

*Títol* : Gestor de CV Online

*Volum* : 1

*Alumne* : Sara Hernanz Hernanz

*Director/Ponent* : Lluís Solano Albajes

*Departament* : Llenguatges i Sistemes Informàtics

*Data* :

## **Dades del Projecte:**

*Títol del projecte: Gestor de CV Online*

*Nom de l'estudiant: Sara Hernanz Hernanz*

*Titulació: Enginyeria Tècnica en Informàtica de Sistemes*

*Crèdits: 15*

*Director/Ponent: Lluís Solano Albajes*

*Departament: LSI*

---

## **MEMBRES DEL TRIBUNAL** (*nom i signatura*)

*President: Lluís Pérez Vidal*

*Vocal: Marc Solé Simó*

*Secretari: Lluís Solano Albajes*

---

## **QUALIFICACIÓ**

*Qualificació numèrica:*

*Qualificació descriptiva:*

*Data:*

---

# **GESTOR DE CV ONLINE**

PFC INGENIERÍA TÉCNICA EN INFORMÁTICA DE  
SISTEMAS

**SARA HERNANZ HERNANZ**  
**DIRECTOR: LLUÍS SOLANO ALBAJES**

**DEPARTAMENT DE LENGUATGES I SISTEMES INFORMÀTICS**



# Índice

<b>1. INTRODUCCIÓN</b>	<b>3</b>
1.1. Definición y objetivos del proyecto	3
<b>2. EL PROCESO DE DESARROLLO DE SOFTWARE Y METODOLOGÍA</b>	<b>5</b>
2.1. El proceso de desarrollo de software	5
2.2. Metodología del Proceso Unificado	5
2.3. Aplicación de la metodología UP en el proyecto	11
<b>3. PLANIFICACIÓN Y COSTES REALES</b>	<b>13</b>
3.1. Calendarios de trabajo	13
3.2. Recursos y costes	15
<b>4. ANÁLISIS DE REQUERIMIENTOS</b>	<b>18</b>
4.1. Estudio previo	18
4.2. Requerimientos funcionales	26
4.3. Requerimientos no funcionales	28
<b>5. ESPECIFICACIÓN</b>	<b>29</b>
5.1. Modelo de Casos de Uso	29
5.1.1. Diagrama de Casos de Uso	30
5.1.2. Especificación de los Casos de Uso	31
5.2. Modelo conceptual	44
5.3. Esquema de la arquitectura	45
5.4. Modelo de comportamiento	47
5.4.1. Diagramas de secuencia	47
5.4.2. Operaciones del sistema	54
5.4.3. Contratos de las operaciones	55
<b>6. DISEÑO</b>	<b>64</b>
6.1. Patrón de Diseño	65
6.1.1. Patrón Modelo-Vista-Controlador	65

6.1.2. Aplicación del Patrón Modelo-Vista-Controlador	70
6.2. Diagramas de secuencia	71
6.3. Diagrama de clases de diseño	74
6.4. Diseño del Modelo de Datos	75
6.5. Diseño externo de la capa de presentación. Diseño de formularios	78
7. IMPLEMENTACIÓN	82
7.1. Tecnología y lenguajes utilizados	82
7.1.1. Apache	82
7.1.2. Lenguajes HTML y CSS , PHP y JavaScript	82
7.1.3. MySQL	85
7.2. Exportación a PDF	86
7.3. Seguridad	86
7.3.1. Autenticación y privilegios	87
7.3.2. CSS y HTTPS	88
8. PRUEBAS	90
9. INSTALACIÓN Y MANTENIMIENTO	92
10. CONCLUSIÓN Y FUTURO DE LA APLICACIÓN	93
11. BIBLIOGRAFIA	94
11.1. Bibliografía digital	95
12. ANEXOS	97
12.1. Programas utilizados	97
12.1.1. Paquete XAMPP	97
12.1.2. Editplus	98
12.1.3. Gant Project	98
12.1.4. PhpMyAdmin	98
12.2. Glosario	99

## 1. INTRODUCCIÓN

### 1.1. Definición y objetivos del proyecto

La docencia y la investigación son las competencias esenciales propias de la Universidad, pero también es sensible a los problemas específicos de la inserción de sus titulados en el mundo social y laboral, para lo cual promueve la participación de sus estudiantes en los *Convenios de Cooperación Educativa* Universidad/Empresa.

Las acciones principales establecidas en los *Convenios de Cooperación Educativa* son:

- potenciar la realización de prácticas en empresas e instituciones
- propiciar los proyectos fin de carrera con tutores de empresa/institución

con el triple objetivo del incremento tecnológico de las empresas mediante el aporte de recursos humanos, el del acercamiento del estudiante a la realidad profesional futura y el de propiciar vías de acceso de la Universidad a las empresas e instituciones.

La gestión de los *Convenios de cooperación educativa* se realiza on-line según el procedimiento siguiente:

- El estudiante tiene que introducir su currículum en la *Bolsa de convenios* on-line acreditándose con un nombre de usuario y una contraseña.
- Una vez colgado el currículum en la Bolsa, el estudiante puede contactar con las empresas que le interese así como recibir directamente ofertas de las empresas.
- Establecido el contacto, la empresa nombrará un tutor y fijará un plan de trabajo. Una vez supervisado este plan de trabajo se firmará el convenio.

Este proyecto aborda el **Módulo de Gestión de Currículum Vitae (CV)** que formaría parte del Sistema de Gestión de Convenios de Cooperación Educativa para la Escuela Técnica Superior de Ingeniería Industrial de Barcelona.

No obstante, este módulo podría gestionar cualquier tipo de CV, no solamente los CV de los estudiantes que quieren participar en dichos convenios.

El proyecto desarrolla una aplicación web que permite la gestión de CV. Los objetivos principales del proyecto son:

- realizar un estudio de los gestores de CV on-line existentes en la actualidad
- definir todas las funcionalidades necesarias para la gestión de CV
- definir un modelo de datos que permita guardar y gestionar toda la información necesaria
- hacer un diseño que se ajuste a los requerimientos funcionales definidos
- implementar las funcionalidades definidas
- realizar las pruebas necesarias para garantizar la corrección de la aplicación
- documentar el trabajo realizado y explicar las decisiones tomadas



## 2. EL PROCESO DE DESARROLLO DE SOFTWARE Y METODOLOGÍA

### 2.1. El proceso de desarrollo de software

La **Ingeniería del Software** se podría definir como el establecimiento y aplicación de principios de la Ingeniería para obtener software. Teniendo en cuenta factores tan importantes como el coste económico, la fiabilidad del sistema y un funcionamiento eficiente que satisfaga las necesidades del usuario.

El **proceso de desarrollo de software** “es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo”. Concretamente “define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo” [Jacobson 1998].

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso se le llama el **ciclo de vida del software** que comprende cuatro grandes fases:

- **Concepción:** define el alcance del proyecto y desarrolla un caso de negocio.
- **Elaboración:** define un plan del proyecto, especifica las características y fundamenta la arquitectura.
- **Construcción:** crea el producto
- **Transición:** transfiere el producto a los usuarios.

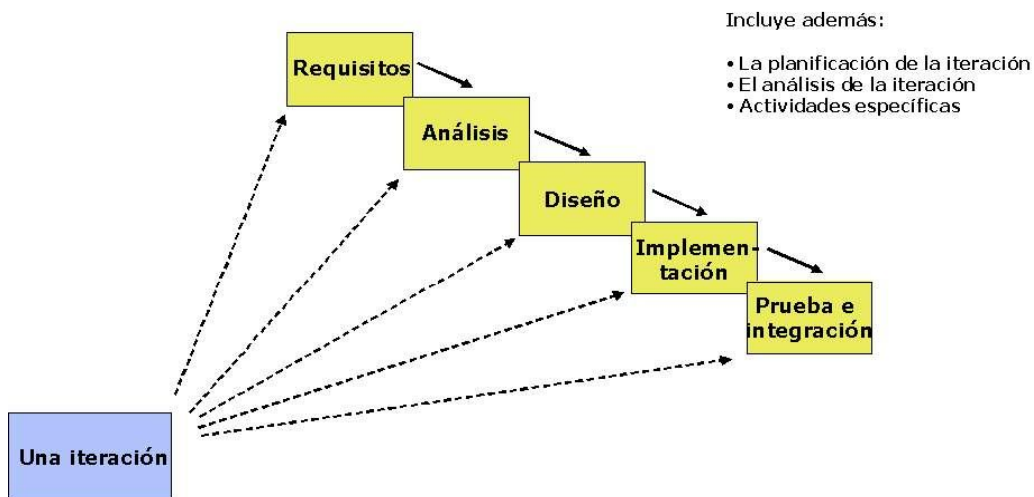
### 2.2. Metodología del Proceso Unificado

El **Proceso Unificado (UP)** es un método iterativo de desarrollo de software que guía a los equipos de proyecto en cómo administrar el desarrollo de un modo controlado mientras se equilibran los requerimientos del negocio, el tiempo al mercado y los riesgos del proyecto.

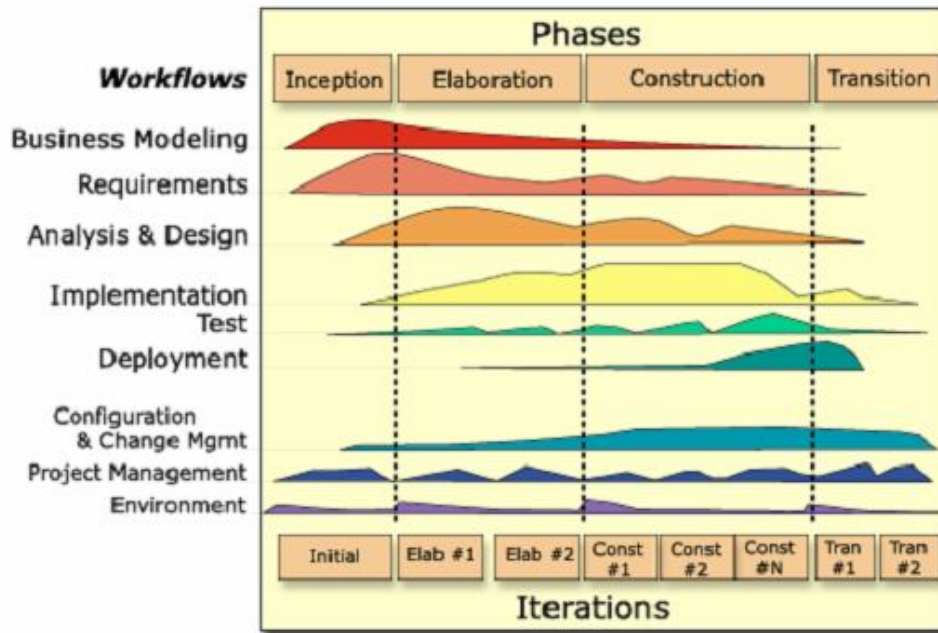
El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura.

Las **características primordiales de la metodología del Proceso Unificado** son:

- **Iterativo e incremental:** el UP está compuesto de cuatro fases denominadas Concepción, Elaboración, Construcción y Transición. Cada una de estas fases se divide a su vez en una serie de iteraciones o ciclos (la de concepción sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo. Cada una de estos ciclos se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada.



En la siguiente figura se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto UP.



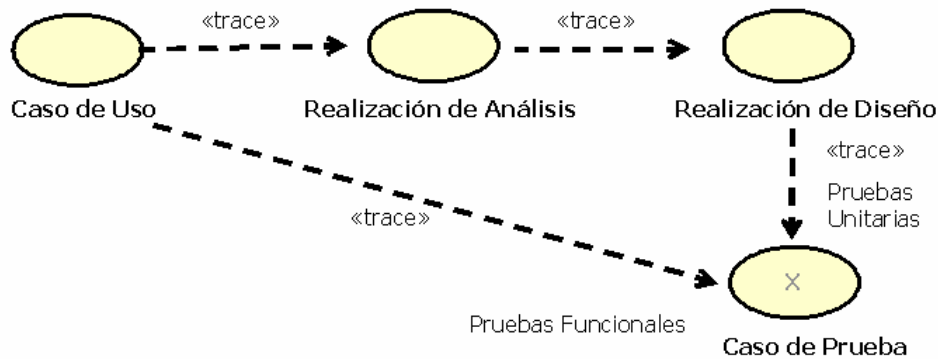
- **Centrado en la arquitectura:** La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema

En el caso de UP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

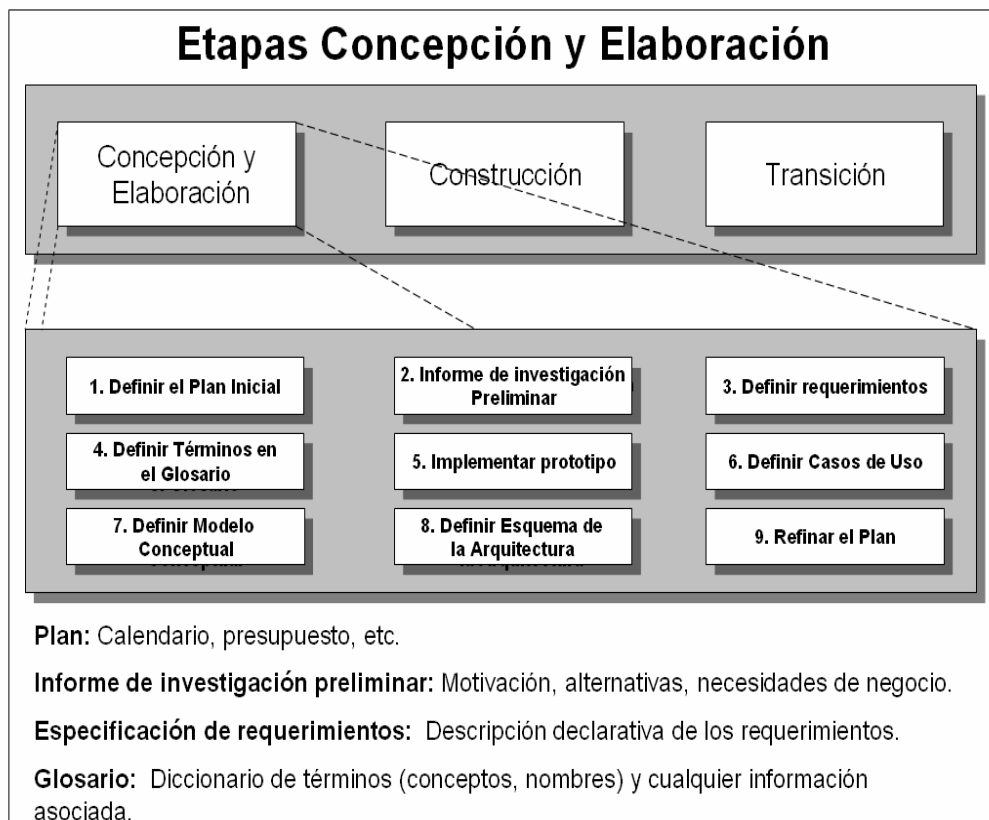
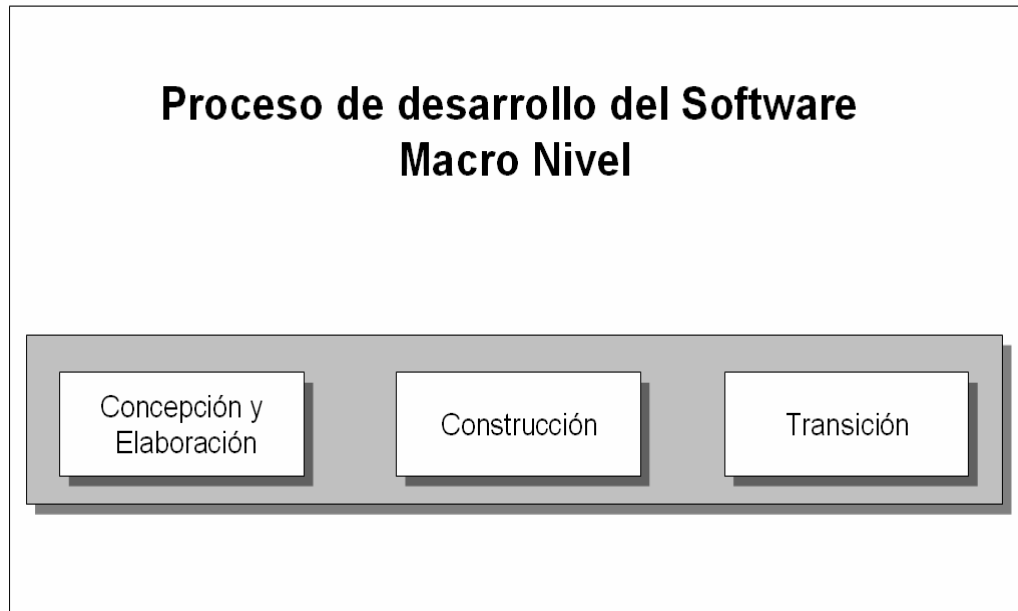
- **Guiado por casos de uso:** en el UP los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que en cada iteración, basándose en los Casos de Uso, se creen los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada Caso de Uso.

