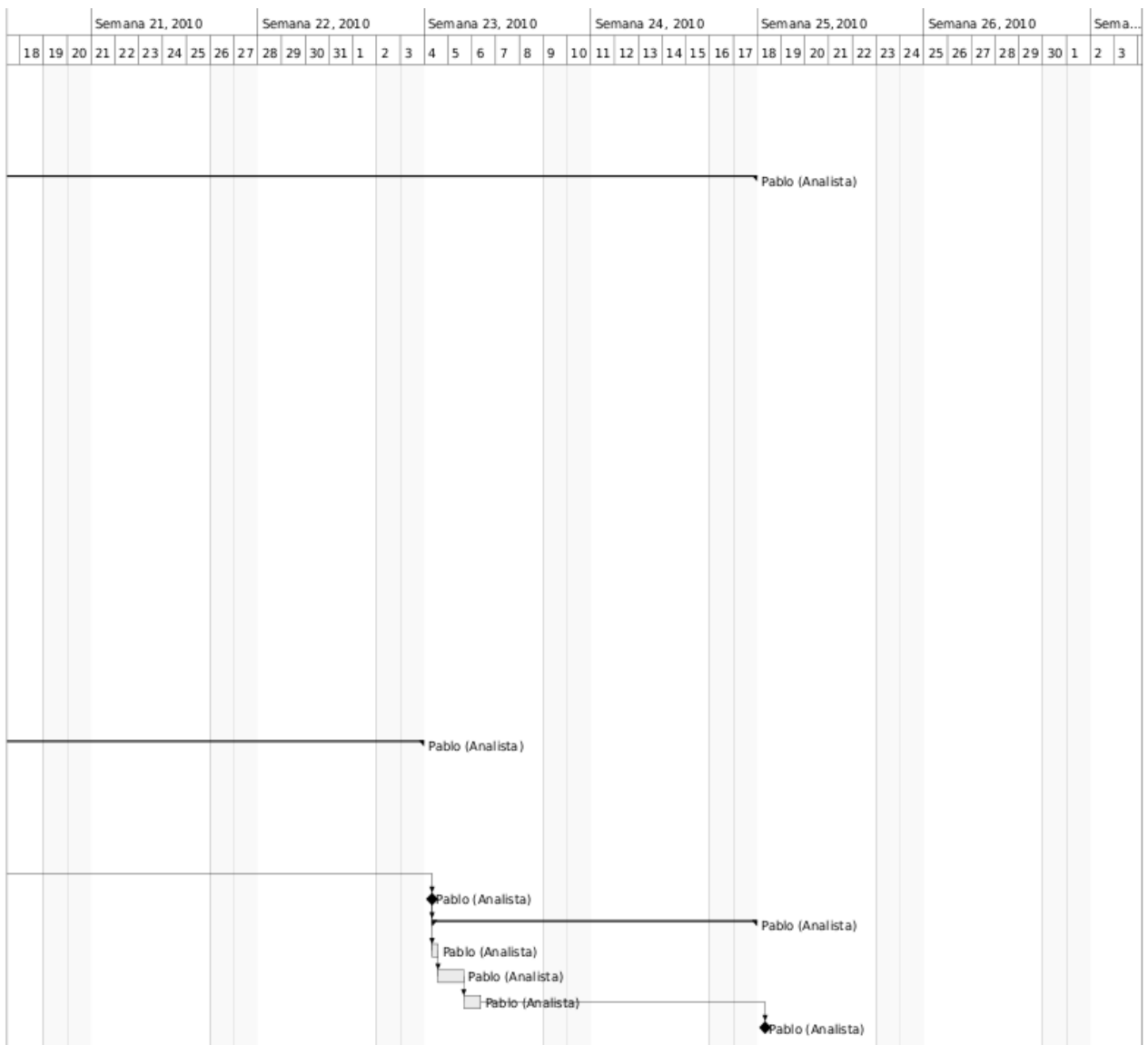
Una vez presentada la tabla de las tareas se presentará a continuación un diagrama de Gantt fragmentado, en el que, de forma gráfica, se puede apreciar los periodos de desarrollo asignados en un principio para el proyecto.

| Nombre  | Trab...       | Semana 48, 2009       |    |    |    |   | Semana 49, 2009 |   |   |   |   | Semana 50, 2009 |   |   |    |    | Semana 51, 2009 |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---------------|-----------------------|----|----|----|---|-----------------|---|---|---|---|-----------------|---|---|----|----|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|
|   |               | 27                    | 28 | 29 | 30 | 1 | 2               | 3 | 4 | 5 | 6 | 7               | 8 | 9 | 10 | 11 | 12              | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| <b>Proyecto informático</b>                             | <b>1d</b>     | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>Estudio inicial del proyecto</b>                     | <b>1d</b>     | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Lectura y comprensión de la propuesta                   | 3h            | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Búsqueda inicial de objetivos y alcance                 | 5h            | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>Desarrollo del proyecto</b>                          | <b>27d 3h</b> | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>Análisis de requisitos</b>                           | <b>4d 6h</b>  | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Definición del alcance y de los objetivos               | 1d 2h         | Pablo (Analista) [63] |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Documentación de los requisitos de datos                | 7h            | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Documentación de los requisitos funcionales             | 1d 2h         | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Diagrama de contexto                                    | 5h            | Pablo (Analista) [63] |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Elaboración de la planificación temporal y los recursos | 5h            | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Revisión con el tutor y mejoras de los requisitos       | 1h            | Pablo (Analista) [13] |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| H1: Fin del análisis de requisitos                      |               | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>Análisis del sistema</b>                             | <b>4d</b>     | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Creación de diagramas de flujo de datos                 | 1d 7h         | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Creación del modelo conceptual                          | 1d 2h         | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Documentación del análisis                              | 5h            | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Revisión con el tutor y mejoras del análisis            | 2h            | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| H2: Fin del análisis                                    |               | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>Diseño del sistema</b>                               | <b>6d 1h</b>  | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Diseño de los procesos y módulos del sistema            | 1d 2h         | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Diseño y creación de las interfaces                     | 1d 7h         | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Diseño de la base de datos                              | 1d 4h         | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Documentación del diseño                                | 1d 2h         | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Revisión con el tutor y mejora del diseño               | 2h            | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| H3: Fin del diseño del sistema                          |               | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>Construcción y puesta en marcha</b>                  | <b>10d 3h</b> | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Puesta en marcha de la base de datos                    | 5h            | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Programación de los procesos                            | 5d            | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Pruebas de integridad entre módulos                     | 2d 4h         | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Documentación de las pruebas                            | 1d 7h         | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Revisión con el tutor y mejora de implementación        | 3h            | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| H4: Fin de la construcción                              |               | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| <b>Redacción de la memoria final</b>                    | <b>2d 1h</b>  | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Estructuración de documentos de cada clase              | 5h            | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Redacción   | 1d 2h         | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| Entrega y exposición                                    | 2h            | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |
| H5: Fin del proyecto                                    |               | Pablo (Analista)      |    |    |    |   |                 |   |   |   |   |                 |   |   |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |

|    |    |    |    |    |    | Semana 1, 2010 |   |   |   |   | Semana 2, 2010 |   |   |   |    | Semana 3, 2010 |    |    |    |    | Semana 4, 2010 |    |    |    |    | Semana 5, 2010 |    |    |    |    | Semana 6, 2010 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |
|----|----|----|----|----|----|----------------|---|---|---|---|----------------|---|---|---|----|----------------|----|----|----|----|----------------|----|----|----|----|----------------|----|----|----|----|----------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----|
| 26 | 27 | 28 | 29 | 30 | 31 | 1              | 2 | 3 | 4 | 5 | 6              | 7 | 8 | 9 | 10 | 11             | 12 | 13 | 14 | 15 | 16             | 17 | 18 | 19 | 20 | 21             | 22 | 23 | 24 | 25 | 26             | 27 | 28 | 29 | 30 | 31 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|    |    |    |    |    |    |                |   |   |   |   |                |   |   |   |    |                |    |    |    |    |                |    |    |    |    |                |    |    |    |    |                |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |
|    |    |    |    |    |    |                |   |   |   |   |                |   |   |   |    |                |    |    |    |    |                |    |    |    |    |                |    |    |    |    |                |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |
|    |    |    |    |    |    |                |   |   |   |   |                |   |   |   |    |                |    |    |    |    |                |    |    |    |    |                |    |    |    |    |                |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |
|    |    |    |    |    |    |                |   |   |   |   |                |   |   |   |    |                |    |    |    |    |                |    |    |    |    |                |    |    |    |    |                |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |







### 3.3.-Modelado de Procesos

A continuación se presentarán los distintos diagramas del modelado de procesos de menor a mayor detalle, acompañados de la documentación adecuada en cada uno de los niveles. Para ello se utilizará una de las técnicas de modelización de procesos más utilizada en las metodologías estructuradas; los diagramas de flujos de datos (DFD).

Como se comentó en la propuesta técnica, antes de describir el contenido del diagrama, hay que tener en cuenta que todos los nombres de los flujos de datos (flechas) hacen referencia a transacciones de información entre el subsistema y las entidades externas con las que se comunica; aunque el nombre de cada uno de estos flujos no lo indiquen en algunos casos.