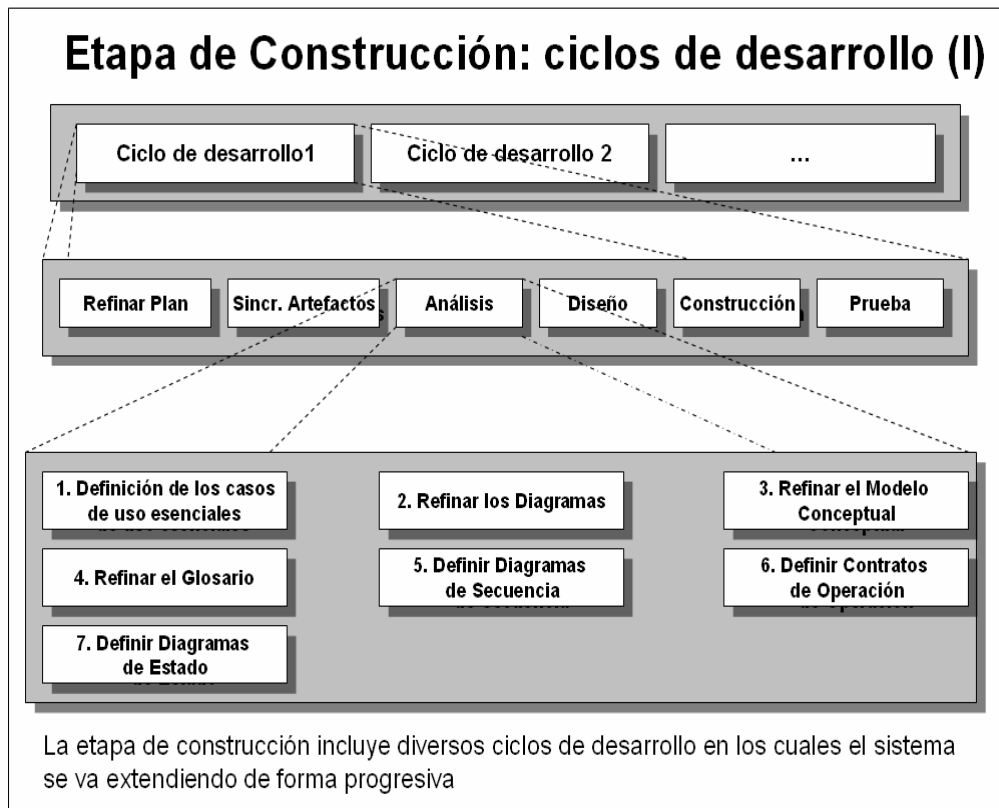
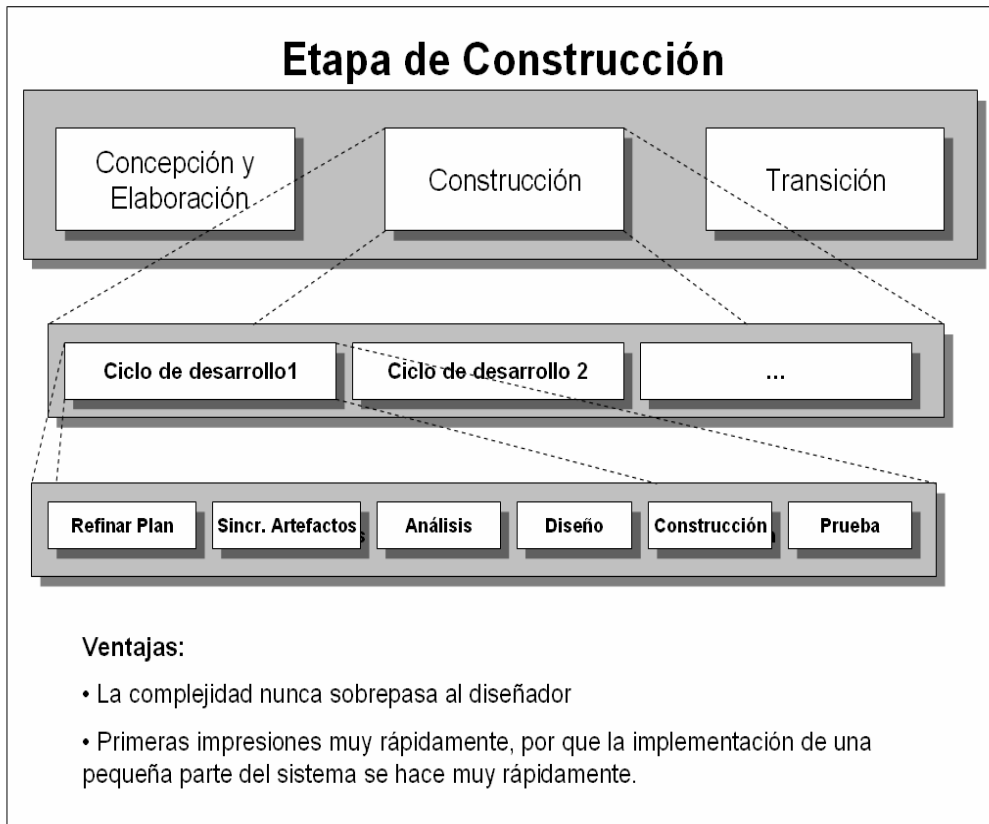
En la siguiente figura se muestra el camino que se sigue en las diferentes actividades del proceso de desarrollo a partir de los Casos de Uso.

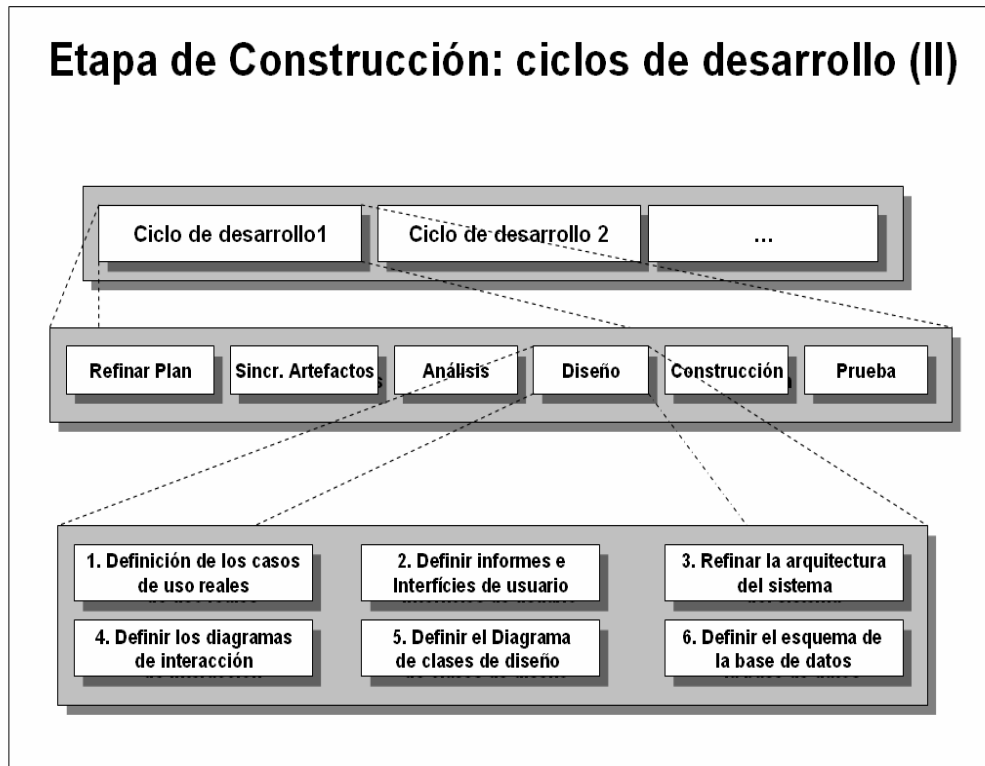


- **Confrontación de riesgos:** el UP requiere que el equipo de proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración, deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.

A continuación se muestra un esquema gráfico de Proceso Unificado de desarrollo de Software.







### 2.3. Aplicación de la metodología UP en el proyecto

A continuación se indica la correspondencia entre las etapas del Proceso Unificado y los apartados de la memoria.

- **Concepción:** Corresponde a los apartados de *Introducción, Planificación y Costes Reales y Análisis de Requerimientos*.
- **Elaboración:** Corresponde a los apartados de *Especificación*
- **Construcción:** Dentro de esta etapa hasta la parte del Análisis sigue correspondiendo al apartado de *Especificación* y el resto de la etapa, a partir de la parte de Diseño, corresponde a los apartados *Diseño, Implementación y Pruebas*
- **Transición:** Corresponde al apartado de *Implantación y mantenimiento*.

Todo el proceso ha sido iterativo e incremental. A partir de cada caso de uso se comienza un nuevo ciclo de desarrollo en el cual se itera sobre el análisis, el diseño, la implementación y las pruebas de manera que en cada iteración se refina el trabajo ya realizado con lo que se obtiene un incremento del producto desarrollado que hace que el sistema mejore, alcanzando así los objetivos planteados inicialmente.

De las cuatro etapas mencionadas en este proyecto solo se han realizado las tres primeras. Sobre la cuarta etapa se explican los pasos a seguir en el apartado correspondiente.

El Proceso Unificado a parte de definir que pasos hay que seguir para el desarrollo de Software, también especifica quién debe realizar estos pasos. En este proyecto todos los roles (jefe de proyecto, analista y programador) han sido desarrollados por mí.



### **3. PLANIFICACIÓN Y COSTES REALES**

Para comenzar con el proceso de desarrollo de un sistema Software es necesario definir una planificación inicial. Esta planificación debe ser realista, estructurada y ajustada a los recursos disponibles. En definitiva, nos ayuda a organizar la metodología de trabajo a seguir para alcanzar los objetivos planteados.

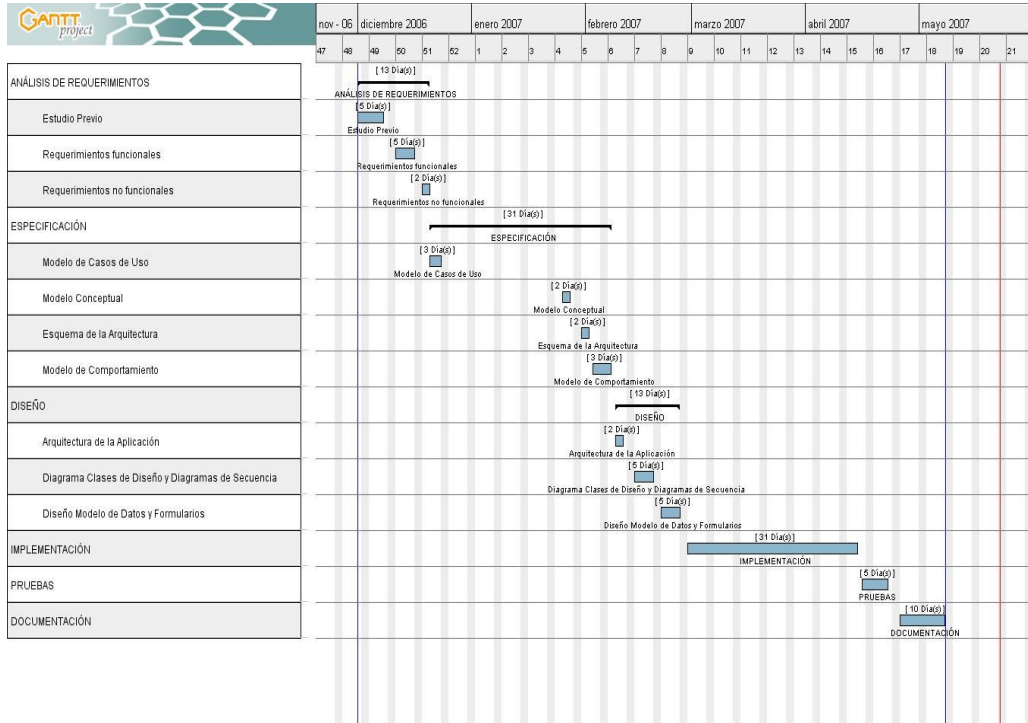
La planificación inicial está compuesta de los calendarios de trabajo, los recursos utilizados y los costes del proyecto. Así mismo, en este capítulo se muestra la desviación que se haya producido y en qué tareas, respecto del calendario de trabajo inicial.

Para representar las fases y actividades de la estructura analítica del proyecto, se ha usado los diagramas de Gantt. El objetivo de dicho diagramas es el de mostrar el tiempo de dedicación previsto para diferentes tareas a lo largo de un tiempo total determinado.

#### **3.1. Calendarios de trabajo**

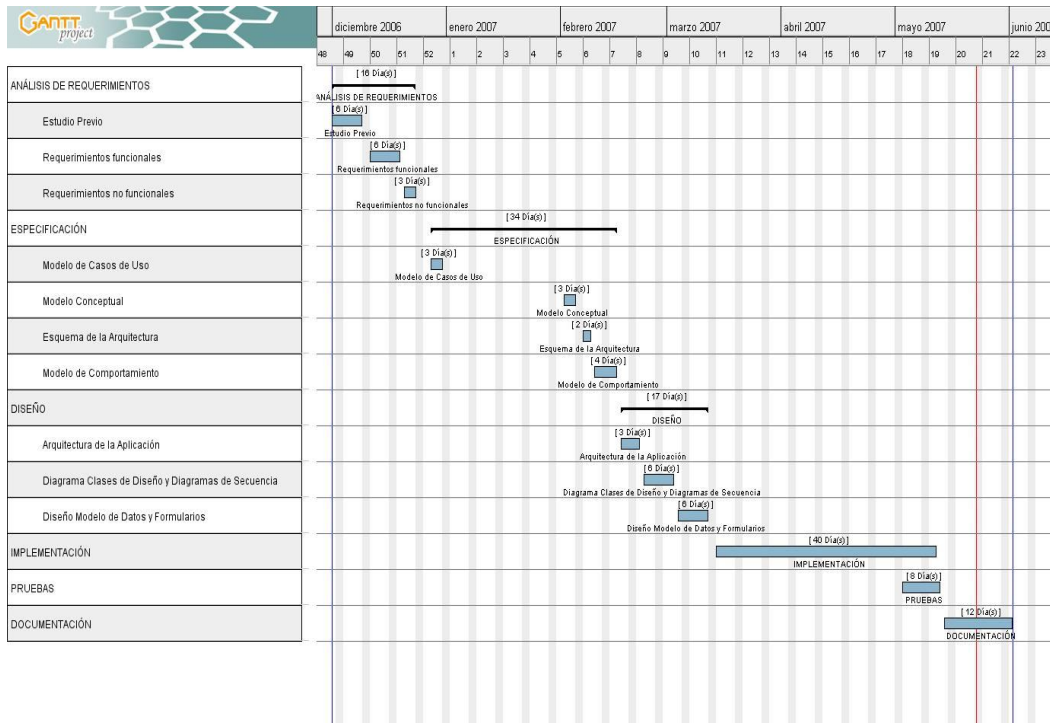
EL trabajo previsto para la realización de un proyecto se estima en una jornada laboral de 4 horas durante 4 meses para las ingenierías técnicas. Por lo que el trabajo requerido es de unas 300-320 horas de dedicación para este proyecto (15 créditos).

Los calendarios de trabajo tienen en cuenta esta estimación y el calendario inicial de trabajo, sin contar los fines de semana ni los días festivos, se muestra en el Diagrama de Gantt siguiente:



**HORAS TOTALES = 80 días \* 4h = 320h.**

El calendario real se muestra a continuación:



**HORAS TOTALES = 94 días \* 4h = 376h.**

Como se puede ver hay diferencias significativas sobre todo en la implementación del sistema debido a las horas dedicadas al aprendizaje de los lenguajes de programación utilizados. También hizo falta solapar tareas como la fase de pruebas. Finalmente el proyecto tuvo una duración de 94 días que equivale a 376 horas de trabajo. Esto representa un 17'5% de desviación entre las horas estimadas y las reales.

### **3.2. Recursos y costes**

En este proyecto tenemos 3 tipos de fuentes de costes:

- Coste del hardware: Coste de las máquinas que intervienen en el sistema
- Coste del software: Coste de los programas y de la plataforma de conexión
- Coste de los recursos humanos: Sueldo de las personas que han trabajado en el proyecto

#### **HARDWARE**

En este apartado solo se tiene en cuenta el entorno de desarrollo del sistema no el de producción (entorno donde se explota el sistema) ya que el servidor de base de datos y el de aplicaciones pertenecen a la empresa/institución donde se implante el sistema.

Para el desarrollo se ha usado un ordenador portátil con las siguientes características:

- Marca – Toshiba
- Modelo – Satellite A110-179
- Procesador – Intel® Core i2 T2250 @ 1.73 Ghz
- Memoria – 1Gb
- Disco Duro – 80Gb
- Monitor – 15 WXGA TFT SCREEN
- Grabador CD – DVD Super Multidriver (2-layer)

Con un coste de 800€

## SOFTWARE

En la siguiente tabla se especifican los productos software usados en este proyecto así como su coste:

Concepto	Producto	Total
Sistema Operativo	Windows XP HOME Edition	Licencia adquirida con el portátil
Entorno de desarrollo	EditPlus	Trial Version
Servidor Apache, MySQL Y PHPMyAdmin	Paquete XAMPP	Gratuito
Gestor de proyectos	Gantt Project 2.0.4	Gratuito
Procesador de textos	Microsoft Word XP	Licencia FIB
Navegador Web	Microsoft Internet Explorer Mozilla	Gratuito
<b>Total</b>		<b>0€</b>

## RECURSOS HUMANOS

A continuación se especifican los perfiles necesarios para el proceso de desarrollo de un sistema Software, sus funciones, horas de trabajo atribuidas y el coste.

- Jefe de proyecto: Se encarga de coordinar el proyecto, de gestionar el tiempo y los recursos. Se le atribuyen un 5% de las horas dedicadas al proyecto completo. Estableceremos un coste de 48€hora.
- Analista: Se encarga de realizar el análisis de requerimientos, la elaboración del modelo de casos de uso, del modelo conceptual y del modelo de comportamiento. Además se le atribuye un 35% de las horas dedicadas a la documentación. Estableceremos un coste de 36€hora.
- Diseñador: Se encarga de realizar el diseño de diagrama de clases y diagramas de secuencia, diseño del modelo de datos y de la interfaz gráfica. Además se le atribuye un 35% de las horas dedicadas a la documentación. Estableceremos un coste de 35€hora.
- Programador: Se encarga de la codificación de la aplicación y de las pruebas. Además se le atribuye un 30% de las horas dedicadas a la documentación. Estableceremos un coste de 24€hora.

Cargo	Horas de trabajo	Coste
Jefe de proyecto	18,8 h	902,4 €
Analista	118,56h	4268,16 €
Diseñador	72,96	2553,6 €
Programador	165,68	3976,32 €
TOTAL		11700,48€

**Tabla resumen perfiles-horas-coste**

## PRESUPUESTO

Una vez vistos los costes detalladamente, mostraremos en la siguiente tabla el desglose y el coste total del proyecto.

Concepto	Coste
HARDWARE	800€
SOFTWARE	0€
RECURSOS HUMANOS	11700,48€
TOTAL	12500,48€

## 4. ANÁLISIS DE REQUERIMIENTOS

El **análisis de requerimientos** es la primera etapa de un proyecto software, en ella se tratan de definir las condiciones o capacidades necesarias para uno o varios usuarios con el fin de solucionar un problema o conseguir un objetivo.

Para la creación global del sistema se necesita comprender todos los objetivos y necesidades del usuario. En primer lugar, hemos de especificar el comportamiento externo del sistema desde el punto de vista del usuario, en forma de requisitos. La determinación de los requerimientos se haya en base a la experiencia, de hablar con los usuarios finales sobre sus necesidades y/o analizando un sistema software existente.

Hay dos tipos de requerimientos: funcionales (QUE debe hacer el sistema), y no funcionales (otros requisitos sobre el entorno (sistema operativo, sistema gestor de base de datos, sistema de archivos, ...), ergonómicos (interfaz gráfica, etc..), de rendimiento, de tiempo, formato de entrega, etc... )

### 4.1. Estudio previo

La gestión de CV no es algo nuevo, por lo que antes de empezar el análisis de los requerimientos se hizo un análisis de los gestores de CV on-line ya existentes en la actualidad.

Se comenzó por buscar páginas web dedicadas a gestionar bolsas de trabajo de dos tipos principales; para usuarios en general o para usuarios especialistas en algún sector concreto. En la siguiente tabla se muestran algunas de las páginas consultadas.

<b>BOLSAS DE TRABAJO PARA CUALQUIER TIPO DE USUARIO</b>	<b>BOLSAS DE TRABAJO PARA USUARIOS ESPECIALIZADOS</b>
<a href="http://www.infojobs.net">www.infojobs.net</a> <a href="http://www.laboris.net">www.laboris.net</a> <a href="http://www.trabajofacil.com">www.trabajofacil.com</a> <a href="http://www.hispavista.com">www.hispavista.com</a> <a href="http://www.bolsadetrabajo.com">www.bolsadetrabajo.com</a> <a href="http://www.monster.es">www.monster.es</a> <a href="http://www.oficinaempleo.com">www.oficinaempleo.com</a> <a href="http://www.adecco.es">www.adecco.es</a> <a href="http://www.primerempleo.es">www.primerempleo.es</a>	<a href="http://www.tecnoempleo.com">www.tecnoempleo.com</a> <a href="http://www.tecnojobs.net">www.tecnojobs.net</a> <a href="http://www.cooiar.net">www.cooiar.net</a> (Colegio Oficial de Ingenieros Industriales de Aragón y la Rioja) <a href="http://www.eic.es">www.eic.es</a> (colegio oficial de ingenieros industriales de Catalunya)

El análisis de estas bolsas de trabajo se centró en hacer una comparativa sobre dos criterios:

- **organización de los datos requeridos en el currículum (formularios)**

A priori se espera que los datos requeridos al usuario para completar su currículum sean más específicos en el segundo tipo de bolsas de trabajo que en el primero. A posteriori se comprueba que no es así en la mayoría de los casos.

Generalmente los datos que se piden en las páginas para cualquier tipo de usuario son más detallados incluso que aquellos que se piden en las páginas para usuarios específicos. Este hecho responde seguramente al objetivo de querer abarcar el mayor número de perfiles posibles y así satisfacer al mayor número de usuarios.

La principal desventaja que tienen las bolsas de trabajo que piden datos tan detallados es la gran cantidad de pantallas que hay rellenar. Esto implica disminuir la facilidad de uso de las mismas y una inversión de tiempo considerable por parte del usuario. En consecuencia puede hacer que los usuarios, sobre todo los menos familiarizados con el manejo de navegadores web, dejen de usar estas páginas.

Como ejemplo de bolsas de trabajo con poco volumen de datos a rellenar tenemos a [www.cooiar.net](http://www.cooiar.net) (Colegio Oficial de Ingenieros Industriales de Aragón y la Rioja) y [www.tecnoempleo.com](http://www.tecnoempleo.com) . A continuación mostramos imágenes de los formularios para la inserción de los datos.



**Nuevo candidato.**

Rellene la ficha de curriculum y pulse "ENVIAR".  
Los campos marcados con (\*) no pueden dejarse en blanco.

Datos personales :	
Nombre (*) :	<input type="text"/>
Apellidos (*) :	<input type="text"/>
Género (*) :	Hombre <input type="button" value="v"/>
Fecha de nacimiento (*) :	dd/mm/aaaa <input type="text"/>
Dirección (*) :	<input type="text"/>
Código postal (*) :	<input type="text"/>
Municipio (*) :	<input type="text"/>
Provincia (*) :	Seleccione una Provincia <input type="button" value="v"/>
Teléfono :	<input type="text"/>
Teléfono 2 :	<input type="text"/>
Correo electrónico :	<input type="text"/>
Nombre de acceso (*) :	<input type="text"/>
Clave de acceso (*) :	Sharco <input type="text"/>
Curriculum :	
Nivel de estudios :	Seleccione un nivel <input type="button" value="v"/>
Titulación :	<input type="text"/>
Formación complementaria :	<input type="text"/>
Informática :	<input type="text"/>
Idioma1 :	Escoja Idioma <input type="button" value="v"/> L: Leido <input type="button" value="v"/> E: Escrito <input type="button" value="v"/> H: Hablado <input type="button" value="v"/>
Idioma2 :	Escoja Idioma <input type="button" value="v"/> L: Leido <input type="button" value="v"/> E: Escrito <input type="button" value="v"/> H: Hablado <input type="button" value="v"/>
Idioma3 :	Escoja Idioma <input type="button" value="v"/> L: Leido <input type="button" value="v"/> E: Escrito <input type="button" value="v"/> H: Hablado <input type="button" value="v"/>
Categoría profesional :	Seleccione una categoría <input type="button" value="v"/>
Experiencia :	Seleccione experiencia <input type="button" value="v"/>
Movilidad laboral :	Seleccione ámbito <input type="button" value="v"/>
Disponibilidad :	Seleccione disponibilidad <input type="button" value="v"/>
Permiso de conducción :	Seleccione una opción <input type="button" value="v"/>
Vehículo propio :	Seleccione una opción <input type="button" value="v"/>
Otros datos de interés :	<input type="text"/>
<input type="button" value="Enviar"/>	

formulario de [www.cooiar.net](http://www.cooiar.net) (Colegio Oficial de Ingenieros Industriales de Aragón y la Rioja)

# TECNOEMPLEO

## bolsa de trabajo

Para hacernos llegar su perfil personal rellene por favor el siguiente formulario. Los campos marcados con un asterisco (\*) deben rellenarse obligatoriamente.

\* Nombre:

\* 1er. apellido:

2º apellido:

\* Sexo:  Varón  Mujer

\* Fecha de nacimiento(DD/MM/AA):

\* NIF/Pasaporte/Tarjeta de residente:

\* Nacionalidad:

Servicio militar/PSS:  Cumplido  Exento  Pendiente

\* Domicilio:

Población:

\* Código Postal:

Provincia:

\* País:

\* Teléfono de contacto 1:

Teléfono de contacto 2:

E-mail:

Estudios académicos finalizados de mayor nivel, especificando descripción, año de obtención del título y población:

Estudios académicos en curso o no finalizados de mayor nivel, especificando descripción, último curso realizado o actual y población:

Otra formación específica:

Experiencia profesional, especificando nombre de Empresa, sector, puesto ocupado, periodo de inicio y finalización:

Idiomas:

Observaciones:

Le garantizamos una absoluta confidencialidad en el tratamiento informático de sus datos.

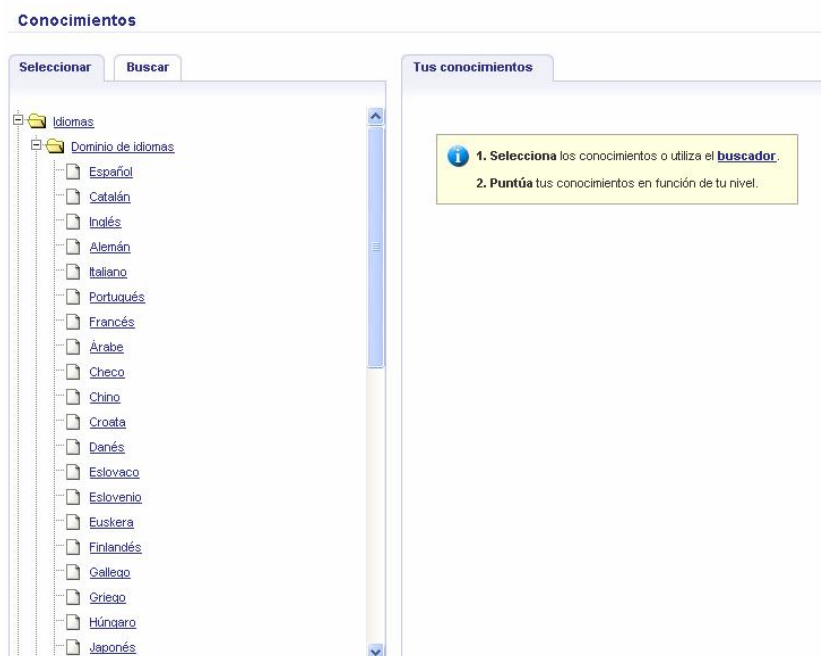
copyright © intec

formulario de [www.tecnoempleo.com](http://www.tecnoempleo.com)

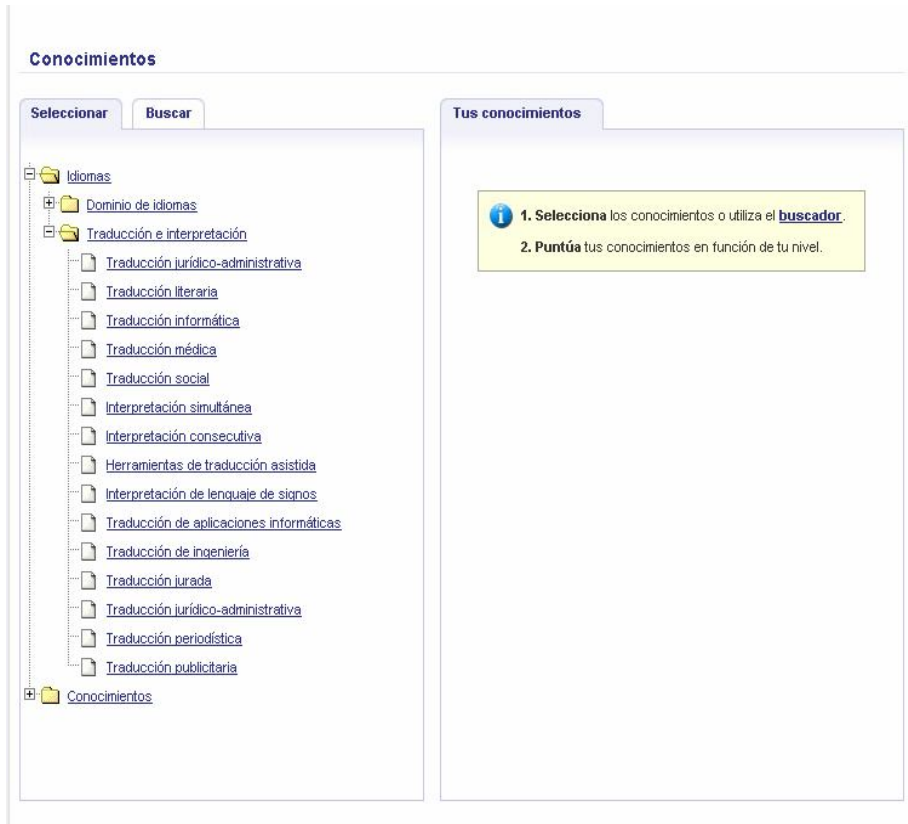
Como ejemplo de bolsa de trabajo con mucho volumen de datos a rellenar tenemos a [www.infojobs.net](http://www.infojobs.net). A continuación mostramos diferentes aspectos la pantalla con más nivel de detalle, que corresponde a los datos sobre los conocimientos del usuario.