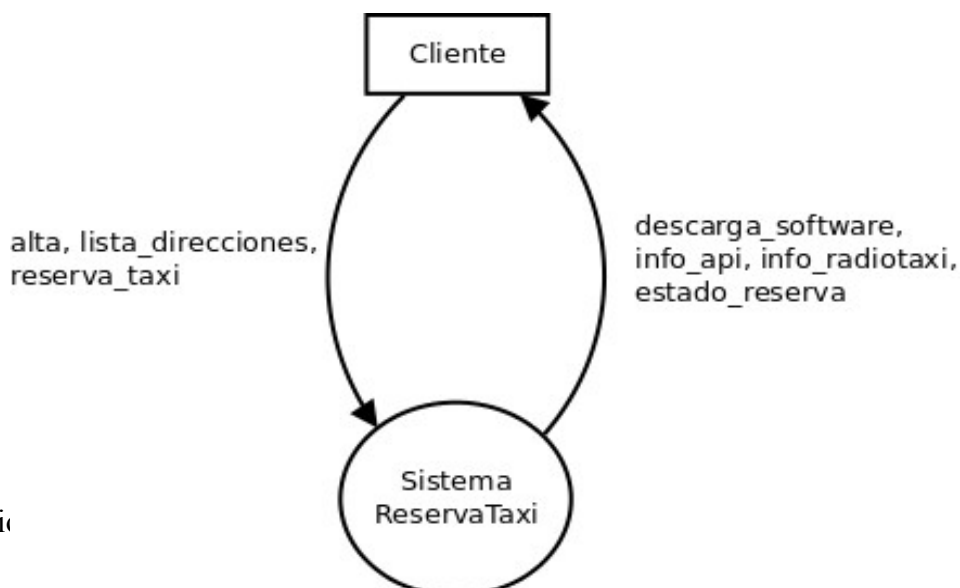
Cada uno de los distintos flujos de datos están separados por comas bajo una misma flecha; en realidad cada flujo debería de estar representado por una flecha, pero para facilitar la comprensión del diagrama y no cargarlo demasiado, se han juntado aquellas que van de una misma entidad a otra en un mismo sentido.

También hay que tener en cuenta que la descripción de los flujos de datos sólo se realizará cuando éstos se encuentren en procesos que no se vayan a explotar más, normalmente encontrados en los diagramas de mayor detalle. En éstos diagramas, cada flujo de datos estará representado por una única flecha.

Empezaremos presentando el primero de los diagramas, el cual ya apareció en la propuesta técnica.

### 3.4.-Diagrama de contexto

El diagrama de contexto es el primero de los diagramas del modelado de procesos, y por lo tanto el que contiene un menor nivel de detalle, con una mayor abstracción de la información.





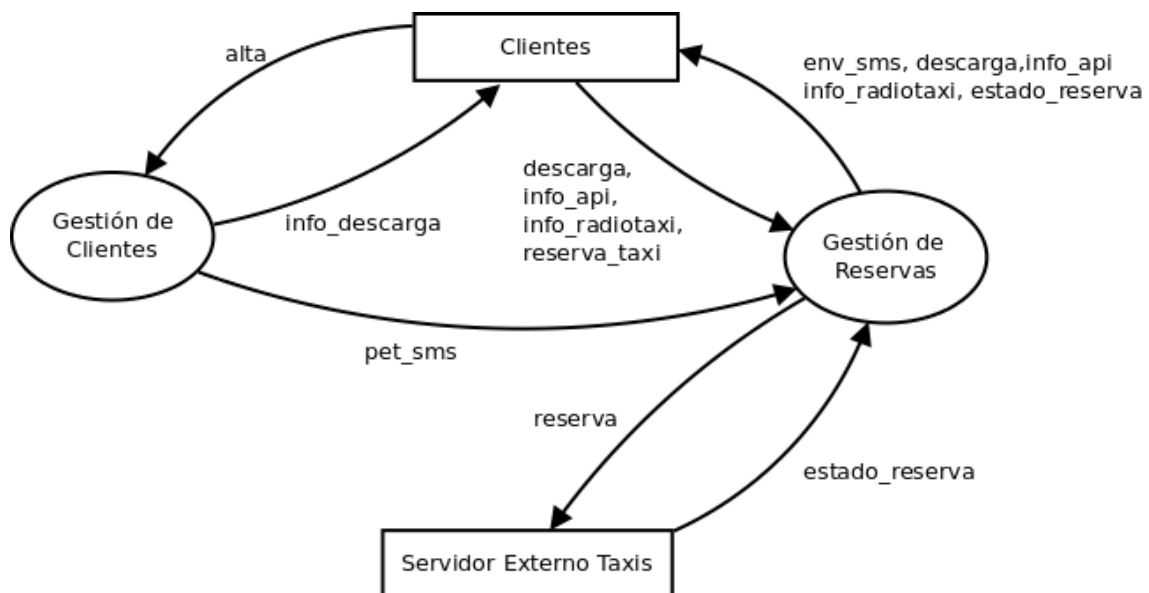
En el diagrama de contexto nos encontramos al subsistema ReservaTaxi y la entidad externa con las que se relaciona, los clientes.

En lo referente a los flujos de datos, de momento no se va a comentar ninguno, se irán comentando cada uno de ellos mediante tablas aclarativas, cuando aparezcan relacionados en procesos que no se vayan a explotar más.

Una vez conocido el diagrama de contexto se pasará a explotar el proceso principal ReservaTaxi para conocer un diagrama de primer nivel.

### 3.5.-Diagrama de primer nivel

En esta sección se presentará la primera explosión del diagrama de procesos, conocido también como el diagrama de primer nivel.



Iniciaremos la documentación del diagrama explicando los distintos procesos que aparecen:

1. Gestión de clientes: Permite gestionar el alta de clientes. El cliente se da de alta y el sistema le facilita un link desde donde se podrá descargar la aplicación.
2. Gestión de reservas: Gestiona las descargas y las reservas del cliente y hace de puente entre el cliente y el servidor externo de la empresa de taxis. En el

diagrama de segundo nivel "Gestión de reservas" veremos todos los procesos que lo componen.

A continuación pasaremos a conocer los flujos de datos que tienen relación con la gestión de clientes, ya que estos procesos no se van a detallar en el segundo nivel.

| Nombre        | Origen              | Destino             | Descripción  | Estructura               |
|---------------|---------------------|---------------------|--|--------------------------|
| alta          | Clientes            | Gestión de Clientes | Registro de nuevo usuario                                | Entero (Numero teléfono) |
| info_descarga | Gestión de Clientes | Clientes            | Información del registro, instrucciones.                 | Mensaje web              |
| pet_sms       | Gestión de Clientes | Gestión de Reservas | Petición de envío de sms a la API de gestión de reservas | Entero (numero teléfono) |

### 3.5.1.-Almacenes

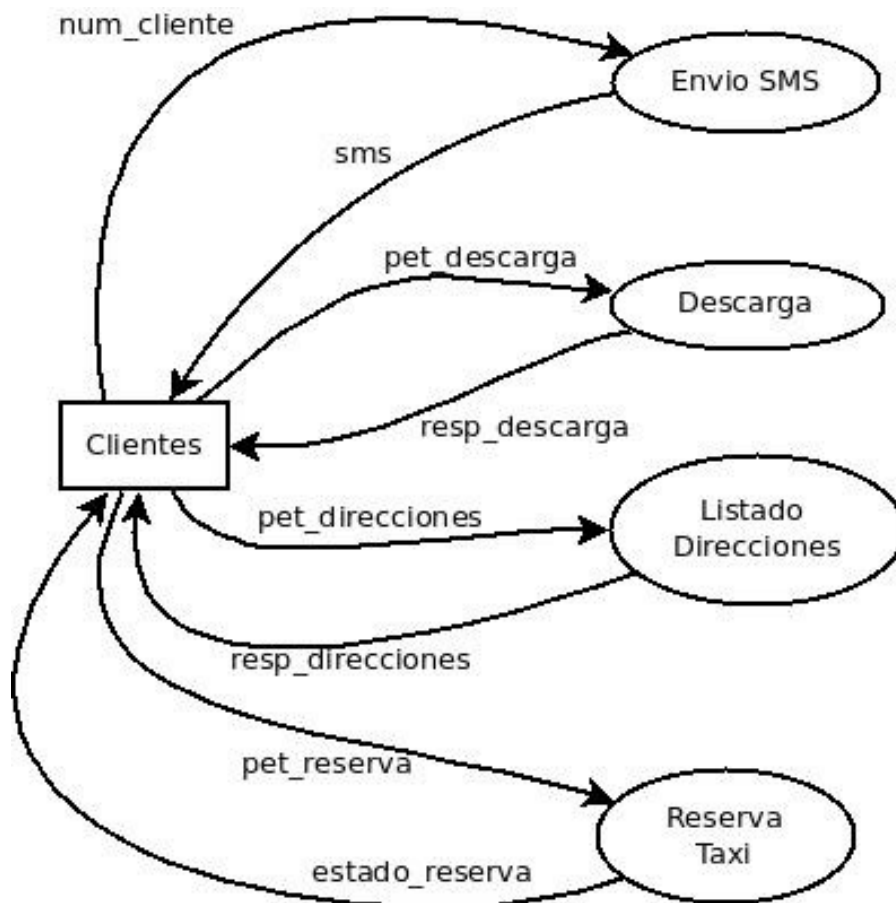
|             |   |
|-------------|---|
| Almacén     | Clientes  |
| Descripción | Este almacén guarda información de cada uno de los clientes que se dan de alta en el subsistema.  |
| Estructura  | Código de cliente + Numero de teléfono + fecha de registro  |
| Almacén     | Descargas   |
| Descripción | Este almacén guarda la información de cada descarga de usuario, con el fin de ver si han activado el software descargado, el modelo de teléfono utilizado y el link de descarga |
| Estructura  | Código cliente + link + activo + fecha de descarga + modelo teléfono  |

### 3.6.-Diagrama de segundo nivel

A continuación se presentará el diagrama de segundo nivel relacionado con la gestión de reservas por parte del subsistema.

#### 3.6.1.-Gestión de reservas

En esta sección se presentará la primera explotación del diagrama de gestión de reservas, perteneciente a los diagramas de segundo nivel.



Iniciaremos la documentación del diagrama explicando los distintos procesos que aparecen:

1 **Envío SMS:** Este proceso permite el envío de un sms al cliente una vez se ha registrado, para que este se descargue el software. Los pasos a seguir son los siguientes:

- Al registrarse el usuario en el proceso de gestión de clientes se llama a una función en la API y le pasa el número de teléfono del usuario.
- El servidor con la API dispone de un módem capaz de enviar SMS, así, manda un mensaje al cliente con el link de descarga.

2 **Descarga:** Proceso que permite la identificación del sistema operativo del terminal de usuario y la descarga del paquete de instalación. Los pasos son los siguientes:

- Cuando el cliente hace clic en el link enviado, se llama a una función (mdetect) en la API que detecta el sistema operativo del Terminal.
- Seguidamente se inicia la descarga del paquete de instalación correspondiente.