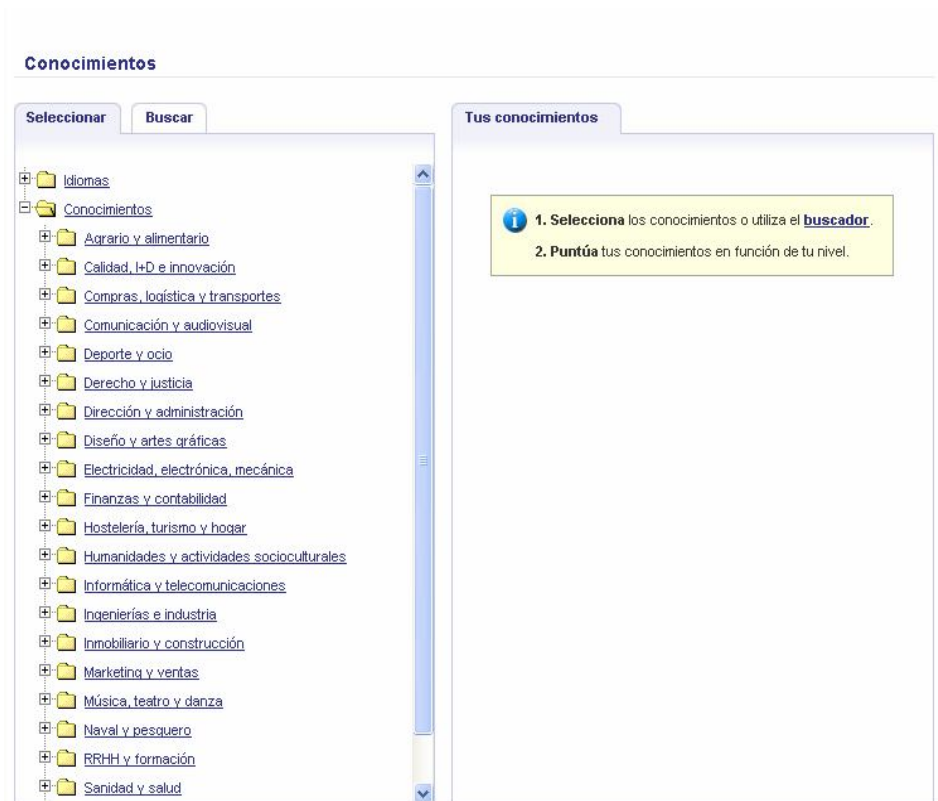
formulario datos sobre conocimientos [www.infojobs.net](http://www.infojobs.net)



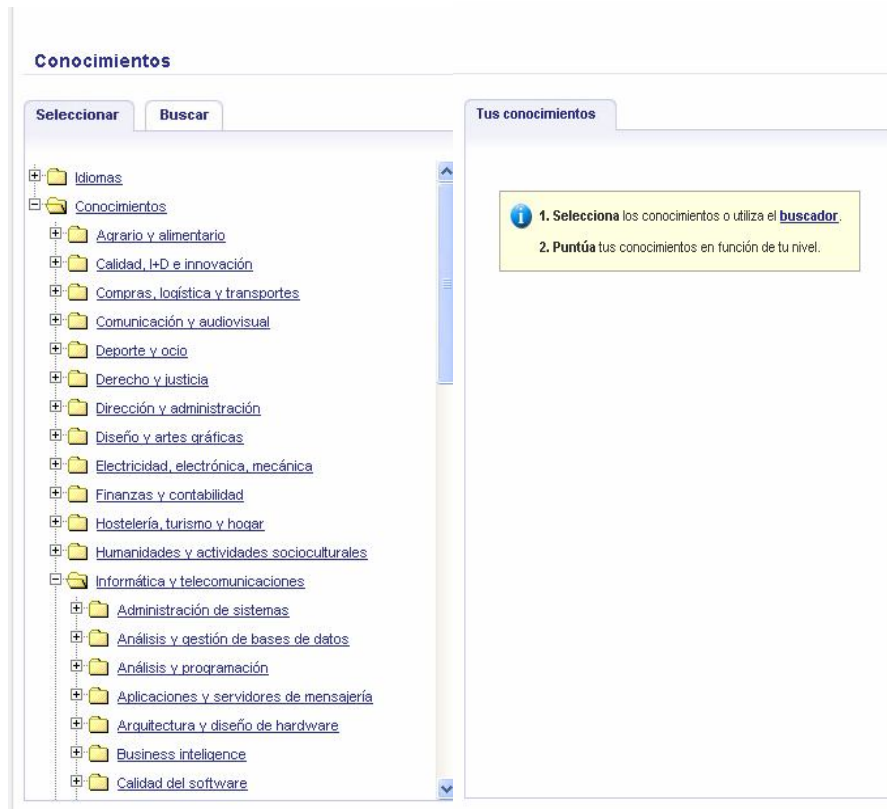
Despliegue de los conocimientos sobre idiomas/dominio de idiomas



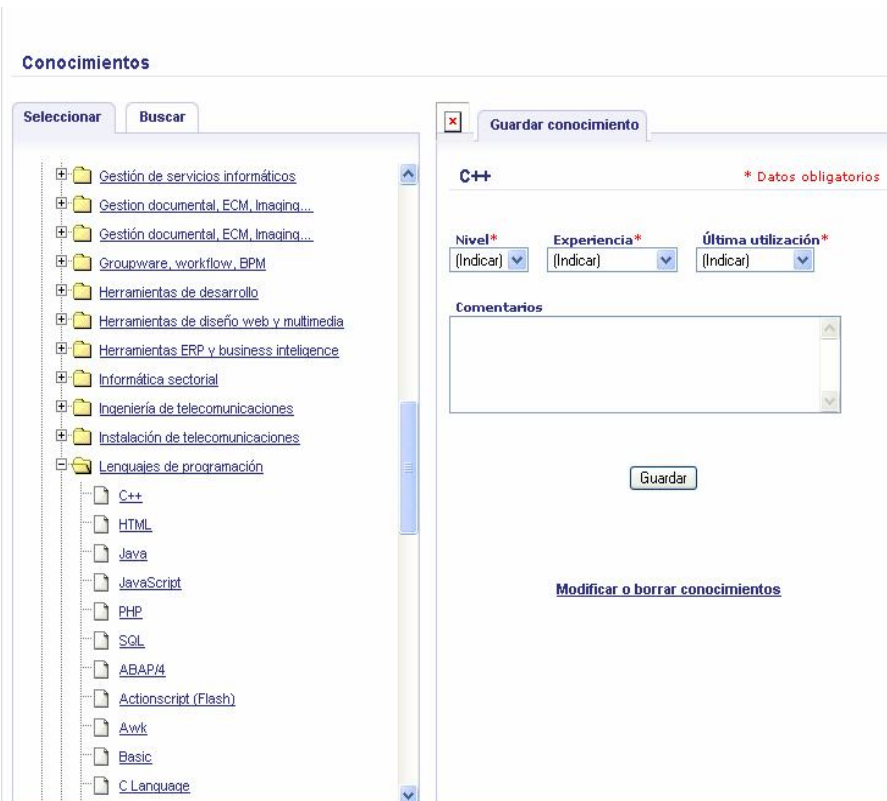
**Despliegue de los conocimientos sobre idiomas/traducción e interpretación**



**Despliegue resto tipo de conocimientos/Informática y telecomunicaciones**



**Despliega resto tipo de conocimientos/Informática y telecomunicaciones**



**Despliega resto tipo de conocimientos/Informática y telecomunicaciones/Lenguajes de programación**

- **funcionalidades del sistema.**

Las funcionalidades de los sistemas estudiados no difieren demasiado. Se realizan las operaciones básicas de gestión de bases de datos (inserción, modificación, consulta y borrado de datos). Además en la mayoría se implementan novedades como la posibilidad de adjuntar archivos con los currículums, cartas de presentación y/o fotos de los usuarios, la posibilidad de generar versiones para imprimir e incluso de generar páginas web con los datos rellenos en los formularios de inserción de CV.

Realizado este estudio previo se plantea que el módulo de gestión de CV, objetivo de este proyecto, pudiera ser capaz de:

- adaptarse dinámicamente a los contenidos de los formularios, desde los más genéricos hasta los más pormenorizados
- ser fácil y rápido de usar, adaptándose a los conocimientos sobre navegadores web de cualquier tipo de usuario
- realizar las funcionalidades estándar de un gestor de CV
- mejorar la gestión de consultas de CV y la presentación de resultados para que la selección de candidatos por parte de las empresas sea más eficiente

## 4.2. Requerimientos funcionales

Para determinar los requerimientos primero de todo se identificaron dos perfiles de usuario dentro del sistema: el **usuario principal** y el **administrador**.

Las necesidades del **usuario principal** determinan los siguientes requerimientos funcionales:

- Tener registrados los datos personales, los datos académicos, los datos sobre su vida laboral, los conocimientos sobre idiomas e informática, así como sus expectativas laborales. Estos datos se podrán registrar rellenando un formulario además de poder adjuntar un archivo con el currículum y/o carta de presentación escrito por el mismo usuario.
- Poder modificar los datos que se tienen registrados
- Poder eliminar los datos que se tienen registrados
- Poder consultar todos los datos registrados, tanto de los datos rellenados en el formulario como del fichero adjuntado, presentándolos en PDF

Las necesidades del **administrador** determinan los siguientes requerimientos funcionales:

- Gestionar los formularios. Añadir, borrar o modificar campos de entrada referentes a actividades económicas y puestos de trabajo realizados en los datos sobre la vida laboral y expectativas laborales del usuario, conocimientos sobre idiomas e informática y las especialidades en los datos académicos.
- Poder borrar los datos que tiene registrado un usuario principal
- Poder buscar filtrando los datos registrados por los usuarios principales obteniendo como resultado una lista de aquellos usuarios que cumplan los criterios del filtro. El filtro será lo más fino posible, pudiendo preguntar por cada uno de los aspectos de los datos del usuario.
- Poder seleccionar un usuario concreto de la lista resultante del filtrado para consultar todos sus datos, presentándolos en PDF
- Mantener estadísticas sobre los usuarios y sus datos

La descripción detallada de las funcionalidades finales de la aplicación se realizará en el apartado de Especificación.

### **4.3. Requerimientos no funcionales**

Una vez vistos los requerimientos funcionales pasamos a enumerar qué otros aspectos debemos tener en cuenta a la hora de diseñar la aplicación:

- Gestor de CV on-line, compatible con cualquier navegador
- Uso de los lenguaje de programación PHP (para el manejo de bases de datos) , HTML (para la creación de las páginas web) y JavaScript (para la validación de los campos de entrada en los formularios)
- Uso del servidor web Apache
- Uso de Mysql como sistema gestor de bases de datos SQL
- Aplicación segura, uso del protocolo https



## 5. ESPECIFICACIÓN

La especificación se describirá con detalle el comportamiento externo del sistema, es decir, desde el punto de vista del usuario final. Partiendo de los requerimientos se describe cada funcionalidad del sistema de forma clara, concisa y sin ambigüedad. Se define aquello que debe y que no debe hacer el sistema.

Para realizar la especificación de un proyecto se utilizan lenguajes que permiten plasmar los requerimientos del usuario en notación de orientación a objetos. No son lenguajes de programación, únicamente nos sirven para abstraer de forma gráfica los requerimientos y realizar el diseño posterior del sistema.

**UML** es el lenguaje más extendido para la especificación y diseño de proyectos, hasta el punto de convertirse en un estándar en 1997, por lo tanto será el que usaremos en este proyecto.

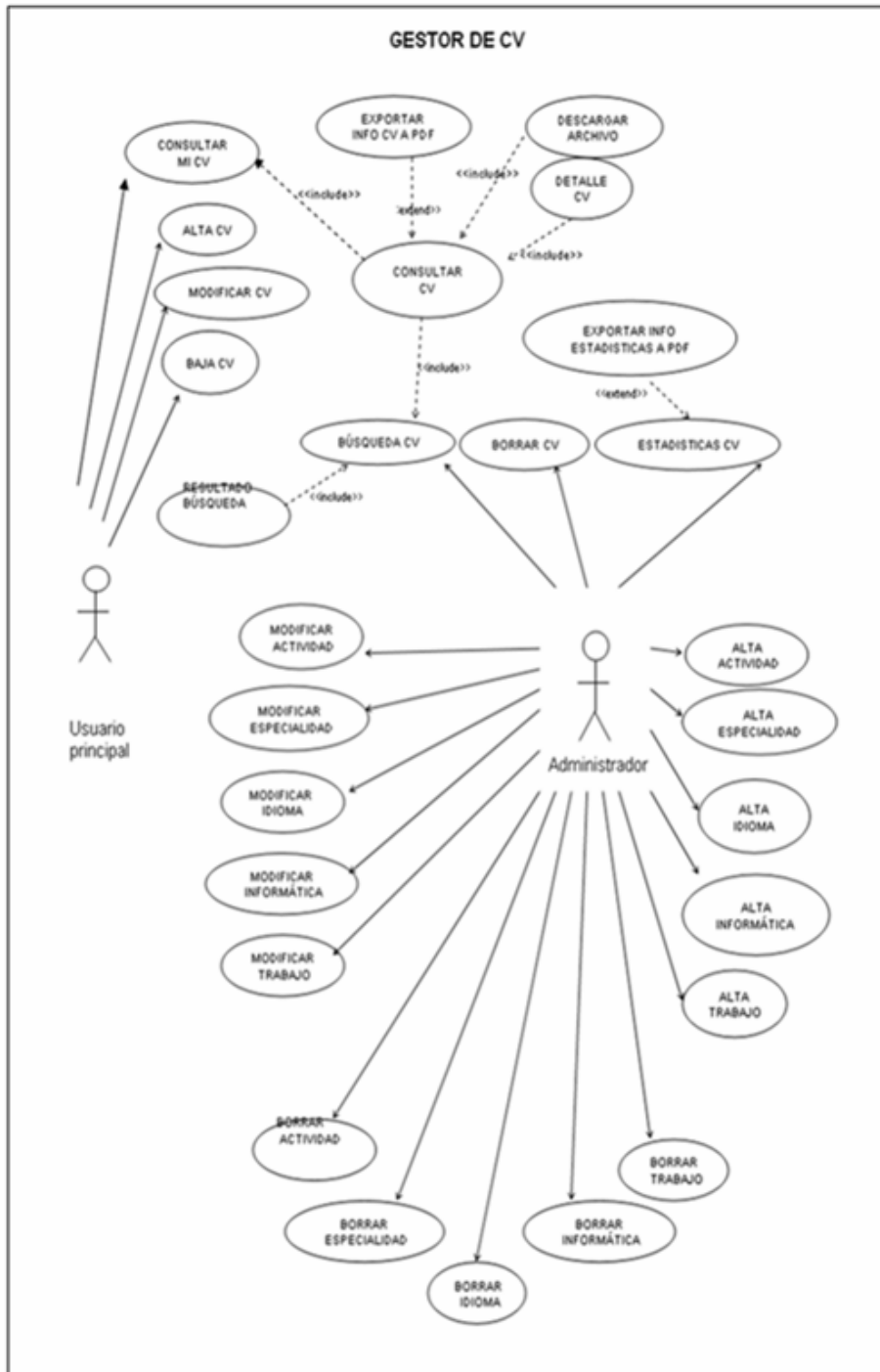
### 5.1. Modelo de Casos de Uso

El modelo de casos de uso identifica las funcionalidades que ha de tener el sistema y las asocia a cada actor/usuario.

El modelo de casos de uso engloba dos partes:

- el diagrama de casos de uso que muestra conjuntamente los diferentes casos de uso del sistema, los actores y las relaciones entre actores y casos de uso
- la especificación de los caso de uso donde se describen brevemente las acciones del caso de uso

### 5.1.1. Diagrama de Casos de Uso



### 5.1.2. Especificación de los Casos de Uso

La Especificación de Casos de Uso nos permite describir de acciones y los requerimientos del caso de uso. En la especificación de un caso de uso se sigue el siguiente esquema para realizar de una manera ordena dicha descripción.

- **Caso de uso:** nombre del caso de uso
- **Actores:** lista de actores, agentes externos al sistema que participan en él. El actor **iniciador** es aquel actor que inicia el caso de uso
- **Resumen:** objetivo del caso de uso
- **Curso típico de acontecimientos:** Descripción detallada (conversación) entre los actores y el sistema
- **Section:** Alternativas que puede seguir el caso de uso en función de las elecciones del actor o de las comprobaciones del sistema
- **Cursos alternativos:** Descripción de las excepciones al curso típico

A continuación comenzamos con la especificación de los casos de uso.

**Alta CV****Caso de uso :** Alta CV**Actors :** Usuario principal (iniciador)**Resumen :** Dar de alta un CV en el sistema**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
1.- El caso de uso comienza cuando uno de los usuarios principales quiera dar de alta su CV.	
	2.- El sistema genera una página con el formulario y los campos necesarios para añadir un CV.
3.- El usuario introduce los datos que se le piden.	
	4.- Se comprueban los datos rellenados por el usuario.
	5.- El sistema pide confirmación del alta de CV
6.- El usuario decide si confirma el alta de su CV.	
	7.- Se registran en el sistema los datos
8.- El usuario recibe por pantalla la confirmación del alta del CV	

**Cursos alternativos:**

- Línea 4: Ya existe un CV con el mismo NIF.
- Línea 4: Alguno de los datos introducidos son incorrectos.
- Línea 6: El usuario decide no confirmar el alta de su CV. Se cancela el proceso.

**Modificar CV****Caso de uso:** Modificar CV**Actores:** Usuario principal (iniciador)**Resumen:** modificar un CV del sistema**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
1.- El caso de uso comienza cuando el usuario quiere modificar su CV	
	2.- El sistema genera una página donde el usuario debe introducir su NIF.
3.- El usuario introduce los datos que se le piden.	
	4.- Se comprueban el NIF rellenados por el usuario.
	5.- El sistema consulta el CV y genera una página con los datos introducidos en el momento del alta, para poder ser modificados.
6.- El usuario modifica los datos que necesite	
	7.- Se comprueban los datos rellenados por el usuario.
	8.- El sistema pide confirmación de la modificación del CV
9.- El usuario decide si confirma la modificación de su CV.	
	9.- Se modifican los datos en el sistema.
10.- El usuario recibe por pantalla la confirmación de modificación del CV	

**Cursos alternativos:**

- Línea 4: No existe un CV con el NIF introducido por el usuario para poder ser modificado.
- Línea 4: El NIF introducido es incorrecto.
- Línea 7: Alguno de los datos introducidos son incorrectos.
- Línea 9: El usuario decide no confirmar la modificación de su CV. Se cancela el proceso.

**Baja CV****Caso de uso:** Baja CV**Actores:** Usuario principal (iniciador)**Resumen:** Dar de baja un CV en el sistema.**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
1.- El caso de uso comienza cuando el usuario quiere dar de baja su CV.	
	2.- El sistema genera una página con un formulario donde se pide el NIF del CV a dar de baja.
3.- El usuario introduce los datos que se le piden.	
	4.- Se comprueban los datos rellenados por el usuario y pide confirmación de baja de CV.
5.- El usuario decide si confirma la baja de su CV.	
	6.- El sistema consulta el CV.
	7.- Se da de baja el CV en el sistema. El CV cambia de estado pero no se elimina completamente.
8.- El administrador recibe por pantalla la confirmación de la baja del CV	

**Cursos alternativos:**

- Línea 4: No existe un CV con el NIF introducido por el administrador para poder dar de baja.
- Línea 4: El NIF introducido es incorrecto.
- Línea 5: El usuario decide no confirmar la baja de su CV. Se cancela el proceso.

**Consultar mi CV****Caso de uso:** Consultar mi CV**Actores:** Usuario principal (iniciador)**Resumen:** consulta del CV del usuario principal, dado su NIF.**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
1.- El caso de uso comienza cuando el usuario quiere consultar su CV.	
	2.- El sistema genera una página donde el usuario debe introducir su NIF.
3.- El usuario introduce los datos que se le piden.	
	4.- Se comprueban los datos rellenados por el usuario y se pide confirmación de consulta.
5.- El usuario decide si confirma la consulta de CV	
	6.- Se genera una página con un enlace para ver el detalle del NIF del usuario
7.- <b>Si</b> el administrador quiere ver el detalle del CV, siguiendo el enlace de la página <b>see section</b> consulta_CV	

**Cursos alternativos:**

- Línea 4: No existe un CV con el NIF introducido por el usuario para poder ser consultado.
- Línea 4: El NIF introducido es incorrecto.
- Línea 5: El administrador decide no confirmar la consulta de CV. Se cancela el proceso.

**Búsqueda CV****Caso de uso:** Búsqueda CV**Actores:** Administrador (iniciador)**Resumen:** Búsqueda de un CV según unos criterios de búsqueda.**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
1.- El caso de uso comienza cuando el administrador quiere buscar un CV.	
	2.- El sistema genera una página donde el administrador puede introducir sus criterios de búsqueda.
3.- El administrador introduce los criterios de búsqueda.	
	4.- Se comprueban los criterios de búsqueda rellenos por el administrador y pide confirmación de búsqueda de CV.
5.- El administrador decide si confirma la búsqueda de CV.	
	6.- El sistema busca todos los CV que coinciden con los criterios de búsqueda.
	7.- Se genera una página con el listado y contador de CV encontrados, con un resumen de los mismos y con enlaces para ver en detalle cada uno de los CV encontrados. <b>include</b> Resultado Búsqueda
8.- El administrador recibe el listado con los recursos que cumplen sus criterios de búsqueda.	
<b>Si</b> el administrador quiere ver el detalle de algún CV en concreto del listado, siguiendo el enlace de la página <b>see section</b> consulta CV	

**Cursos alternativos:**

- Línea 5: El administrador decide no confirmar la búsqueda de CV. Se cancela el proceso.
- Línea 6 : No se encuentra ningún CV que cumpla los criterios introducidos.



**Section:** consulta\_CV**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
	1.- El sistema genera una página donde el actor podrá escoger que tipo de detalle del CV quiere ver.
2.- El actor escoge el informe con el tipo de detalle del CV que quiere ver.	
<b>Si</b> el actor quiere ver el detalle del archivo adjunto en el formulario de alta <b>see section</b> descargar_archivo  <b>Si</b> el actor quiere ver el detalle de los datos del CV introducido en el formulario de alta <b>see section</b> detalle_CV	

**Cursos alternativos :**

· Línea 2: El administrador decide no escoger ninguno de los tipos de detalle del CV ofrecidos por el sistema. Se cancela el proceso.

**Section:** descargar\_archivo**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
	1.- El sistema descarga el archivo adjuntado por el usuario en el alta
2.- El actor ve el archivo en pantalla	

**Section:** detalle\_CV

**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
	1.- El sistema genera una página con los datos del CV introducido en el formulario de alta.
2.- El actor ve los datos por pantalla en formato PDF	

### **Resultado Búsqueda**

**Caso de uso:** Resultado Búsqueda

**Actores:** -

**Resumen:** Se obtiene un listado de los CV que cumplen los criterios de búsqueda

**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
	1.- El sistema muestra una relación de los CV que cumplen los criterios de búsqueda. Devuelve un listado con un resumen de los datos de cada CV

### **Exportar info CV a PDF**

Extiende el caso de uso Consultar CV. El sistema exporta la información obtenida a formato PDF

**Borrar CV****Caso de uso:** Borrar CV**Actores:** Administrador (iniciador)**Resumen:** Eliminar un CV definitivamente del sistema.**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
1.- El caso de uso comienza cuando el administrador quiere eliminar un CV.	
	2.- El sistema genera una página con un formulario donde se pide el NIF del CV a borrar.
3.- El administrador introduce los datos que se le piden.	
	4.- Se comprueban los datos rellenados por el administrador y pide confirmación de borrado de CV.
5.- El administrador decide si confirma el borrado de su CV.	
	6.- El sistema consulta el CV.
	7.- Se elimina completamente el CV en el sistema.
8.- El administrador recibe por pantalla la confirmación del borrado del CV	

**Cursos alternativos:**

- Línea 4: No existe un CV con el NIF introducido por el administrador para poder ser borrado.
- Línea 4: El NIF introducido es incorrecto.
- Línea 5: El administrador decide no confirmar el borrado de su CV. Se cancela el proceso.

**Estadísticas****Caso de uso:** Estadísticas**Actores:** Administrador (iniciador)**Resumen:** Consultar las estadísticas generadas por el sistema sobre los CV dados de alta.**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
1.- El caso de uso comienza cuando el administrador quiere consultar las estadísticas de los CV dados de alta.	
	2.- El sistema genera una página donde el administrador podrá escoger que tipo de estadísticas quiere ver. Muestra una relación de los diferentes estudios estadísticos realizados en el sistema
3.- El administrador escoge el informe que quiere ver. <b>see section</b> Resumen Estadísticas de los CV	
	4.- El sistema consulta los CV y genera una página con los datos del informe escogido.
5.- El administrador ve los datos por pantalla en PDF.	

**Cursos alternativos:**

- Línea 3: El administrador decide no escoger ninguno de los tipos de estadísticas ofrecidos por el sistema. Se cancela el proceso.

**Section:** Resumen Estadísticas de los CV**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
	1.- El sistema muestra una relación de los CV según su estado.
2.- El actor ve los datos por pantalla en formato PDF	

**Exportar estadísticas a PDF**

Extiende el caso de uso Estadística CV. El sistema exporta la información obtenida a formato PDF.

**Alta Actividad**

Los casos de uso *Alta especialidad*, *Alta idioma*, *Alta informática* y *Alta Trabajo* son análogos a este caso de uso.

**Caso de uso:** Alta Actividad

**Actores:** Administrador (iniciador)

**Resumen:** Dar de alta una actividad económica en el sistema

**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
1.- El caso de uso comienza cuando el administrador quiere dar de alta una nueva actividad económica para que aparezca en los formularios.	
	2.- El sistema genera una página con el formulario y los campos necesarios para añadir una actividad económica.
3.- El administrador introduce los datos que se le piden.	
	4.- Se comprueban los datos rellenados por el administrador
	5.- El sistema pide confirmación del alta de la actividad económica
6.- El administrador decide si confirma el alta de la nueva actividad económica.	
	7.- Se registran en el sistema los datos
8.- El administrador recibe por pantalla la confirmación del alta de la nueva actividad económica	

**Cursos alternativos:**

- Línea 4: Alguno de los datos introducidos son incorrectos.
- Línea 6: El administrador decide no confirmar el alta de la actividad económica. Se cancela el proceso.

**Modificar Actividad**

Los casos de uso *Modificar especialidad*, *Modificar idioma*, *Modificar informática* y *Modificar Trabajo* son análogos a este caso de uso.

**Caso de uso:** Modificar Actividad

**Actores:** Administrador (iniciador)

**Resumen:** modificar una actividad económica del sistema

**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
1.- El caso de uso comienza cuando el administrador quiere modificar una actividad económica.	
	2.- El sistema genera una página con el formulario y los campos necesarios para modificar una actividad económica.
3.- El administrador introduce los datos que se le piden.	
	4.- Se comprueban los datos rellenados por el administrador
	5.- El sistema pide confirmación de la modificación de la actividad económica.
6.- El administrador decide si confirma la modificación de la actividad económica.	
	7.- Se modifican en el sistema los datos
8.- El administrador recibe por pantalla la confirmación de la modificación de la actividad económica	

**Cursos alternativos:**

- Línea 4: Alguno de los datos introducidos son incorrectos.
- Línea 6: El administrador decide no confirmar la modificación de la actividad económica. Se cancela el proceso.

**Borrar Actividad**

Los casos de uso *Borrar especialidad*, *Borrar idioma*, *Borrar informática* y *Borrar Trabajo* son análogos a este caso de uso.

**Caso de uso:** Borrar Actividad

**Actores:** Administrador (iniciador)

**Resumen:** Eliminar definitivamente una actividad económica del sistema

**Curso típico de acontecimientos:**

Acciones de los actores	Respuesta del sistema
1.- El caso de uso comienza cuando el administrador quiere borrar una actividad económica.	
	2.- El sistema genera una página con el formulario y los campos necesarios para borrar una actividad económica.
3.- El administrador introduce los datos que se le piden.	
	4.- Se comprueban los datos rellenados por el administrador
	5.- El sistema pide confirmación del borrado de la actividad económica.
6.- El administrador decide si confirma el borrado de la actividad económica.	
	7.- Se elimina completamente del sistema los datos
8.- El administrador recibe por pantalla la confirmación de la modificación de la actividad económica	

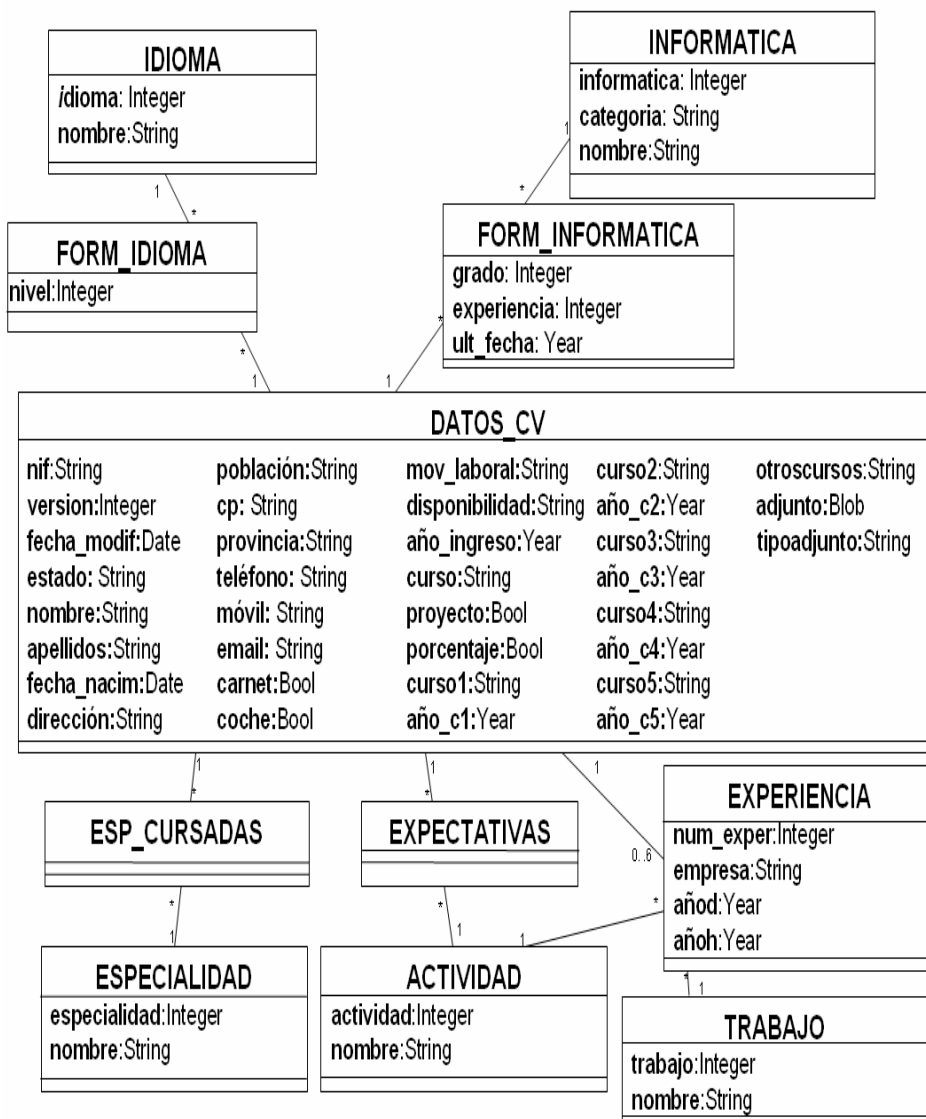
**Cursos alternativos:**

- Línea 4: Alguno de los datos introducidos son incorrectos.
- Línea 6: El administrador decide no confirmar la modificación de la actividad económica. Se cancela el proceso.

## 5.2. Modelo conceptual

El Modelo Conceptual representa los conceptos más significativos en el dominio del problema utilizando Clases de Objetos, Asociación entre clases de Objetos y Atributos de las clases. Nos describe la parte estática del problema, no la parte dinámica del sistema.

A continuación se muestra el modelo conceptual obtenido para nuestro sistema representado con un **diagrama de clases** normalizado.





### 5.3. Esquema de la arquitectura

Al ser un sistema basado en la Web e Internet el cual tiene que soportar el acceso simultáneo y concurrente de varios usuarios, la arquitectura propuesta es la **Arquitectura Cliente-Servidor (C/S)**.

La **arquitectura Cliente/Servidor** es una nueva tendencia en el desarrollo de redes locales, debido a la necesidad de compartir datos, para poder optimizar los recursos de hardware y software. El paradigma cliente-servidor tiene como objetivo optimizar el uso tanto del hardware como del software a través de la separación de funciones: el servidor administra la parte que debe ser compartida por varios usuarios, y el cliente maneja sólo lo particular de cada usuario.

Las principales características de la arquitectura C/S son:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

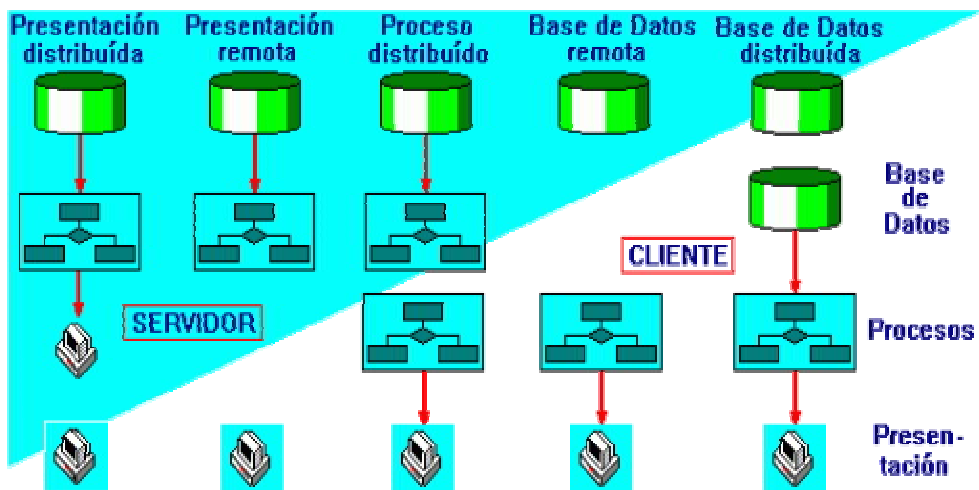
Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario
- Captura y validación de los datos de entrada
- Generación de consultas e informes sobre las bases de datos

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos
- Control de accesos concurrentes a bases de datos compartidas
- Enlaces de comunicaciones con otras redes de área local o extensa

Esta arquitectura se puede clasificar en cinco niveles, según las funciones que asumen el cliente y el servidor, tal y como se puede ver en el siguiente diagrama:



**Presentación distribuida:** El cliente asume parte de las funciones de presentación de la aplicación, ya que siguen existiendo programas en el servidor dedicados a esta tarea. Esta técnica dificulta su mantenimiento en las aplicaciones orientadas a terminales. Además, el servidor ejecuta todos los procesos y almacena la totalidad de los datos.