3 **Listado de direcciones:** Éste proceso se lanza automáticamente con la aplicación del móvil y permite obtener el listado de direcciones frecuentes desde donde el cliente suele pedir un taxi. Los pasos a seguir son los siguientes:

- Al iniciarse el software en el terminal, se hace una llamada automática al servidor con la API pasándole el número de teléfono como identificador de usuario.
- La API analiza los datos y transforma la llamada en una consulta a la base de datos Oracle del servidor de la empresa de Taxis.
- Ésta consulta devuelve una lista de direcciones
- La API devuelve al terminal mediante JSON dicha lista, que es mostrada por pantalla para que el usuario seleccione su preferida.

4 **Reserva Taxi:** Permite reservar un taxi en la dirección seleccionada. Pasos:

- El usuario selecciona una de las cinco direcciones favoritas
- Aparece una pantalla con la dirección seleccionada y un botón para confirmar la reserva y otro para cancelar y volver atrás.
- Si el usuario acepta la reserva, se manda la petición a la API.
- La API recoge la petición y la codifica para llamar a la función pertinente en el servidor de la empresa de taxis. Esta función devuelve un valor.
- La API gestiona el valor devuelto por el servidor de taxis y envía al terminal una de dos:
  - OK: Número de reserva
  - Descripción del error, que dependiendo del valor enviado por el servidor será un breve texto explicativo u otro.

A continuación se documentarán los flujos de datos que interaccionan con los subprocesos del proceso “Gestión de Reservas”.

| Nombre           | Origen              | Destino             | Descripción   | Estructura                    |
|------------------|---------------------|---------------------|---|-------------------------------|
| num_clinete      | Clientes            | Envio SMS           | Numero Cliente  | Entero (Numero teléfono)      |
| sms              | Envio SMS           | Clientes            | SMS con link de descarga                              | Mensaje de texto de telefonía |
| pet_descarga     | Clientes            | Descarga            | Llamada a la API para descargar                       |                               |
| resp_descarga    | Descarga            | Clientes            | Inicio Descarga                                       |                               |
| pet_direcciones  | Clientes            | Listado Direcciones | Pide la lista de direcciones favoritas                |                               |
| resp_direcciones | Listado Direcciones | Clientes            | Envia el listado de direcciones favoritas del cliente |                               |
| pet_reserva      | Clientes            | Reserva Taxi        | Solicita un taxi en la direccion seleccionada         |                               |
| estado_reserva   | Reserva Taxi        | Clientes            | Envia el estado de la reserva (ok o error)            |                               |

### **3.7.-Detalle de los Casos de Uso del apartado web**

En las siguientes tablas podemos encontrar; el código del requisito, su nombre, la

versión, su autor, sus requisitos asociados, una pre condición que se debe cumplir para tener el requisito, la secuencia de pasos a cumplir, una postcondición en la que quedará el subsistema después del requisito, las excepciones que pueden ocurrir en la ejecución del requisito, la importancia del requisito, y en algunos casos un breve comentario si procede.

|                      |   |  |
|----------------------|---|--|
| RF-W01               | Alta Cliente  |  |
| Versión              | 1.0   |  |
| Requisitos asociados | RD-A01 Usuarios<br>RD-A02 Radiotaxis<br>RD-A03 Descargas  |  |
| Descripción          | Inicio de todos el proceso. Se crea el usuario, se le asigna un radiotaxi y se graban sus datos para estadísticas y activaciones. |  |
| Pre condición        | El cliente no debe estar ya registrado, esto se comprueba mediante su numero de teléfono.   |  |
| Secuencia Normal     | <b>Paso</b>   | <b>Acción</b>  |
|                      | P1  | El cliente introduce su numero de teléfono   |
|                      | P2  | El sistema verifica que el imput tiene un formato correcto y que el usuario no existe ya   |
|                      | P3  | En caso de realizarse el alta correctamente, el sistema mostrará un mensaje con instrucciones y mandará un SMS con el link de descarga y la clave de activación. |
| Postcondición        | El sistema se queda pendiente de la confirmación del usuario y activará el mismo cuando introduzca la clave enviada por sms.      |  |
| Excepciones          | <b>Excepción</b>  | <b>Acción</b>  |
|                      | E1  | Si no están los campos rellenos correctamente se avisará de ello al cliente y se impedirá el alta hasta que se cumplimente de forma adecuada                     |
|                      | E2  | Si el cliente ya estaba dado de alta, se volverá a mandar un sms con una nueva clave de activación   |
|                      | E3  |  |

### 3.8.-Casos de Uso del sistema API de reserva

La siguiente pre condición es la misma para cada requisito funcional, ya que son llamadas desde el terminal móvil al servidor del proyecto:

La aplicación tiene que estar validada y el cliente asociado con el servidor central.

|                      |  |   |
|----------------------|--|---|
| RF-A01               | Versión Procesador   |   |
| Versión              | 1.0  |   |
| Requisitos asociados |  |   |
| Descripción          | Devuelve la versión del procesador de la API                           |   |
| Secuencia Normal     | Paso   | Acción  |
|                      | P1   | Al iniciarse el sistema en el terminal móvil, se pregunta a la API la versión de esta, para posibles actualizaciones. |
|                      | P2   | En caso de ser versiones distintas, se procede a la descarga del nuevo software.                                      |
| Postcondición        | Si se actualiza el software, se borra la versión anterior del sistema. |   |

|                      |  |  |
|----------------------|--|--|
| RF-A02               | Descarga Software  |  |
| Versión              | 1.0  |  |
| Requisitos asociados |  |  |
| Descripción          | Detecta el terminal desde donde se quiere descargar y su sistema operativo, para enviare el paquete de instalación correcto. |  |

|                      |  |  |
|----------------------|--|--|
| RF-A03               | Petición Direcciones   |  |
| Versión              | 1.0  |  |
| Requisitos asociados | RD-W03 (Radio Taxi ID, Radio Taxi Empresa)<br>RD-W01 (Numero Teléfono)   |  |
| Descripción          | Devuelve una lista con las direcciones frecuentes desde donde el cliente suele pedir taxis. La API hace esta consulta en el servidor central de la empresa de taxis. |  |
| Pre condición        | Se llama previamente a RF-A04 para comprobar si se ha hecho ya esta reserva.   |  |
| Secuencia Normal     | Paso   | Acción   |
|                      | P1   | Al iniciar el software, automáticamente se hace esta llamada al servidor   |
|                      | P2   | La API recoge los datos del cliente y hace la consulta en el servidor central de la empresa de Taxis                           |
|                      | P3   | La API devuelve por JSON una lista con los destinos que la empresa de taxis ha determinado son los mas frecuentes del cliente. |

|                      |  |  |
|----------------------|--|--|
| RF-A04               | Comprobación de petición   |  |
| Versión              | 1.0  |  |
| Requisitos asociados | RD-W03 (Radio Taxi ID, Radio Taxi Empresa)<br>RD-W01 (Numero Teléfono)<br>RD-A01 (ID Calle)<br>RD-A02 (Numero Calle) |  |
| Descripción          | Realiza una consulta en el servidor de la empresa de taxis, comprobando si se ha realizado ya la reserva del taxi.   |  |

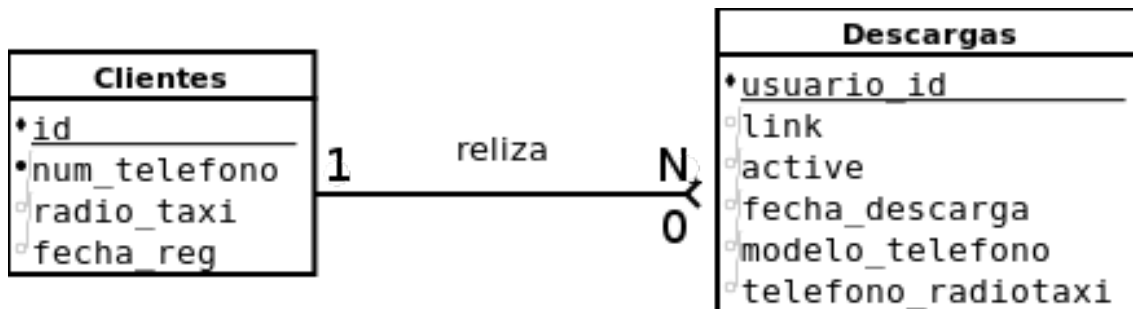


|                      |  |   |
|----------------------|--|---|
| RF-A05               | Crear Reserva  |   |
| Versión              | 1.0  |   |
| Requisitos asociados | RD-W03 (Radio Taxi ID, Radio Taxi Empresa)<br>RD-W01 (Numero Teléfono)<br>RD-A01 (ID Calle)<br>RD-A02 (Numero Calle) |   |
| Descripción          | Realiza la reserva del taxi en el servidor de la empresa de taxis y devuelve el resultado.                           |   |
| Pre condición        | Se ha llamado previamente a RF-A03 para obtener la lista de destinos.  |   |
| Secuencia Normal     | Paso   | Acción  |
|                      | P1   | Se selecciona el destino de la lista  |
|                      | P2   | Aparece mas información de la reserva en pantalla, pudiendo aceptar o rechazar la reserva |
|                      | P3   | Al aceptar se manda la petición a la API.   |
| Postcondición        | Se notifica al usuario del estado de la reserva, con un id de reserva o con un código de error dependiendo del caso. |   |

|                      |  |
|----------------------|--|
| RF-A06               | Info Procesador                                |
| Versión              | 1.0  |
| Requisitos asociados |  |
| Descripción          | Devuelve información sobre la empresa de taxis |

### 3.9.-Modelado de datos

En el modelado de datos nos encontramos con la descripción de todos los datos que se mueven por el subsistema, y las relaciones que tienen entre ellos. A continuación se verá el diagrama de modelado de datos entidad-relación.