**Presentación remota:** La aplicación está soportada directamente por el servidor, excepto la presentación que es totalmente remota y reside en el cliente. Los terminales del cliente soportan la captura de datos, incluyendo una validación parcial de los mismos y una presentación de las consultas.

**Proceso distribuido:** La lógica de los procesos se divide entre el cliente y el servidor. El diseñador de la aplicación debe definir los servicios y las interfaces del sistema de información de forma que los papeles de cliente y servidor sean intercambiables, excepto en el control de los datos que es responsabilidad exclusiva del servidor.

**Gestión de datos remota:** El cliente realiza tanto las funciones de presentación como los procesos. Por su parte, el servidor almacena y gestiona los datos que permanecen en una base de datos centralizada.

**Bases de datos distribuidos:** El reparto de tareas es como en el caso anterior y además el gestor de base de datos divide sus componentes entre el cliente y el servidor. Las interfaces entre ambos están dentro de las funciones del gestor de datos y, por lo tanto, no tienen impacto en el desarrollo de las aplicaciones.

Para el diseño de esta aplicación, se ha elegido la configuración de la arquitectura cliente-servidor con bases de datos distribuidas. El cliente será un navegador Web y se encarga de presentación de los datos y de realizar peticiones al servidor. El servidor recibe, procesa accediendo a los datos en caso necesario y responde a las peticiones con la información demandada por el cliente para que éste la presente correctamente.

## 5.4. Modelo de comportamiento

El Modelo de Comportamiento define la parte dinámica del sistema, es decir, cual debe ser el comportamiento en cada situación y la forma de proceder. Los diagramas de secuencia, los contratos de las operaciones del sistema.

### 5.4.1. Diagramas de secuencia

Los diagramas de secuencia del sistema toman como punto de partida a los casos de uso y muestran, para cada escenario particular de un caso de uso, los **acontecimientos** generados por los actores externos, su orden y los acontecimientos internos en el sistema (**operaciones**) que resultan de su invocación.

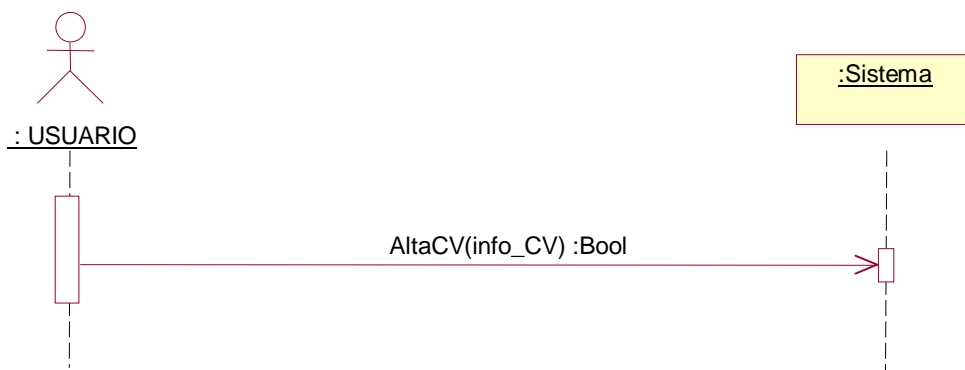
Se define un diagrama de secuencia para cada curso relevante de acontecimientos de un caso de uso. Para construir un diagrama de secuencia se siguen los pasos descritos a continuación:

- Dibujar una línea vertical que representa el **sistema**.

- Dibujar una línea para cada **actor** que interactúa **directamente** con el sistema.
- Del curso de acontecimientos del caso de uso, identificar los acontecimientos externos generados por los actores y mostrarlos en el diagrama.

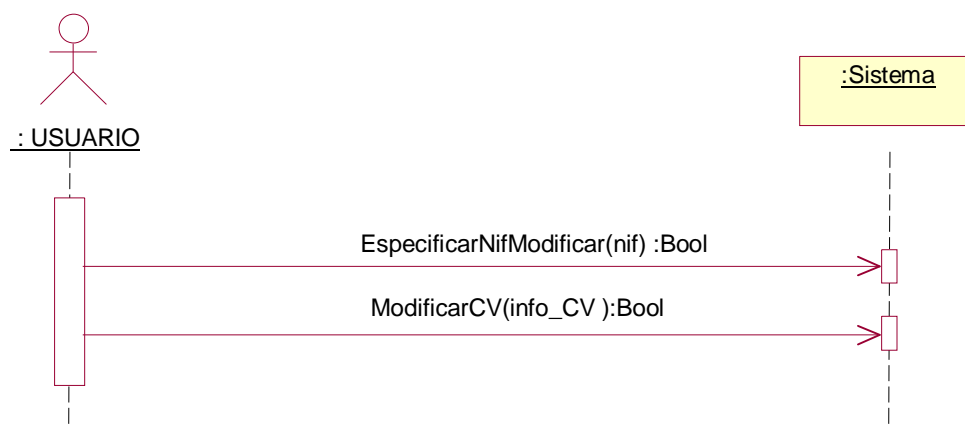
A continuación se muestra los diagramas de secuencia obtenidos para nuestro sistema.

### Alta CV



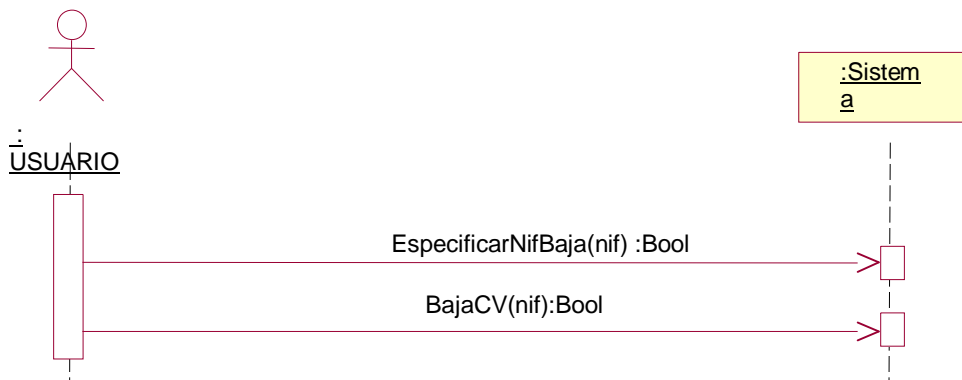
info\_CV = nif + nombre + apellidos + fecha\_nacim + dirección + población + cp + provincia + teléfono+ móvil +email + carnet+coche + mov\_laboral + disponibilidad + año\_ingreso + curso + proyecto+ porcentaje + curso1 + año\_c1 +curso2 + año\_c2 + curso3 + año\_c3 + curso4 + año\_c4 + curso5 + año\_c5+otroscursos+ adjunto + tipoadjunto + {idioma, nivel}+ {informatica, categoría, grado, experiencia, ult\_fecha} + {especialidad} + {actividad}+ {num\_exper, empresa, añod, añooh,actividad, trabajo}

### Modificar CV

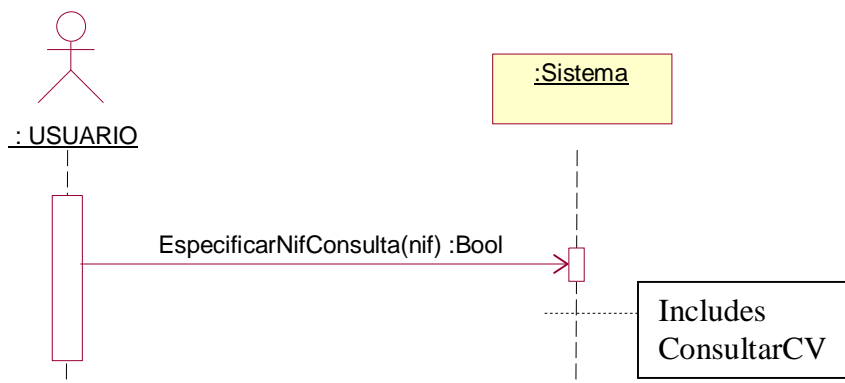


Info\_CV = nif + nombre + apellidos + fecha\_nacim + dirección + población + cp + provincia + teléfono + móvil +email + carnet +coche + mov\_laboral + disponibilidad + año\_ingreso + curso + proyecto+ porcentaje + curso1 + año\_c1 + curso2 + año\_c2 + curso3 + año\_c3 + curso4 + año\_c4 + curso5 + año\_c5 + otros cursos +adjunto + tipoadjunto+{idioma, nivel}+{informatica, categoría, grado, experiencia, ult\_fecha} +{especialidad} + {actividad}+ {num\_exper, empresa, añod, año, actividad, trabajo}

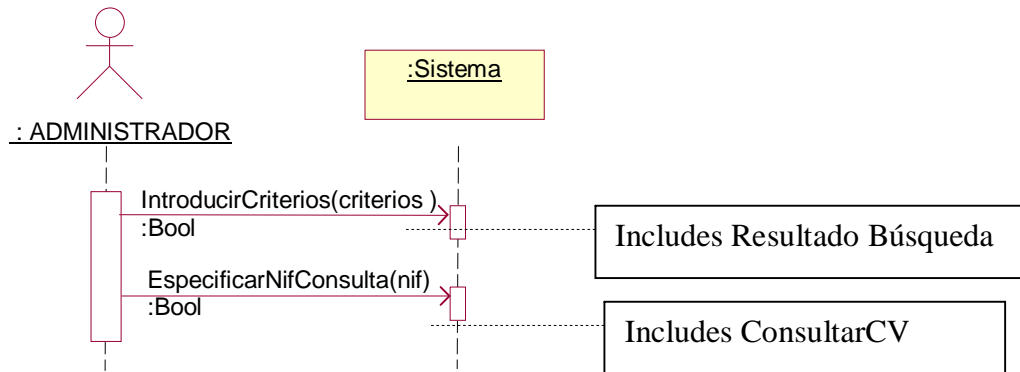
**Baja CV**



**Consultar Mi CV**

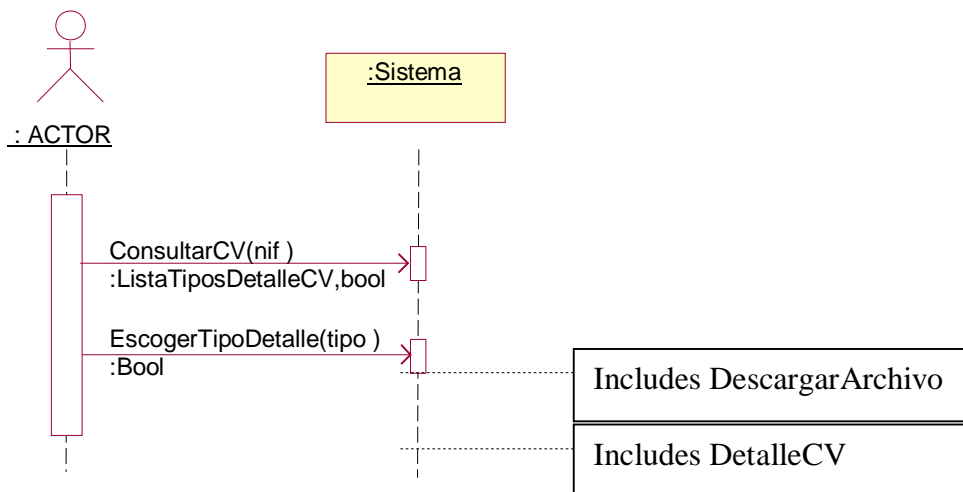


**Búsqueda CV**

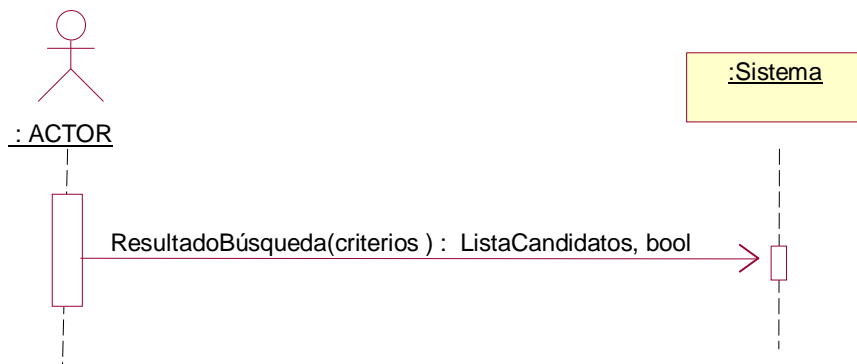


criterios = carnet + coche + mov\_laboral + disponibilidad + año\_ingresod +  
 año\_ingresoh + cursod + cursoh + proyecto + porcentaje + otros cursos +  
 {idioma, nivel}+{informatica, categoría} + {especialidad} + {actividad}+  
 {actividad, trabajo}

**Consultar CV**

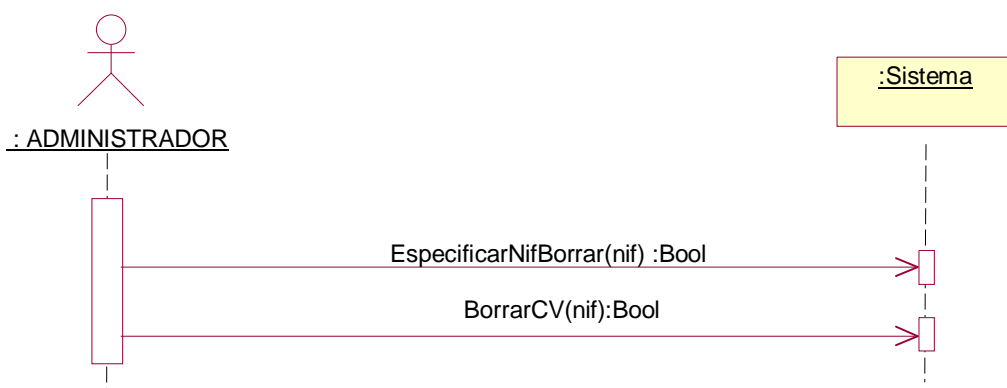


ListaTiposDetalleCV: {Descargar archivo adjunto, Exportar CV a PDF}

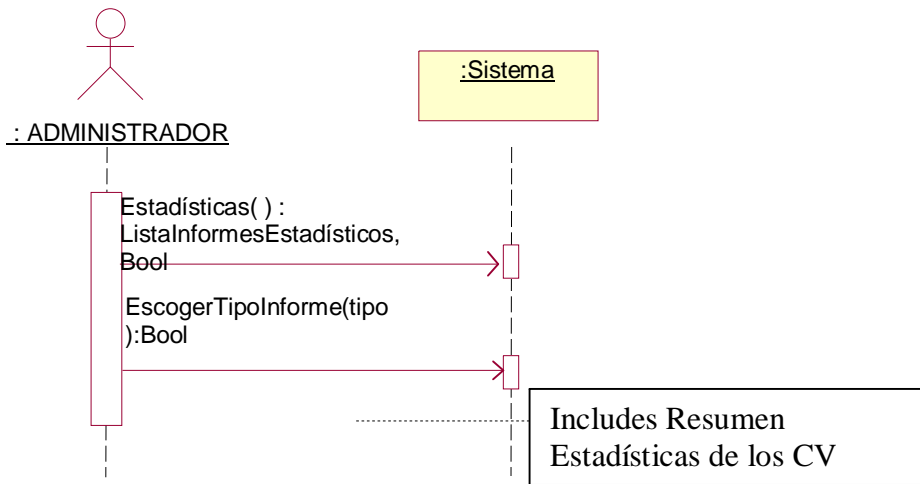
**Resultado Búsqueda**

criterios = carnet + coche + mov\_laboral + disponibilidad + año\_ingresod +  
 año\_ingresoh + cursod + cursoh + proyecto + porcentaje + otros cursos +  
 {idioma, nivel} + {informatica, categoría} + {especialidad} + {actividad} +  
 {actividad, trabajo}

ListaCandidatos = nif + año\_ingreso + curso + proyecto + porcentaje + {especialidad} +  
 {informatica} + {actividad + trabajo} + {actividad}

**Borrar CV**

**Estadísticas**



ListaInformesEstadísticos: {Exportar Resumen de los CV a PDF}

**Alta Actividad**

Los diagramas de secuencia *Alta especialidad*, *Alta idioma*, *Alta informática* y *Alta Trabajo* son análogos a este.

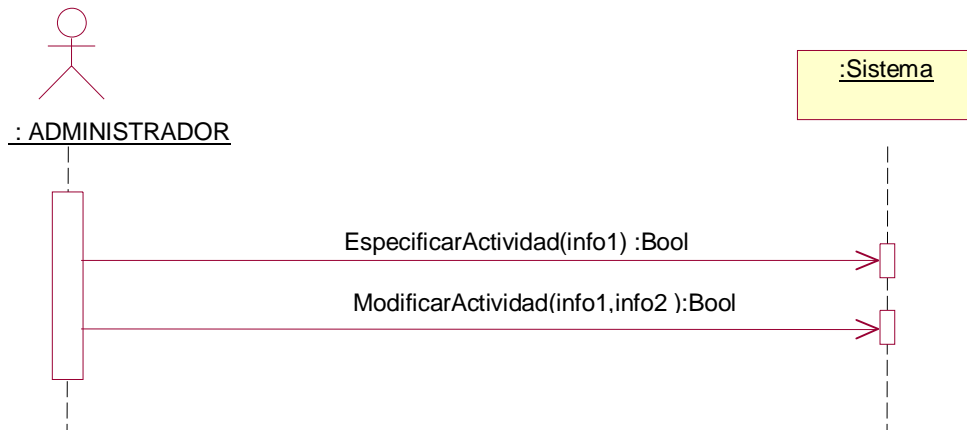


- info (Alta Actividad) = actividad
- info (Alta Especialidad) = especialidad
- info (Alta Idioma) = idioma
- info (Alta Informática) = informática + categoría
- info (Alta Trabajo) = trabajo



**Modificar Actividad**

Los diagramas de secuencia *Modificar especialidad*, *Modificar idioma*, *Modificar informática* y *Modificar Trabajo* son análogos a este.



info1 e info2 (Modificar Actividad) = actividad

info1 e info2 (Modificar Especialidad) = especialidad

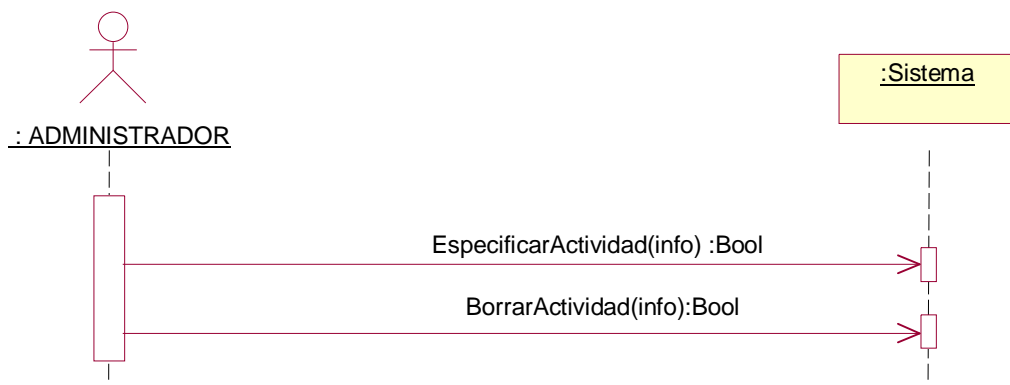
info1 e info2 (Modificar Idioma) = idioma

info1 e info2 (Modificar Informática) = informática + categoría

info1 e info2 (Modificar Trabajo) = trabajo

**Borrar Actividad**

Los diagramas de secuencia *Borrar especialidad*, *Borrar idioma*, *Borrar informática* y *Borrar Trabajo* son análogos a este.



info (Borrar Actividad) = actividad

info (Borrar Especialidad) = especialidad

info (Borrar Idioma) = idioma

info (Borrar Informática) = informática + categoría

info (Borrar Trabajo) = trabajo

## 5.4.2. Operaciones del sistema

Las operaciones del sistema se agrupan como operaciones del tipo especial “Sistema”. En cambio, las operaciones no se asignan a objetos concretos durante la etapa de especificación.

SISTEMA
AltaCV(info_CV) : Boolean
EspecificarNifModificar(nif) : Boolean
ModificarCV(info_CV) : Boolean
EspecificarNifBaja(nif) : Boolean
BajaCV(nif) : Boolean
EspecificarNifConsulta(nif) : Boolean
ConsultaCV(nif):ListaTiposDetalleCV, Boolean
EscogerTipoDetalle(tipo) : Boolean
IntroducirCriterios(criterios) : Boolean
ResultadoBúsqueda(criterios):ListaCandidatos, Boolean
EspecificarNifBorrar(nif) : Boolean
BorrarCV(nif) : Boolean
Estadisticas( ):ListaInformesEstadisticos, Boolean
EscogerTipoInforme(tipo) : Boolean
EspecificarActividad(actividad) : Boolean
AltaActividad(actividad) : Boolean
ModificarActividad(actividad1,actividad2) : Boolean
BorrarActividad(actividad) : Boolean
EspecificarEspecialidad(especialidad) : Boolean
AltaEspecialidad (especialidad) : Boolean
ModificarEspecialidad (especialidad1,especialidad2) : Boolean
BorrarEspecialidad (especialidad) : Boolean
EspecificarIdioma(idioma) : Boolean
AltaIdioma(idioma) : Boolean
ModificarIdioma(idioma1,idioma2) : Boolean
BorrarIdioma(idioma) : Boolean
EspecificarInformatica(informática, categoría) : Boolean
AltaInformatica(informática, categoría) : Boolean
ModificarInformatica(informática1, categoría2, informática2, categoría2) : Boolean
BorrarInformatica(informática, categoría) : Boolean
EspecificarTrabajo(trabajo) : Boolean
AltaTrabajo(trabajo) : Boolean
ModificarTrabajo(trabajo1,trabajo2) : Boolean
BorrarTrabajo(trabajo) : Boolean

### 5.4.3. Contratos de las operaciones

Los contratos de las operaciones describen el comportamiento del sistema en términos de cuáles son los **cambios de estado** de la base de información y cuales son las **salidas** que el sistema proporciona cuando se invoca la operación.

El tipo de descripción es declarativo, el énfasis se pone en **qué** hará la operación más que en **cómo** lo hará. Los contratos de las operaciones incluyen primordialmente las **precondiciones** y **post-condiciones** que describen los cambios de estado y las **salidas**.

Los componentes de los contratos de las operaciones son básicamente:

**Nombre:** Nombre y argumentos de la operación (signatura de la operación)

**Responsabilidades:** Descripción informal del propósito de la operación

**Caso de uso:** Caso de uso con el que esta relacionada la operación

**Precondiciones:** Descripción de lo que se asume sobre el estado del sistema antes de la invocación de la operación

**Post-condiciones:** Cambios de estado que se han producido

**Salida:** Descripción de la salida que proporciona la operación.

A continuación se muestra los contratos de las operaciones para nuestro sistema.

#### AltaCV

**Nombre:** AltaCV(info\_CV)

**Responsabilidades:** Registrar un CV

**Caso de uso:** Alta CV

**Precondiciones:** Los argumentos obligatorios han de tener valor, todos los valores de los argumentos han de ser correctos.

**Post-condiciones:**

1. Si se da alguna de las siguientes condiciones la operación es inválida y se devuelve Falso
  - 1.1. Ya existe el CV con el nif de entrada y el estado del CV es diferente a 'dado de baja'
  - 1.2. Si se han rellenado los datos de conocimientos de idiomas y el nif de entrada ya tiene formación en dicho idioma, siempre y cuando el CV ya existe y su estado es diferente a 'dado de baja'
  - 1.3. Si se han rellenado los datos de conocimientos informáticos y el nif de entrada ya tiene formación en dicho conocimiento informático siempre y cuando el CV ya existe y su estado es diferente a 'dado de baja'

- 1.4. Si se han rellenado los datos de las especialidades cursadas y el nif de entrada ya cursa dicha especialidad, siempre y cuando el CV ya existe y su estado es diferente a 'dado de baja'
- 1.5. Si se han rellenado los datos de las expectativas profesionales y el nif de entrada ya tiene como expectativa profesional dicha actividad, siempre y cuando el CV ya existe y su estado es diferente a 'dado de baja'
- 1.6. Si se han rellenado los datos de experiencia profesional y el nif de entrada ya tiene experiencia profesional en dichos actividad y trabajo, siempre y cuando el CV ya existe y su estado es diferente a 'dado de baja'
- 1.7. El nif ya tiene 6 experiencias profesionales registradas
2. En caso contrario, la operación es válida, se devuelve Cierto y
  - 2.1. Se crea un nuevo objeto DADES\_CV con estado 'activo', la fecha\_modif es la fecha del proceso y la versión es 1, siempre y cuando el CV no existe. Si el CV del nif de entrada ya existe y su estado es 'dado de baja' se cambia el estado a 'activo', la fecha\_modif es la fecha del proceso y la versión se aumenta en una unidad con respecto a la existente.
  - 2.2. Siempre y cuando el CV no existe y se han rellenado los datos de conocimientos de idiomas, se crea un nuevo objeto FORM\_IDIOMA con las correspondientes asociaciones entre DADES\_CV e IDIOMA. En caso contrario se haría como en la operación ModificarCV.
  - 2.3. Siempre y cuando el CV no existe y se han rellenado los datos de conocimientos informáticos, se crea un nuevo objeto FORM\_INFORMATICA con las correspondientes asociaciones entre DADES\_CV e INFORMATICA. En caso contrario se haría como en la operación ModificarCV.
  - 2.4. Siempre y cuando el CV no existe y se han rellenado los datos de las especialidades cursadas, se crea un nuevo objeto ESP\_CURSADAS con las correspondientes asociaciones entre DADES\_CV y ESPECIALIDAD. En caso contrario se haría como en la operación ModificarCV.
  - 2.5. Siempre y cuando el CV no existe y se han rellenado los datos de las expectativas profesionales, se crea un nuevo objeto EXPECTATIVAS con las correspondientes asociaciones entre DADES\_CV y ACTIVIDAD. En caso contrario se haría como en la operación ModificarCV.
  - 2.6. Siempre y cuando el CV no existe y se han rellenado los datos de experiencia profesional, se crea un nuevo objeto EXPERIENCIA con las correspondientes asociaciones entre DADES\_CV, ACTIVIDAD y TRABAJO y num\_exper aumentando cada vez en una unidad. En caso contrario se haría como en la operación ModificarCV.

**Salida:** Boolean

**EspecificarNifModificar**

**Nombre:** EspecificarNifModificar(nif)

**Responsabilidades:** Inicia un proceso de modificación de CV dado el nif

**Caso de uso:** Modificar CV

**Precondiciones:** El nif a modificar debe tener valor

**Post-condiciones:**

1. Si el valor del nif es correcto se devuelve Cierto
2. en caso contrario se devuelve Falso

**Salida:** Boolean

**ModificarCV**

**Nombre:** Modificar(info\_CV)

**Responsabilidades :** Modificar los datos registrados de un CV

**Caso de uso:** Alta CV

**Precondiciones:** Los argumentos obligatorios han de tener valor, todos los valores de los argumentos han de ser correctos.

**Post-condiciones:**

1. Si se da alguna de las siguientes condiciones la operación es inválida y se devuelve Falso
  - 1.1. No existe el CV con el nif de entrada o si existe el estado del CV es 'dado de baja'
  - 1.2. El nif ya tiene 6 experiencias profesionales registradas y no se quiere modificar ninguna de éstas.
2. En caso contrario, la operación es válida, se devuelve Cierto y
  - 2.1. Se modifican los datos del objeto DADES\_CV correspondiente al nif de entrada, el estado se pone a 'activo', la fecha\_modif es la fecha del proceso y la versión se aumenta en una unidad con respecto a la existente.
  - 2.2. Si según los datos de entrada el objeto FORM\_IDIOMA con sus correspondientes asociaciones entre DADES\_CV e IDIOMA ya existe, se modifican los atributos del objeto. En caso contrario se crea un nuevo objeto FORM\_IDIOMA con las correspondientes asociaciones.
  - 2.3. Si según los datos de entrada el objeto FORM\_INFORMÁTICA con sus correspondientes asociaciones entre DADES\_CV e INFORMÁTICA ya existe, se modifican los atributos del objeto. En caso contrario se crea un nuevo objeto FORM\_INFORMATICA con las correspondientes asociaciones. Si se da el caso de que exista un objeto FORM\_INFORMATICA cuyos datos de entrada no se hayan informado se eliminará el objeto y sus correspondientes asociaciones.
  - 2.4. Si según los datos de entrada el objeto ESP\_CURSADAS con sus correspondientes asociaciones entre DADES\_CV y ESPECIALIDAD ya existe no se realiza ningún tipo de operación sobre el objeto. En caso contrario se crea un nuevo objeto ESP\_CURSADAS con las correspondientes asociaciones. Si se da el caso de que exista un objeto ESP\_CURSADAS cuyos datos de entrada no se hayan informado se eliminará el objeto y sus correspondientes asociaciones.
  - 2.5. Si según los datos de entrada el objeto EXPECTATIVAS con sus correspondientes asociaciones entre DADES\_CV y ACTIVIDAD ya existe no se realiza ningún tipo de operación sobre el objeto. En caso contrario se crea un nuevo objeto EXPECTATIVAS con las

correspondientes asociaciones. Si se da el caso de que exista un objeto EXPECTATIVAS cuyos datos de entrada no se hayan informado se eliminará el objeto y sus correspondientes asociaciones.

- 2.6. Si según los datos de entrada el objeto EXPERIENCIA con sus correspondientes asociaciones entre DADES\_CV, ACTIVIDAD y TRABAJO ya existe, se modifican los atributos del objeto. En caso contrario se crea un nuevo objeto EXPERIENCIA con las correspondientes asociaciones y num\_exper una unidad mayor. Si se da el caso de que exista un objeto EXPERIENCIA cuyos datos de entrada no se hayan informado se eliminará el objeto y sus correspondientes asociaciones.

**Salida:** Boolean

### **EspecificarNifBaja**

**Nombre:** EspecificarNifBaja(nif)

**Responsabilidades :** Inicia un proceso de baja de CV dado el nif

**Caso de uso:** Baja CV

**Precondiciones:** El nif a dar de baja debe tener valor

**Post-condiciones:**

1. Si el valor del nif es correcto se devuelve Cierto
2. en caso contrario se devuelve Falso

**Salida:** Bolean

### **BajaCV**

**Nombre:** Baja(nif)

**Responsabilidades :** Da de baja un CV

**Caso de uso:** Baja CV

**Precondiciones:**

**Post-condiciones:**

1. Si se da alguna de las siguientes condiciones la operación es inválida y se devuelve Falso
  - 1.1. No existe el CV con el nif de entrada o si existe el estado del CV es 'dado de baja'
2. En caso contrario, la operación es válida, se devuelve Cierto y
  - 2.1. Se modifican los datos del objeto DADES\_CV correspondiente al nif de entrada, el estado se pone a 'dado de baja' y la fecha\_modif es la fecha del proceso.

**Salida:** Boolean

### **EspecificarNifConsulta**

**Nombre:** EspecificarNifConsulta(nif)

**Responsabilidades :** Inicia un proceso de consulta de CV dado el nif

**Caso de uso:** Consulta Mi CV

**Precondiciones:** El nif a consultar debe tener valor

**Post-condiciones:**

1. Si el valor del nif es correcto se devuelve Cierto,
2. en caso contrario se devuelve Falso

**Salida:** Boolean

### **ConsultaCV**

**Nombre:** ConsultaCV(nif):ListaTiposDetalleCV

**Responsabilidades :** Consulta los datos del CV del nif de entrada

**Caso de uso:** Consultar CV

**Precondiciones:** El nif a consultar debe tener valor y ser correcto

**Post-condiciones:**

1. Si se da alguna de las siguientes condiciones la operación es inválida y se devuelve Falso
  - 1.1. No existe el objeto CV correspondiente al nif de entrada que se quiere consultar o si existe tienen estado diferente a 'activo'
2. En caso contrario, la operación es válida, se devuelve Cierto y
  - 2.1. Se devuelve una lista de enlaces con los datos consultados, por un lado el detalle del CV (datos de los objetos DADES\_CV, FORM\_IDIOMA, FORM\_INFORMÁTICA, ESP\_CURSADAS, EXPECTATIVAS y EXPERIENCIA correspondientes al nif de entrada) y por otro lado al archivo adjunto del objeto DADES\_CV (si este existe)

**Salida:** Boolean

### **EscogerTipoDetalle**

**Nombre:** EscogerTipoDetalle(tipo)

**Responsabilidades :** Muestra los datos del CV consultado según el tipo de entrada

**Caso de uso:** Consultar CV

**Precondiciones:** Existen los datos consultados

**Post-condiciones:**

1. La operación es válida, devuelve Cierto y
  - 1.1. Si el tipo es ver detalle del CV, se exportan los datos a PDF y se muestran
  - 1.2. Si el tipo es ver archivo adjunto, se descarga el archivo y se muestra

**Salida:** Boolean

### IntroducirCriterios

**Nombre:** IntroducirCriterios(criterios)

**Responsabilidades:** Inicia un proceso de consulta avanzada de CV's

**Caso de uso:** Búsqueda CV

**Precondiciones:** Los criterios de búsqueda deben tener valor

**Post-condiciones:**

3. Si los valores de los criterios son correctos se devuelve Cierto,
4. en caso contrario se devuelve Falso

**Salida:** Boolean

### ResultadoBúsqueda

**Nombre:** ResultadoBúsqueda(criterios):ListaCandidatos

**Responsabilidades :** Filtra los CV según los criterios de búsqueda de entrada y devuelve un listado de los CV que cumplan el filtro, así como el total de éstos, y un resumen de sus datos.