#### 3.9.1.-Descripción de Entidades

| Entidad:     | Clientes |                                |         |
|--------------|----------|--------------------------------|---------|
| Atributos    |          |                                |         |
| Nombre       | Tipo     | Descripción                    | Dominio |
| Id           | Entero   | Código del cliente             | Único   |
| num_telefono | Entero   | Número de teléfono del cliente |         |
| fecha_ref    | Fecha    | Fecha de registro del cliente  |         |
| radio_taxi   | Cadena   | 10                             |         |

| Entidad:           | Descargas    |   |         |
|--------------------|--------------|---|---------|
| Atributos          |              |   |         |
| Nombre             | Tipo         | Descripción   | Dominio |
| usuario_id         | Entero       | Código del usuario                                      | Único   |
| lilink             | Cadena (32)  | Link de descarga personalizado                          |         |
| Active             | Cadena (1)   | Estado del cliente                                      | 0,1,2   |
| fecha_descarga     | Fecha        | Fecha en la que se produce la descarga                  |         |
| modelo_telefono    | Cadena (120) | Modelo del telefono desde donde se solicita la descarga |         |
| telefono_radiotaxi | Int          | Teléfono del radiotaxi del cliente                      |         |

### 3.9.2.-Descripción Relacionales

La relación entre las entidades presentadas Clientes-Descargas es la siguiente:  
Un cliente puede realizar 0 o N descargas

### 3.9.3.-Reglas de negocio

Cualquier usuario se podrá registrar en el sistema y se descargará el software, pero la aplicación no se activará si ese usuario no está previamente registrado como cliente de la compañía de taxis.

# **4.-DISEÑO**

## **4.1.-Introducción**

El diseño del sistema informático es una fase fundamental del desarrollo del sistema, el objetivo del cuál es conectar la visión lógica desarrollada en el análisis con la actividad propia de la programación y otras tareas necesarias para poner el sistema en funcionamiento.

De esta forma, el análisis del sistema se ha centrado en qué ha de realizar el sistema, mientras que el diseño físico se preocupa por la manera en que se han de satisfacer estos procesos. Durante la fase de diseño se identifica, analiza y determina como los datos que se han definido en la fase de análisis han de procesarse, almacenarse, y como y en que formatos ha de obtenerse la información de salida.

Según los elementos que constituyen un sistema informático, el diseño se centra en estudiar el análisis y desarrollar modelos que representen el aspecto físico del sistema para cada uno de estos elementos: las interfaces de usuario, los procesos que ha de soportar el software y las bases de datos.

Así, el objetivo es conseguir un diseño del sistema que sea efectivo, fiable y mantenible, es decir, que cumpla los requisitos analizados, las especificaciones y las restricciones de manera correcta.

El diseño conceptual se ha incluido en la fase anterior de análisis del sistema, mediante la elaboración del modelo conceptual de datos (MCD). En esta fase, se ha elaborado, a partir de este MCD, en primer lugar el diseño lógico de la base de datos y posteriormente el diseño físico de ésta.

En esta sección se añade la descripción de la "Stored Procedure" de la base de datos a la cual accede la API para realizar la reserva real del taxi. Se ha tenido que hacer una integración de ambos sistemas ya que la base de datos local está diseñada en MySQL y la remota situada en el servidor de la empresa de taxis en Oracle, pero esto lo analizaremos mas adelante.

Una Stored Procedure es una función programada que realiza su propia gestión en la base de datos instalada, haciendo Inserts, Deletes y Updates donde haga falta. Esta gestión es totalmente transparente para nosotros. El cliente nos ha dicho como llamarla y nos hemos la hemos integrado en la API.

## **4.2.-Diseño de la base de datos**

### **4.2.1.-Diseño lógico**

El diseño lógico es el proceso de construir un esquema de la información, basándose en un modelo de base de datos específico, independientemente del sistema gestor de base de datos concreto que se vaya a utilizar y de cualquier otra consideración física.

Además de todo esto, el esquema lógico es una fuente de información para el posterior diseño físico. También juega un papel importante durante la etapa de mantenimiento del sistema, ya que permite que los futuros cambios que se realicen sobre los programas de aplicación o sobre los datos, se representen correctamente en la base de datos.

Así, se transforma el esquema conceptual desarrollado en la fase anterior, en un esquema lógico que utiliza las estructuras de datos del modelo de base de datos que se utilizará. En nuestro caso el modelo que sigue la base de datos que se va a diseñar es el modelo relacional.

#### **4.2.1.1.-Tablas del diseño lógico**

**Cientes** (id, num\_telefono, radio\_taxi, fecha\_reg)

**Descargas** (usuario\_id, link, active, fecha\_descarga, modelo\_telefono, telefono\_radiotaxi)

#### **4.2.1.2.- Stored Procedure**

**ReservaTaxi** (id\_empresa, f\_recogida, calle\_rec, numero\_rec, piso\_rec, calle\_des, numero\_des, piso\_des, id\_serv)

#### **4.2.1.2.- Descripción de atributos**

Tabla Clientes

**ID** es el identificador del cliente, único, entero y autoincremental.

**Num\_telefono** lo usa la base de datos de la empresa de taxis como identificador propio. Se usará para interactuar con ella.

**Radio\_taxi** es una cadena que indica la compañía de taxis que usa el usuario. Este dato es rellenado por el cliente a la hora del alta, o directamente la central de taxis

envía una página de registro al cliente, la cual introducirá este dato automáticamente

**Fecha\_reg** es un dato tipo fecha, y registra el momento del alta del cliente.

### Tabla Descargas

**Usuario\_id** es el mismo que el de la tabla clientes.

**Link** es una cadena generada por el sistema, que se enviará por sms al cliente para que se descargue el programa.

**Active** es una variable que el software del teléfono usará para saber en que estado está el usuario. Dependiendo de su valor, sabrá si se lo ha descargado, si el usuario existe en la base de datos de la empresa, o por el contrario si ha habido algún tipo de error.

**Fecha\_descarga** es una variable tipo date, que nos muestra, como su nombre indica, la fecha de la descarga.

**Modelo\_telefono.** Cadena que indica el modelo del telefono que va a descargar la aplicación, lo detecta la página php de la api a la que se accede para realizar la descarga y lo guarda en esta variable.

**Teléfono\_radiotaxi.** Esta variable guarda el telefono de la empresa asignada al cliente.

### Stored Procedure ReservaTaxi

Los atributos de la llamada que utilizamos son los siguientes:

Id\_empresa, id\_cliente (el numero de telefono), calle\_rec y numero\_rec.

El resto de atributos son opcionales y para la reserva desde nuestro software para terminales móviles, no los usamos.

El atributo **id\_serv** se recoge y nos indicará el localizador del servicio o en su caso, el tipo de error que ha sufrido la stored procedure.

## 4.2.2.-Diseño físico

El diseño físico es el proceso de producir la descripción de la implementación de la base de datos en memoria secundaria: estructuras de almacenamiento y métodos de acceso que garanticen un acceso eficiente a los datos. Así, en el diseño físico se especifica cómo se guarda la información.

En general, el propósito del diseño físico es describir cómo se va a implementar físicamente el esquema lógico obtenido anteriormente.

Para llevar a cabo esta etapa, se tiene que elegir un sistema gestor de bases de datos. Para la realización de este proyecto se ha escogido MySQL.

En primer lugar se presentan las sentencias para la creación de las tablas de la base de datos.

### Sentencias de creación de las tablas

```
--  
-- Estructura de tabla para la tabla `clientes`  
--  
CREATE TABLE IF NOT EXISTS `clientes` (  
  `id` int(11) NOT NULL auto_increment,  
  `phone_number` int(11) NOT NULL,  
  `radio_taxi` text character set latin1 collate latin1_spanish_ci,  
  `registration_date` datetime NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=87 ;  
--  
-- Estructura de tabla para la tabla `descargas`  
--  
CREATE TABLE IF NOT EXISTS `descargas` (  
  `user_id` int(11) NOT NULL,  
  `link` varchar(32) NOT NULL,  
  `active` varchar(1) NOT NULL,  
  `download_date` datetime default NULL,  
  `phone_model` varchar(120) default NULL,  
  `phone_number_rt` int(10) NOT NULL,  
  PRIMARY KEY (`user_id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```



## Código de lanzamiento de la Stored Procedure

```
$stid = oci_parse($conn, "begin
RESERVA_VICTOR(:ID_EMPRESA,:ID_CLIENTE,:F_RECOGIDA,:CALLE_REC,:NUME
RO_REC,:PISO_REC,:CALLE_DES,:NUMERO_DES,:PISO_DES,:ID_SERV,:ERRCODE)
; end;");

oci_bind_by_name($stid, ":ID_EMPRESA",
$radioTaxiCompany);

oci_bind_by_name($stid, ":ID_CLIENTE", $phone);
oci_bind_by_name($stid, ":F_RECOGIDA", $today);
oci_bind_by_name($stid, ":CALLE_REC", $streetId);
oci_bind_by_name($stid, ":NUMERO_REC", $streetNumber);
oci_bind_by_name($stid, ":PISO_REC", $null);
oci_bind_by_name($stid, ":CALLE_DES", $null);
oci_bind_by_name($stid, ":NUMERO_DES", $null);
oci_bind_by_name($stid, ":PISO_DES", $null);
oci_bind_by_name($stid, ":ID_SERV", $id_serv,8);
oci_bind_by_name($stid, ":ERRCODE", $err_code,8);
oci_execute($stid);
```

### 4.2.2.1.-Integración de la Stored Procedure

Para poder llamar a la *Stored Procedure* ha hecho falta hacer una integración de varios sistemas, ya que en nuestro servidor disponemos de una base de datos MySQL, pero hay que acceder a la base de datos basada en Oracle, del servidor de la empresa cliente.

Para ello se ha instalado el OCI 10.2 (Oracle Client Interface) y la extensión de php oci8.

Para ello, ejecutamos el siguiente script en el servidor:

```
# The script is tested on a virtual machine running OS Linux Ubuntu Server 8.04.1 i386
created on a Mac OS X running Leopard 10.5.3.

# It will install and configure OracleClient v10.2 and the OCI8 v1.3.3.
# 1. You need to be logged as root
# 2. You need to be connected to internet
# 3. You need to download the Oracle Client from the Oracle website
# It downloads and installs of the basic components
apt-get update
apt-get --force-yes -y upgrade
apt-get install --force-yes -y ssh unzip php5-dev php-pear build-essential libaio1
# It create the directory that will contain the 'Instant Client Package - Basic Lite' and the
```

```

'Instant Client Package - SDK'
mkdir -p /opt/oracle
cd /opt/oracle
cp /root/*.zip /opt/oracle
unzip \*.zip
cd /opt/oracle/instantclient_10_2
# It creates 2 links to the libraries libclntsh and libocci
ln -s libclntsh.so.10.1 libclntsh.so
ln -s libocci.so.10.1 libocci.so
# It downloads and compiles the OCI8 module
# can be also use the command
# pecl install oci8
# but the it will be required to specify manually the ORACLE_HOME
mkdir -p /usr/local/src
cd /usr/local/src
pecl download oci8
tar xzf oci8*
cd oci*
phpize
./configure --with-oci8=shared,instantclient,/opt/oracle/instantclient_10_2
make
make install
# It adds the oci8.so extension to the php.ini
echo "extension=oci8.so" >> /etc/php5/cli/php.ini
echo "extension=oci8.so" >> /etc/php5/apache2/php.ini
# It sets the environment variables
cd /etc/init.d
cat apache2 | sed 's/ENV="env -i LANG=C
PATH=VusrVlocalVbin:VusrVbin:Vbin"/ENV="\env -i LANG=C
PATH=VusrVlocalVbin:VusrVbin:Vbin
ORACLE_HOME=VoptVoracleVinstantclient_10_2 TNS_ADMIN=VoptVoracleVtns
LD_LIBRARY_PATH=VoptVoracleVinstantclient_10_2\'/" > apache2.tmp
cp apache2 apache2.backup
rm apache2
mv apache2.tmp apache2
chmod 755 apache2
cd
echo "LD_LIBRARY_PATH=\"/opt/oracle/instantclient_10_2\"" >> .profile
echo "export LD_LIBRARY_PATH" >> .profile
# It creates some test pages
echo "<? phpinfo(); ?>" >> /var/www/info.php
# It restarts apache
/etc/init.d/apache2 restart
echo ""
echo "You can now try to connect the page http://your_ip_address/info.php"
echo "If everything went fine you should see the module oci8 in the list"

```

Como se ve, va a instalar un montón de paquetes, necesarios para realizar la conexión desde la API.

Php5-dev aporta paquetes necesarios de PHP5 para compilar módulos adicionales.

Php-pear es un entorno de desarrollo y sistema de distribución para componentes de código PHP. Consiste en una gran lista de bibliotecas de código PHP que permiten hacer ciertas tareas de manera más rápida y eficiente, reutilizando código escrito previamente por otras personas.

Build-essential contiene una lista informativa de los paquetes considerados esenciales para la creación de paquetes Debian.

Libaio1 permite al entorno de usuario el uso de llamadas del sistema I/O asíncronas del kernel de Linux, necesario para el acceso remoto a la base de datos del cliente.

### ***4.3.-Diseño de las interfaces de usuario***

Las interfaces de usuario son el mecanismo a través del cuál se establece un diálogo entre el sistema informático y las personas. Por ello, se ha de intentar obtener interfaces de usuario apropiadas y desarrollar un entorno amigable.

Se va a diferenciar entre dos interfaces de usuario, la WEB y la de aplicación móvil.

#### **4.3.1.- Interfaces de usuario WEB**

En este apartado se presentará una captura de las pantallas que aparecen en el portal de registro del sistema. Puesto que en la web de aprovisionamiento solo se realiza el registro del usuario y la descarga, solo consta de dos pantallas. Un breve formulario donde se introduce el numero de teléfono y la siguiente pantalla mostrando el link de descarga y el estado del registro.

# RESERVATAXI

Porfavor, rellene el siguiente formulario para darse de alta en nuestro sistema de reservas on-line.

\* Número de teléfono

Enviar

Restablecer

El numero de teléfono será guardado en la base de datos local y con el se podrán obtener el resto de datos, desde el servidor de la empresa.

Para agilizar el proceso, la web dispone de una función javascript que comprobará que se ha introducido un numero de teléfono, es decir, una cadena de enteros, con la longitud mínima de 9 y máxima de 13.

# RESERVATAXI

**En breves instantes le llegará un SMS con un link.**

Abralo y se conectará a internet para acceder a la página de descarga del acceso directo

**Clique el enlace desde su terminal** (será igual a este):

Seleccione el botón **Descargar** y acepte la descarga.

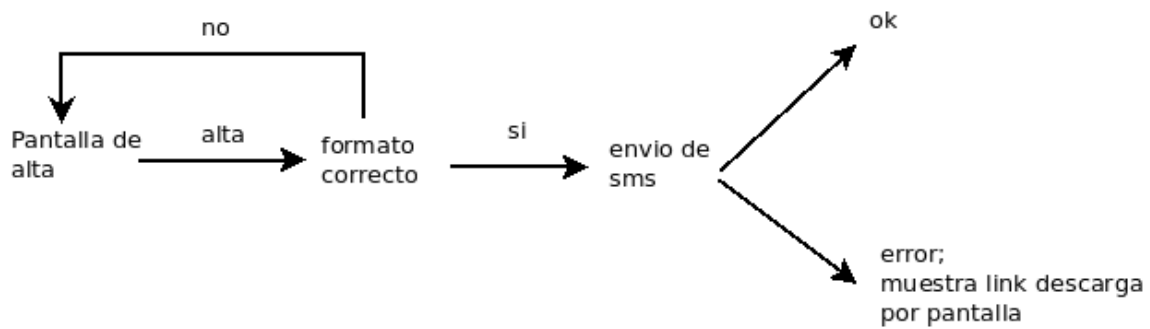
Siga las indicaciones y seleccione **Permitir** o **Aceptar** en los mensajes que le vayan apareciendo

Al finalizar la descarga, su movil le indicará donde ha guardado el icono de acceso directo (según el modelo, podrá cambiar de ubicación). Por lo general, se instalará en **Aplicaciones**, y es fácilmente reconocible porque aparecerá un icono con una T en verde.

El envío del mensaje es gratuito, mientras que la conexión y la navegación serán facturadas por su operador móvil, según la tarifa que tenga contratada.

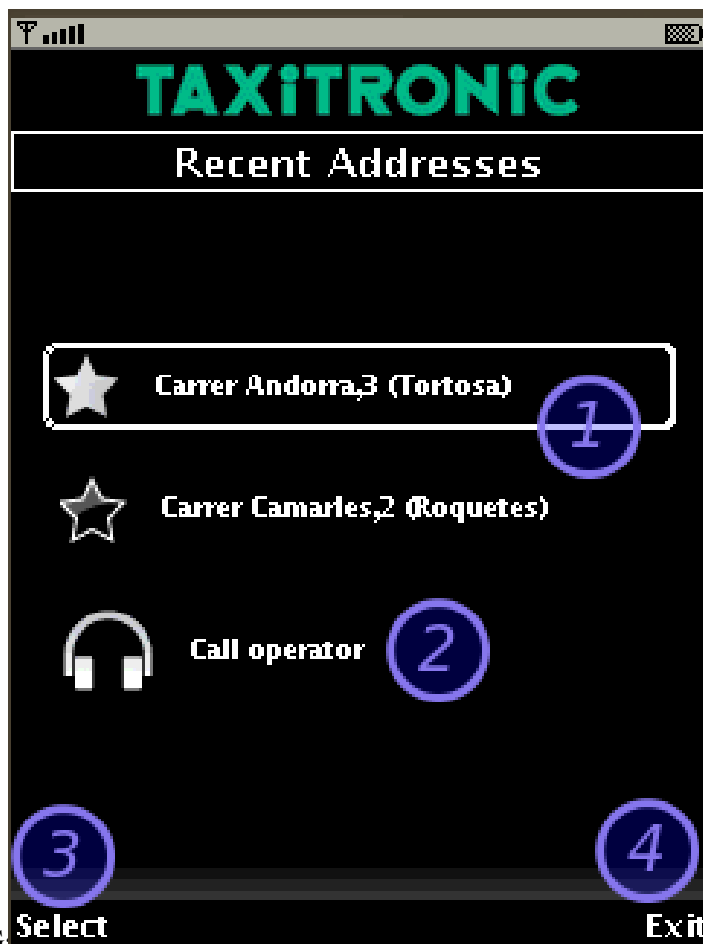
Si todo funciona correctamente, aparecerá el la información de la descarga, con instrucciones de como instalar el software.

Aquí podemos ver el esquema de fucnioamiento:



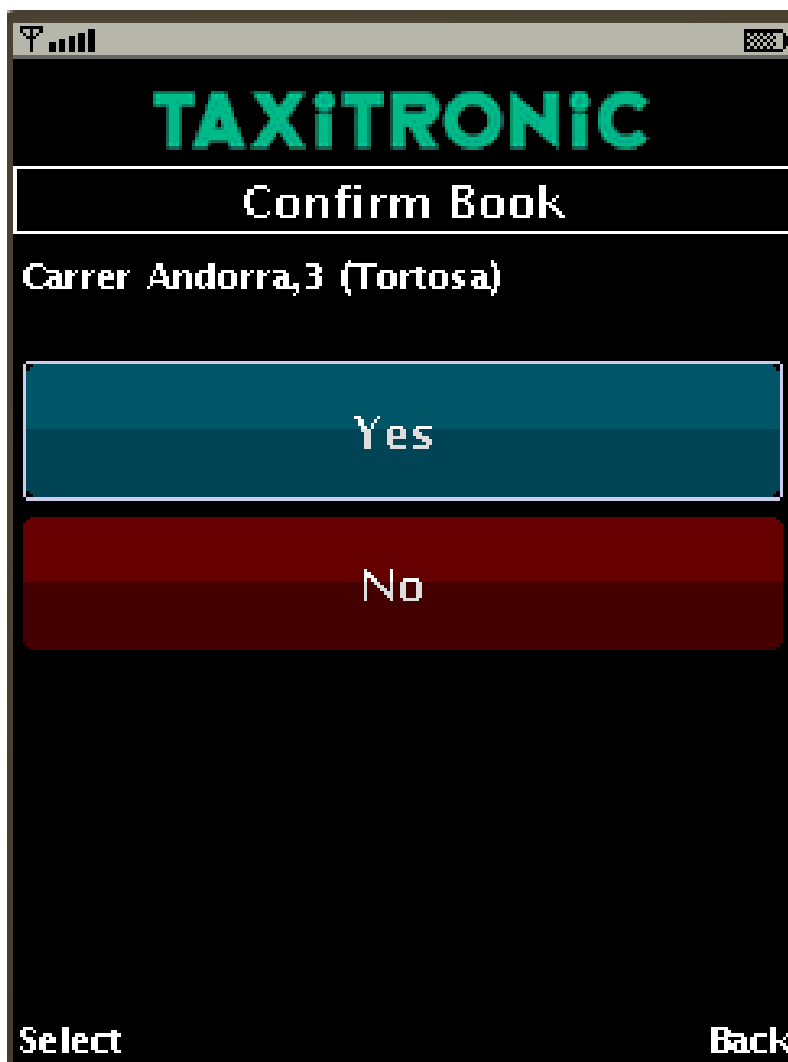
### 4.3.2.- Interfaces de usuario del sistema móvil

#### Pantalla Inicial



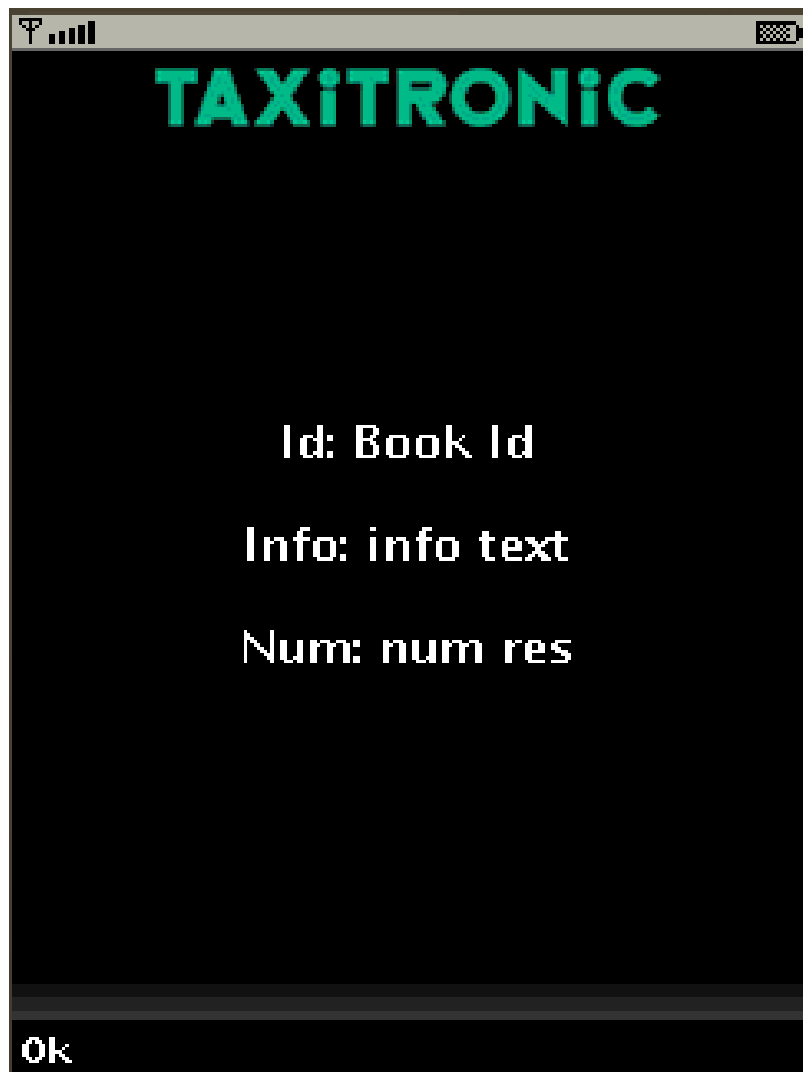
- 1.- Lista con los últimos 5 lugares de recogida
- 2.- Call operator Taxi hará que el terminal llame a nuestro radiotaxi.
- 3.- Modo alternativo de seleccionar la opción (normalmente se usa el botón central del pad, o un toque sobre la opción en pantallas táctiles)
- 4.- Salida del programa

#### Pantalla Confirmación



Nos da la opción de confirmar o rechazar la reserva. En caso de rechazar la reserva, no llevará a la pantalla de selección, o pantalla inicial.

## Pantalla de información de reserva



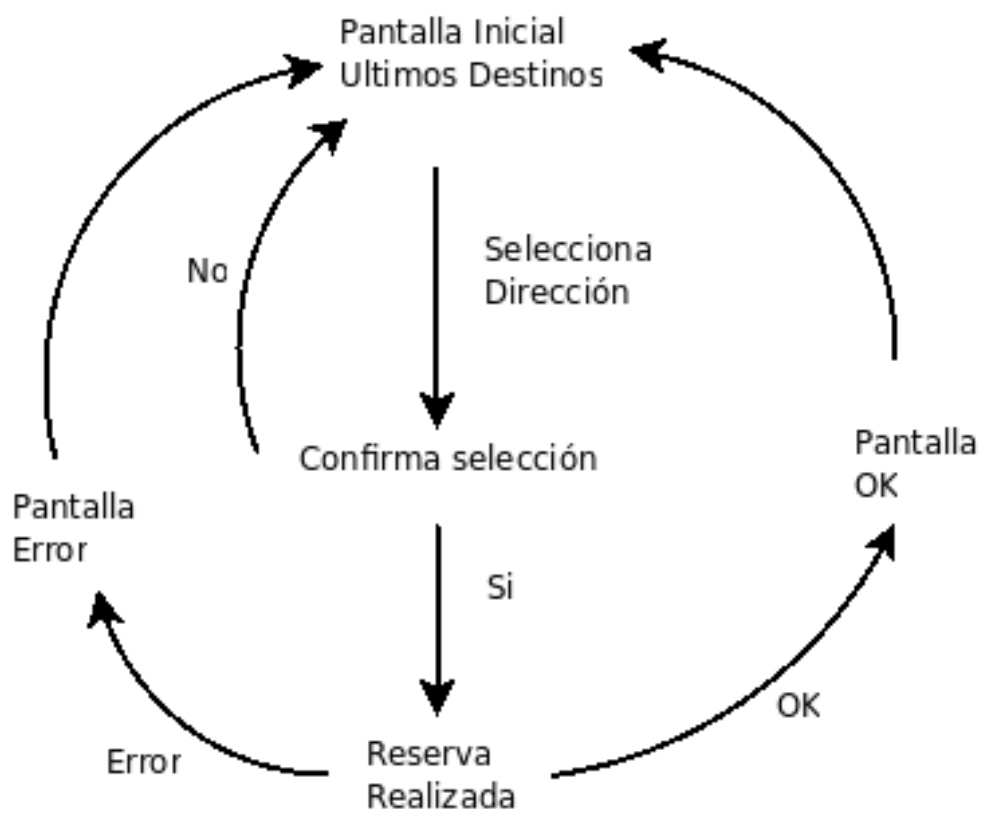
En esta pantalla nos aparece el resultado de la reserva, mostrando tres datos.

**Book Id.** Es el numero de la reserva, asignado por el servidor de la empresa de taxis.

**Info** muestra la calle donde se realizará la recogida.

**Num,** el numero de la calle.

El funcionamiento mostrado en un gráfico es el siguiente:





#### **4.4.-Diseño de procesos**

A continuación se presentaran los procesos que se usarán en el sistema, tanto si hacen uso de las interfaces como si no para llevar a cabo todas las tareas necesarias para el cumplimiento de los requisitos del proyecto.

Los procesos se dividirán en tres bloques, dependiendo de quién los ejecute o a quien les afecte más de forma directa. De este modo tenemos los procesos del alta de usuarios, de la API y de la aplicación del móvil

#### 4.4.1.-Diseño de procesos WEB

|  |  |
|--|--|
| comprueba_format o_datos_alta                | Permite comprobar que los datos introducidos son completamente válidos   |
| Entrada de datos:                            | Numero de telefono   |
| Salida de datos:                             | Ninguna  |
| Interfaces utilizadas:                       | “Alta usuario” cuando pulsa “Enviar”   |
| Tablas de la base de datos a las que accede: | Ninguna  |
| Funcionamiento:                              | <p>➔ Comprobar que están rellanados todos los campos y que tiene un formato correcto, que no contiene letras y el numero de dígitos sea entre 9 y 16</p> <p>➔ Si todo es correcto</p> <p>➔ se pasa a procesar los datos y enviar el sms</p> <p>➔ Si no, sale un mensaje de error javascript pidiendo que</p> |
| alta_usuario                                 | Crea el usuario en la base de datos, genera su link de descarga y enviá el sms al terminal con el link de descarga   |
| Entrada de datos:                            | Numero de telefono, fecha del alta   |
| Salida de datos:                             | Ninguna  |
| Tablas de la base de datos a las que accede: | Clientes y Descargas   |
| Funcionamiento:                              | <p>➔ Se crea el link</p> <p>➔ Se añade el usuario a la base de datos</p> <p>➔ Se enviá el sms</p> <p>➔ Si todo es correcto</p> <p>➔ se muestran los detalles de instalación del software</p> <p>➔ Si no, sale un mensaje de error explicativo</p>  |

#### 4.4.1.-Diseño de procesos API y Software Móvil

Puesto que el terminal se comunicará con la API constantemente, los procesos son los mismos, y se analizarán desde el punto de vista de la API.

|  |   |
|--|---|
| gestión_direcciones                          | Resuelve la petición del listado de direcciones   |
| Entrada de datos:                            | Numero de telefono  |
| Salida de datos:                             | Lista de taxis  |
| Tablas de la base de datos a las que accede: | Nombre de calle, ID de calle y Numero en la calle, consultado en la base de datos de la empresa de taxis  |
| Funcionamiento:                              | <ul style="list-style-type: none"> <li>➔ Al arrancar el software del terminal móvil se envía la petición con el numero de telefono como id</li> <li>➔ Se recogen los datos y se hace la consulta en la base de datos de la empresa de taxis</li> <li>➔ Se devuelve al telefono por json la lista</li> </ul> |

|  |  |
|--|--|
| crear_reserva_taxi                           | Genera la reserva del taxi en la direccion deseada   |
| Entrada de datos:                            | Numero de telefono, ID calle, fecha_reserva  |
| Salida de datos:                             | Estado reserva   |
| Tablas de la base de datos a las que accede: | Desconocido (el servidor de la empresa de taxis se encarga de gestionar la reserva)  |
| Funcionamiento:                              | <ul style="list-style-type: none"> <li>➔ Se recogen los datos enviados desde el terminal</li> <li>➔ Se rellena los parametros de la funcion que va a activar la reserva del servidor externo</li> <li>➔ Se envia el resultado al terminal</li> </ul> |

# **5.- DETALLES DE IMPLEMENTACIÓN**

## **5.1.-Tecnologías empleadas**

Para el desarrollo del proyecto se pretende hacer uso de software libre allí donde sea posible su uso. En la adquisición de cada una de las tecnologías se tiene en cuenta, en distintos grados de importancia, tanto su estabilidad/calidad, su costo y su curva de aprendizaje, siendo ésta lo menos acentuada posible.

Como dispositivo hardware para la realización del proyecto se posee de un ordenador portátil IBM Lenovo T60. Como sistema operativo, se usará GNU/Linux Ubuntu versión 9.10, versión estable de dicha distribución.

Como dispositivo para probar el apartado móvil se ha usado un smartfone con symbian, el Nokia 8500 XpressMusic.

El servidor real es un rack con un procesador Intel Xeon quad core y 4 gigas de ram. Se le ha instalado un Ubuntu Server y un modem usb para el envío de sms.

Para la redacción de los distintos documentos necesarios para el análisis y diseño del proyecto se usará OpenOffice 2.0. En el análisis del proyecto se utilizará un editor de diagramas conocido como Dia, en realidad este programa no tiene la versatilidad de un CASE , pero dada la envergadura del proyecto y la falta de Herramientas CASE en el mundo del software libre, el editor de diagramas Dia cumple dentro de lo que cabe los requisitos.

En la fase de implementación se hará uso de Eclipse. Ésta herramienta facilitaran tanto la programación como la depuración del código necesario para llevar a cabo todas las funcionalidades requeridas por el subsistema y presentadas durante el proyecto.

El apartado web de aprovisionamiento se ha realizado con HTML, Javascript y PHP, la API del servidor en PHP y el software del teléfono con J2ME.

El sistema se aloja en un servidor proporcionado por la empresa re taxis, al cual se le instala un Ubuntu 9.10-desktop-i386, con las siguientes particiones:

- 8,3 GB EXT4 montado en /
- 9,2 GB EXT4 montado en /opt
- 56 GB EXT4 montado en /var
- 1 GB de SWAP

En el servidor se ha instalado un Aache, Mysql Server y los módulos necesarios de PHP.

Se ha lanzado también el script de instalación de Oracle OCI que se ha explicado en otros apartados.

Para realizar el envío del sms, con el link de instalación, se ha usado un modem usb con una tarjeta SIM conectado al servidor.

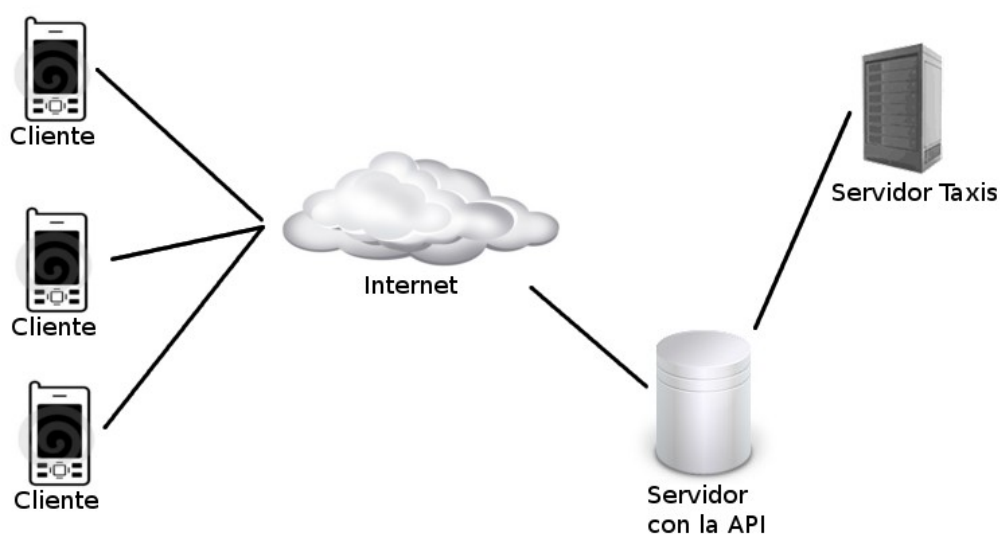
Para poder usar este modem, ha sido necesario instalar el paquete Gnokii, y configurarlo.

Además, se ha programado una copia de seguridad, que se hace en una máquina externa, con sistema operativo Windows. Para ello se ha instalado el Samba, se ha montado la unidad externa en nuestro sistema y se ha programado un script que será llamado por crontab en nuestro sistema, haciendo copias de la página web, el Apache y de la base de datos MySQL.

## 5.2.-Arquitectura final del sistema

Una vez terminado el proyecto se debe obtener un subsistema integrable con los demás subsistemas de la empresa, con un robusto servicio web que permita atender las reservas de clientes, realizar estadísticas, ofrecer un servicio en tiempo real de manera que ni los clientes se queden esperando inútilmente ni los taxistas hagan carreras a direcciones erróneas.

La interfaz de usuario estará realizada mediante las últimas tecnologías en el desarrollo J2ME. El funcionamiento interno del subsistema del servidor estará programado en PHP, el cual accederá a una base de datos local gestionada por el sistema gestor de base de datos y a la base de datos del servidor central de taxis.



Una vez presentada la tabla de las tareas se presentará a continuación un diagrama de Gantt fragmentado, en el que, de forma gráfica, se puede apreciar los periodos de desarrollo asignados en un principio para el proyecto.

### **5.5.-Presupuesto del proyecto**

Una vez conocido el funcionamiento del proyecto, el alcance que pretende tener, su despliegue tecnológico y el desarrollo temporal, se presentará por último el coste total necesario para llevar el proyecto a buen cauce.

Costes relacionados con el desarrollo del proyecto:

| Concepto   | Precio   |
|--|----------|
| Amortización portátil                              | 200,00 € |
| SO Ubuntu Linux más software GNU de administración | 0,00 €   |
| OpenOffice 2.0                                     | 0,00 €   |
| Editor de diagramas Dia                            | 0,00 €   |
| Visor PDF's Acrobat Reader®                        | 0,00 €   |
| MySQL Server                                       | 0,00 €   |
| Eclipse  | 0,00 €   |
| J2ME Polish  | 0,00 €   |

Costes relacionados con los recursos humanos:

| Tarea                           | Precio           |
|---------------------------------|------------------|
| Estudio del proyecto            | 320 €            |
| Análisis de requisitos          | 1520,61 €        |
| Análisis del sistema            | 1280 €           |
| Diseño del sistema              | 1775 €           |
| Construcción y puesta en marcha | 2520 €           |
| Redacción de la memoria final   | 680 €            |
| <b>COSTE TOTAL</b>              | <b>8095,61 €</b> |

Por último se presentará el coste total final del proyecto estimado:

| Concepto                    | Precio           |
|-----------------------------|------------------|
| Costes en el desarrollo     | 200 €            |
| Costes en el funcionamiento | 671,76 €         |
| Costes de mano de obra      | 8095,61 €        |
| <b>COSTE TOTAL PROYECTO</b> | <b>8967,22 €</b> |



## **5.6.-Implementación**

En esta sección del desarrollo del proyecto se conocerán todos los aspectos vinculados a la implementación del proyecto. Teniendo en cuenta tanto el desarrollo de la base de datos como los módulos realizados en los distintos lenguajes de programación necesitados.

### **5.6.1.-Modulo WEB**

Contiene toda la programación necesaria para la gestión del alta de usuarios. Se divide en los siguientes ficheros con la siguiente funcionalidad.

- *index.php*: Contiene el formulario de alta de usuario y una función javascript que se encarga de verificar el buen formato del input.
- *add\_user.php*: inserta en la base de datos al usuario, enviá un sms al usuario usando la librería gnokii,
- *error.php*: contiene y gestiona los errores
- *downloaded.php*: gestiona las descargas, primero detectando que terminal solicita la descarga (usando la librería mdetect) y posteriormente enviando el paquete pertinente, alojado tambien en el mismo directorio.
- *config.inc.php*: contiene la configuración de acceso a la base de datos y varias variables de configuración

### **5.6.2.-Modulo API**

Contiene toda la programación necesaria para la gestión de reserva de taxis desde el terminal móvil y comunica directamente con el servidor central de la empresa de taxis.

- *client\_get\_address\_processor.php*: recoge los datos del cliente local y enviá una petición al servidor de la empresa de taxis, obteniendo una lista de 5 direcciones que será reenviada por json al terminal.
- *radio\_taxi\_hasBook.php*: comprueba si se ha realizado ya una reserva de taxi.
- *radio\_taxi\_create\_dispatch\_processor.php*: realiza la reserva en la dirección seleccionada y enviada desde el terminal, haciendo una

llamada a una función especial programada en el servidor de la empresa de taxis.

### 5.6.3.-Modulo Aplicación móvil J2ME

Puesto que este lenguaje es Java, tendremos una clase principal llamada ReservaTaxi.java, donde se configura el display y desde donde se llamaran al resto de clases, donde cada clase será una pantalla distinta. Lo que se ha hecho es basarse en el modulo original descargable de Polish J2ME y se han añadido la configuración necesaria y las siguientes pantallas:

- *FirstConfigWindow.java*: pantalla que se inicia solo si es la primera vez que se ejecuta el software.
- *LoadingWindow.java*: pantalla de arranque
- *WaitingWindow.java*: pantalla que aparece cuando se calcula que se va a tardar un tiempo en obtener respuesta, por ejemplo en la primera petición de lista de direcciones o en la misma reserva de un taxi.
- *BookWindow.java*: hace la llamada a la API para obtener la lista de direcciones y las muestra por pantalla.
- *BookedWindow.java*: muestra el mensaje de que ya se ha hecho la reserva
- *ConfirmationBookWindow.java*: muestra de nuevo la dirección seleccionada y dos botones, uno para aceptar y otro para cancelar.
- *DispatchInfoWindow.java*: muestra la información de la reserva.

## **6.-PRUEBAS, PUESTA EN MARCHA**

## **6.1.- Pruebas**

Una vez completado la programación de los tres subsistemas, se han montado de manera local y generado respuestas aleatorias a las llamadas para probar cada uno de los casos.

Se han hecho las pruebas pertinentes con el módem GSM, y el envío de sms.

Se han probado también diferentes terminales móviles y ajustado los gráficos para poder presentar las interfaces de la manera mas universal posible.

Durante el la realización del proyecto se han probado muchos dispositivos con diferentes características, tal y como el tamaño ta la pantalla, el tipo de input (teclado o pantalla táctil), y se han ido haciendo modificaciones hasta conseguir la versatilidad requerida.

### **6.1.1.-Creación de usuario, envío de sms e instalación**

En la sencilla página web de inicio, introucimos nuestro número de telefono y le damos a Enviar.

Una función Javascript se encargará de verificar que se ha introducido un numero de telefono correcto, es decir, una cadena de enteros, entre 9 y 13 numeros. Mientras no se cumpla éste requisito, la página no dejara al usuario seguir adelante.

# RESERVATAXI

Porfavor, rellene el siguiente formulario para darse de alta en nuestro sistema de reservas on-line.

\* Número de teléfono

Enviar

Restablecer

Si todas las comprobaciones han sido superadas, el sistema enviará un sms al numero indicado y nos mostrará un mensaje informativo por pantalla, como éste:

# RESERVATAXI

**En breves instantes le llegará un SMS con un link.**

Abralo y se conectará a internet para acceder a la página de descarga del acceso directo

**Click el enlace desde su terminal** (será igual a este):

Seleccione el botón **Descargar** y acepte la descarga.

Siga las indicaciones y seleccione **Permitir** o **Aceptar** en los mensajes que le vayan apareciendo

Al finalizar la descarga, su movil le indicará donde ha guardado el icono de acceso directo (según el modelo, podrá cambiar de ubicación). Por lo general, se instalará en **Aplicaciones**, y es fácilmente reconocible porque aparecerá un icono con una T en verde.

El envío del mensaje es gratuito, mientras que la conexión y la navegación serán facturadas por su operador móvil, según la tarifa que tenga contratada.

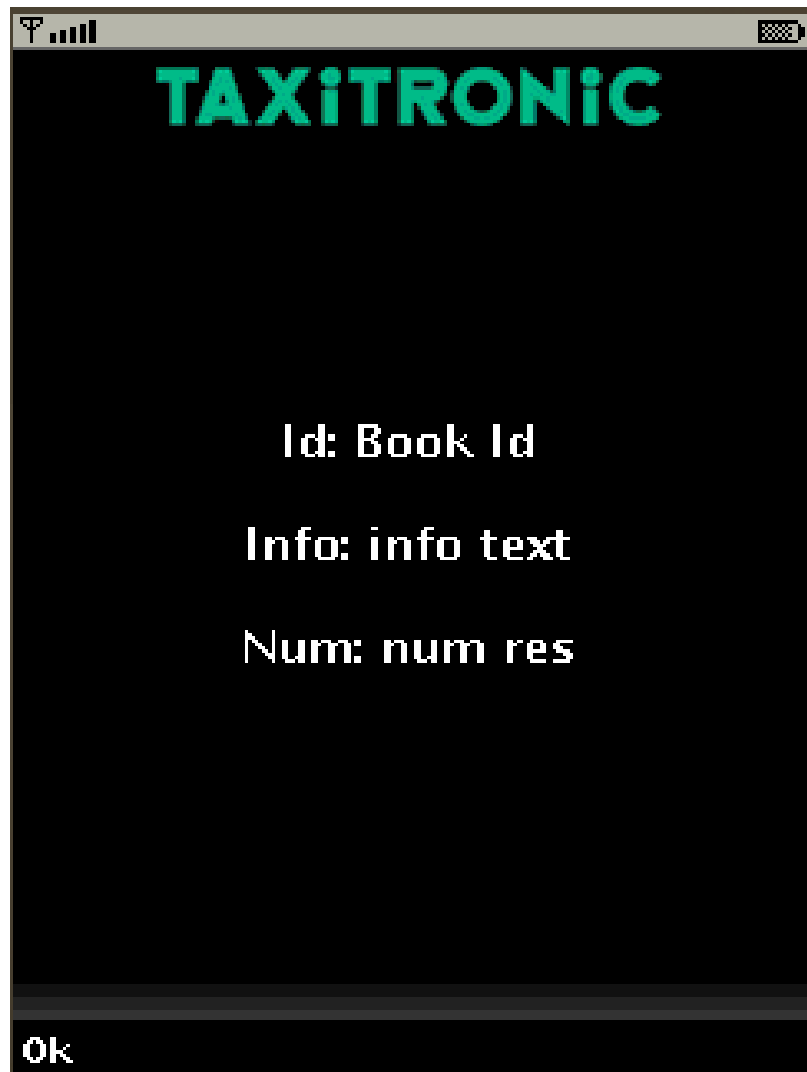
En el SMS enviado, nos aparece un link que apunta a la API, donde ésta detectará el modelo de nuestro terminal y nos enviará el paquete de instalación con el software de RESERVATAXI.

### 6.1.2.-Reserva de taxi

Una vez instalado el software, iniciamos el programa y podemos comprobar en la barra de estado como se va conectando al servidor y el proceso de arranque sucede sin errores. Acto seguido aparece la pantalla de inicio, donde se muestran las direcciones que el servidor nos ha mandado como favoritas:



Seleccionando la opción deseada, procederemos a crear la reserva, obteniendo del servidor una respuesta que será mostrada por nuestro terminal:



Pudiendo comprobar el identificador de la reserva, que usamos en caso de incidencia, llamando al operador de taxis, un campo con información donde aparece el numero de la calle donde se realizará la recogida, y el numero de la calle.

Una vez se han hecho todas las posibles pruebas, corregido errores y ajustado detalles, se ha configurado la API para que apunte al servidor real, y se han borrado las respuestas aleatorias.

Cuando ya ha estado todo montado, se ha procedido a la reserva de un taxi, usando un usuario demo, que posteriormente fue deshabilitado. En el instante de la prueba hubo una operadora de la empresa de taxi pendiente

del proceso, y en caso de hacerse la reserva, ella la cancelaría. Efectivamente, en el primer intento no funcionó, y se tuvieron que corregir errores que se generaban al pasar las cadenas con las direcciones de un protocolo a otro, y en la llamada de la "Stock Procedure" ResevaTaxi.

Una vez pulido cada detalle, se procede a la puesta en marcha.

## **6.2.- Puesta en marcha**

La puesta en marcha del sistema, una vez todo ya funciona a la perfección, trata de llevar el servidor que hemos configurado al centro de calculo de la empresa de Taxis, donde está su servidor con toda la gestión de los taxis, y la base de datos a la que accedemos para reservar un taxi.

Se le ha asignado una dirección IP al servidor, dentro del rango de la empresa del cliente, y hemos apuntado a la dirección local del servidor central, ya que antes apuntábamos a la dirección publica.

Hecho esto, se activa el sistema y se monitoriza con atención constante durante las primeras horas y poco a poco vamos mirando los logs del sistema para comprobar que todo funciona correcto, hasta que al final teniendo la seguridad de que todo funciona correctamente se dejan las comprobaciones por completo, ya que si hay algún error grande seriamos avisados por la compañía de taxis casi instantáneamente.



## **7.-CONCLUSION**

## 7.1.- Conclusión

Como primer contacto con el mundo de la demanda y la oferta, al ser un proyecto real he aprendido a trabajar bajo presión tanto de los jefes como del cliente.

A pesar de las dificultades el proyecto ha salido adelante y con ayuda de los compañeros de trabajo hemos logrado sacarlo adelante, cosa que agradezco infinitamente.

Es un proyecto base, que de hecho ha servido para posteriores proyectos, donde se han implementado interfaces mucho mas complejas y abastecido mas mercado (iPhone ha sido básicamente lo que faltaba).

Actualmente existe una aplicación hecha en colaboración con nuestra empresa, GooTaxi , la cual permite reservar taxis haciendo click en un mapa desde el terminal.

Debido a la presión y al ser un prototipo, muchas interfaces no están muy cuidadas ya que fueron casi de prueba pero el cliente decidió que se conformaba con la funcionalidad. Hubiera sido muy sencillo implantar el estilo de la web donde se ha colocado este servicio, por ejemplo.

Además, a pesar de no haber sido pedido por el cliente, falta un servicio de gestión de usuarios, una interfaz que permitiera modificar, crear y borrar usuarios inválidos o ya no activos.

Sabiendo como funciona el servidor del cliente, se podría haber ofrecido mas información al usuario, como por ejemplo el tiempo estimado hasta la recogida, o la dirección donde se encuentra el taxi que acepta la petición de recogida.

## **8.-BIBLIOGRAFIA**

- [www.php.net](http://www.php.net)
- [www.mysql.com](http://www.mysql.com)
- [www.oracle.com](http://www.oracle.com)
- [www.enough.de/products/j2me-polish/](http://www.enough.de/products/j2me-polish/)
- [www.ubuntu.es](http://www.ubuntu.es)
- [www.gnokii.org](http://www.gnokii.org)
- [www.efectosjavascript.com](http://www.efectosjavascript.com)