**Caso de uso:** Resultado Búsqueda

**Precondiciones:** Los criterios de búsqueda deben tener valor y ser correctos

**Post-condiciones:**

1. la operación es válida, se devuelve Cierto y se devuelve también
  - 1.1. Una lista con los CV que cumplen los criterios y con estado 'activo', además se muestran algunos datos de los objetos DADES\_CV, FORM\_IDIOMA, FORM\_INFORMÁTICA, ESP\_CURSADAS, EXPECTATIVAS y EXPERIENCIA.
  - 1.2. El número total de los CV que cumplen los criterios
  - 1.3. Enlaces con la consulta de cada CV por nif

**Salida:** Boolean

### EspecificarNifBorrar

**Nombre:** EspecificarNifBorrar(nif)

**Responsabilidades :** Inicia un proceso de borrado de CV dado el nif

**Caso de uso:** Borrar CV

**Precondiciones:** El nif a borrar debe tener valor

**Post-condiciones:**

1. Si el valor del nif es correcto se devuelve Cierto
2. en caso contrario se devuelve Falso

**Salida:** Boolean



**BorrarCV****Nombre:** Borrar(nif )**Responsabilidades :** Borra un CV**Caso de uso:** Borrar CV**Precondiciones:****Post-condiciones:**

1. Si se da alguna de las siguientes condiciones la operación es inválida y se devuelve Falso
  - 1.1. No existe el CV con el nif de entrada
2. En caso contrario, la operación es válida, se devuelve Cierto y
  - 2.1. Se elimina el objeto DADES\_CV correspondiente al nif de entrada
  - 2.2. Se elimina todos los objetos FORM\_IDIOMA que existan y sus correspondientes asociaciones entre DADES\_CV e IDIOMA, correspondientes al nif de entrada
  - 2.3. Se elimina todos los objetos FORM\_INFORMÁTICA que existan y sus correspondientes asociaciones entre DADES\_CV e INFORMÁTICA, correspondientes al nif de entrada
  - 2.4. Se elimina todos los objetos ESP\_CURSADAS que existan y sus correspondientes asociaciones entre DADES\_CV y ESPECIALIDAD, correspondientes al nif de entrada
  - 2.5. Se elimina todos los objetos EXPECTATIVAS que existan y sus correspondientes asociaciones entre DADES\_CV y ACTIVIDAD, correspondientes al nif de entrada
  - 2.6. Se elimina todos los objetos EXPERIENCIA que existan y sus correspondientes asociaciones entre DADES\_CV, ACTIVIDAD y TRABAJO, correspondientes al nif de entrada

**Salida:** Boolean**Estadísticas():ListaInformesEstadísticos****Nombre:** Estadísticas():ListaInformesEstadísticos**Responsabilidades:** Consulta las estadísticas del sistema**Caso de uso:** Estadísticas**Precondiciones:****Post-condiciones:**

1. la operación es válida, se devuelve Cierto y
  - 1.1. Se devuelve una lista de enlaces con los informes estadísticos realizados sobre los CV.

**Salida:** Boolean

**EscogerTipoInforme**

**Nombre:** EscogerTipoInforme(tipo)

**Responsabilidades :** Muestra los datos del informe estadístico consultado según el tipo de entrada

**Caso de uso:** Estadísticas

**Precondiciones:** Existen los datos consultados

**Post-condiciones:**

2. La operación es válida, devuelve Cierto y
  - 2.1. se exportan los datos a PDF y se muestran el numero de CV de cada estado y un listado de los CV cuya fecha\_mod sea mayor o igual a un año

**Salida:** Boolean

**EspecificarActividad**

Los contratos de las operaciones *Especificar especialidad*, *Especificar idioma*, *Especificar informática* y *Especificar Trabajo* son análogos a este.

**Nombre:** EspecificarActividad(actividad)

**Responsabilidades :** Inicia un proceso de modificación de una actividad

**Caso de uso:** Modificar Actividad, Borrar Actividad

**Precondiciones:** La actividad debe tener valor y ser correcto

**Post-condiciones:**

**Salida:** Boolean

**AltaActividad**

Los contratos de las operaciones *Alta especialidad*, *Alta idioma*, *Alta informática* y *Alta Trabajo* son análogos a este.

**Nombre:** AltaActividad(actividad)

**Responsabilidades :** Registrar una actividad

**Caso de uso:** Alta Actividad

**Precondiciones:** Los argumentos han de tener valor y no existe el objeto actividad que se va a dar de alta

**Post-condiciones:**

1. La operación es válida, devuelve Cierto y se crea un nuevo objeto ACTIVIDAD

**Salida:** Boolean

**ModificarActividad**

Los contratos de las operaciones *Modificar especialidad*, *Modificar idioma*, *Modificar informática* y *Modificar Trabajo* son análogos a este.

**Nombre:** ModificarActividad(actividad1,actividad2)

**Responsabilidades :** Modificar los datos registrados de una actividad

**Caso de uso:** Modificar actividad

**Precondiciones:** Los argumentos han de tener valor y existe el objeto actividad1 que se va a modificar

**Post-condiciones:**

1. La operación es válida, devuelve Cierto y se modifican los atributos del objeto actividad1.

**Salida:** Boolean

**BorrarActividad**

Los contratos de las operaciones *Borrar especialidad*, *Borrar idioma*, *Borrar informática* y *Borrar Trabajo* son análogos a este.

**Nombre:** BorrarActividad(actividad)

**Responsabilidades :** Borrar los datos registrados de una actividad

**Caso de uso:** Borrar actividad

**Precondiciones:** Los argumentos han de tener valor y existe el objeto actividad que se va a modificar

**Post-condiciones:**

1. La operación es válida, devuelve Cierto
  - 1.1. Elimina el objeto actividad.
  - 1.2. Elimina todos los objetos EXPECTATIVAS que tengan una asociación con la actividad de entrada
  - 1.3. Elimina todos los objetos EXPERIENCIA que tengan una asociación con la actividad de entrada

**Salida:** Boolean

## 6. DISEÑO

En la fase de diseño se establece el comportamiento dinámico del sistema, es decir, como debe reaccionar ante los acontecimientos. El resultado obtenido de la etapa de diseño facilita enormemente la implementación posterior de nuestro sistema, pues proporciona la estructura básica del sistema y como los diferentes componentes actúan y se relacionan entre ellos.

Una buena metodología de diseño consiste en utilizar patrones. Un patrón de diseño ofrece una solución concreta a los posibles problemas que podamos encontrar en la construcción de nuestro sistema, indicando la mejor forma de modelar los objetos

La elección de la arquitectura interna de nuestro sistema es otro de los puntos básicos de la etapa de diseño. La arquitectura más utilizada es en capas, el número de las cuáles depende de diferentes factores, aunque la más utilizada es la arquitectura en 3 capas: Presentación, Dominio y Acceso a datos. No obstante, al escoger la arquitectura dependerá también de la tecnología que utilicemos (XML Web services, .NET Remoting, etc..).

El diseño de la interfaz gráfica de usuario es otra de las tareas que se deben realizar dentro de la etapa de diseño. La interfaz de usuario de una aplicación debe de ser el máximo de intuitiva posible para el usuario final, y debe adaptarse a los procesos que los usuarios realizan para incrementar al máximo el rendimiento de los mismos.

Para almacenar los datos de la aplicación, es necesario también diseñar la estructura de la base de datos que va a contener esos datos. El diseño de la base de datos es importante para optimizar el tamaño ocupado por los datos y la agilidad de consulta de los mismos.

## 6.1. Patrón de Diseño

En la especificación se escogió la arquitectura Cliente/Servidor para el diseño de la aplicación. Una vez hecho esto se necesita escoger un patrón que proponga una solución adecuada al problema.

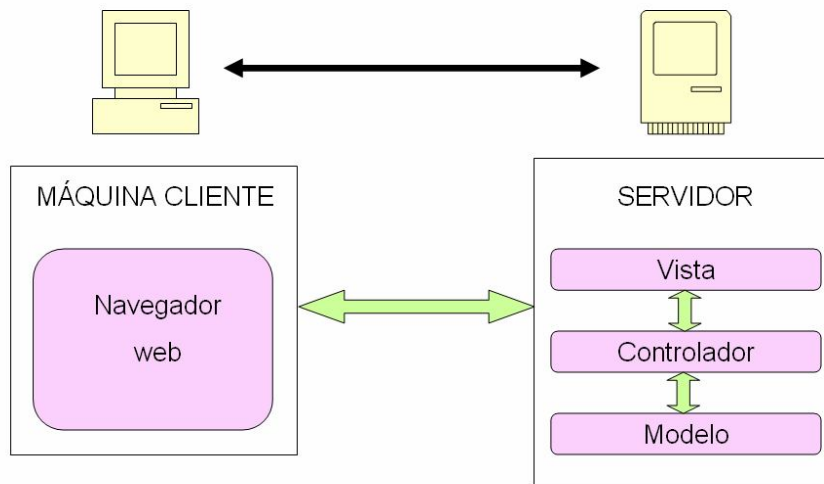
Para nuestro proyecto hemos escogido el Patrón Modelo-Vista-Controlador, por ser el más adecuado para los sistemas interactivos.

### 6.1.1. Patrón Modelo-Vista-Controlador

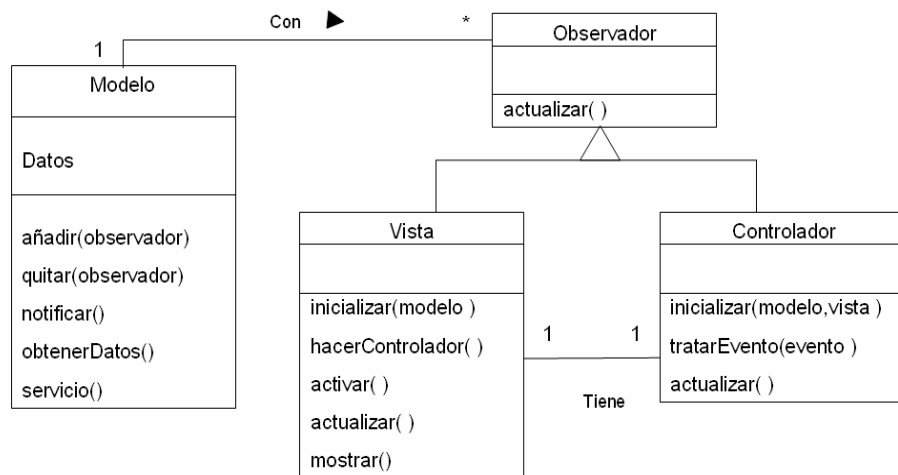
El **Modelo Vista Controlador (MVC)** es un patrón de arquitectura de software que separa los **datos** de una aplicación, la **interfaz de usuario**, y la **lógica de control** en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la *Vista* es la página HTML y el código que provee de datos dinámicos a la página, el *controlador* es el Sistema de Gestión de Base de Datos y el *Modelo* es el modelo de datos.

- **Modelo:** Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de éstos y permite derivar nuevos datos.
- **Vista:** Presenta el modelo en un formato adecuado para interactuar, la interfaz de usuario.
- **Controlador:** Responde a eventos, usualmente acciones del usuario e invoca cambios en el Modelo y probablemente en la Vista.

## ESQUEMA DE LA ARQUITECTURA DE LA APLICACIÓN



## ESTRUCTURA ESTÁTICA DEL MVC

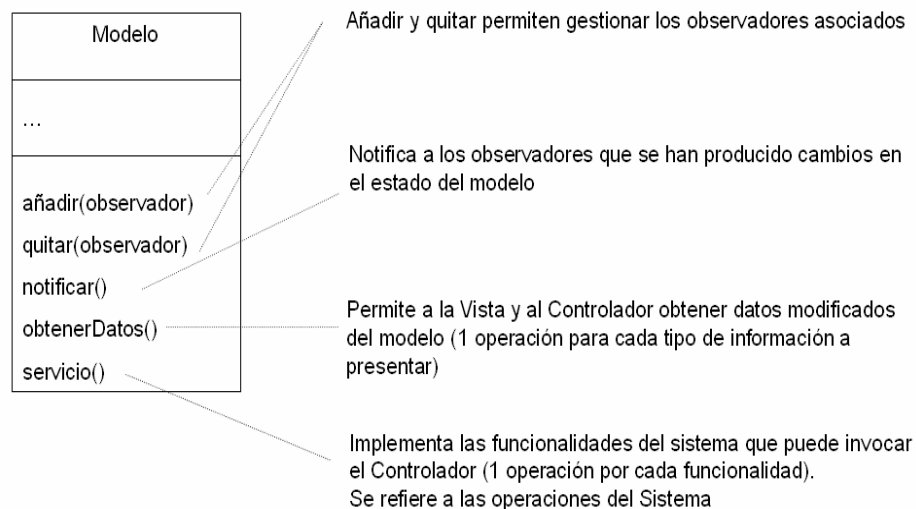


- El Modelo tiene siempre asociados n observadores
- En general, hay tantas parejas Vista-Controlador como informaciones y formatos diferentes se quieran mostrar de un modelo determinado
- Las operaciones actualizar (de los controladores), mostrar y tratarEvento se acostumbra a redefinir en las subclases de Vista y Controlador que se definen para un modelo determinado.
- Cuando el Modelo cambia todos sus observadores han de actualizarse a este cambio.

## DESCRIPCIÓN DEL MODELO

Las características que describen el Modelo son:

- El Modelo encapsula las funcionalidades y los datos del sistema.
- Es independiente de los mecanismos de presentación de la información y de la interacción con el usuario
- Proporciona al Controlador los servicios para satisfacer las peticiones del usuario
- Mantiene un mecanismo de coordinación con las Vistas y Controladores asociados (patrón Observador), para notificarles cualquier cambio en su estado.

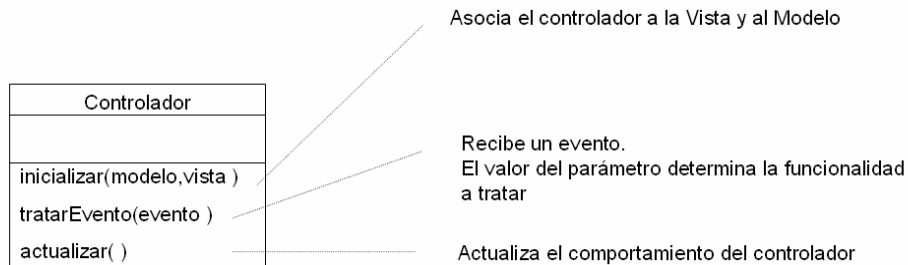


## DESCRIPCIÓN DEL CONTROLADOR

Las características que describen el Controlador son:

- El usuario interactúa con el sistema únicamente mediante Controladores
- Gestiona los eventos de presentación y de modificación del modelo generados por el usuario
- La forma como recibe estos eventos depende de la plataforma utilizada para interactuar con el usuario (manipulador de eventos)

- Traduce los eventos de presentación en invocaciones a servicios proporcionados por el Modelo y peticiones de funcionalidades propias de la Vista
- El comportamiento del Controlador puede depender del estado del Modelo

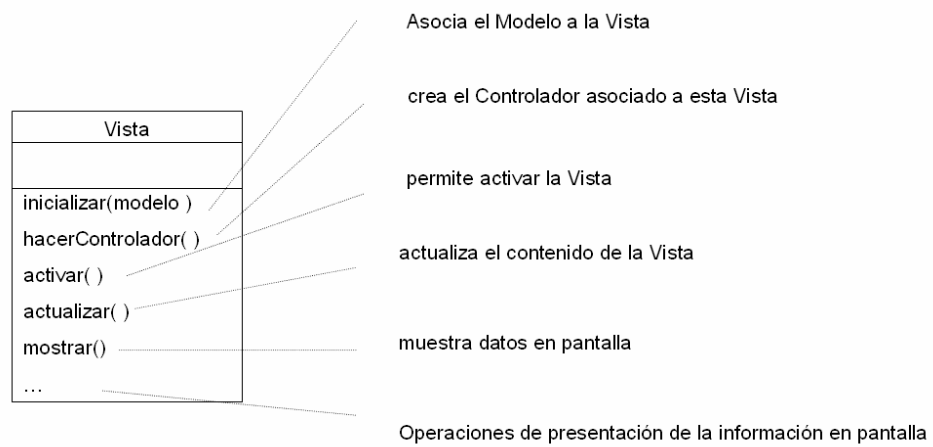


## DESCRIPCIÓN DE LA VISTA

Las características que describen la Vista son:

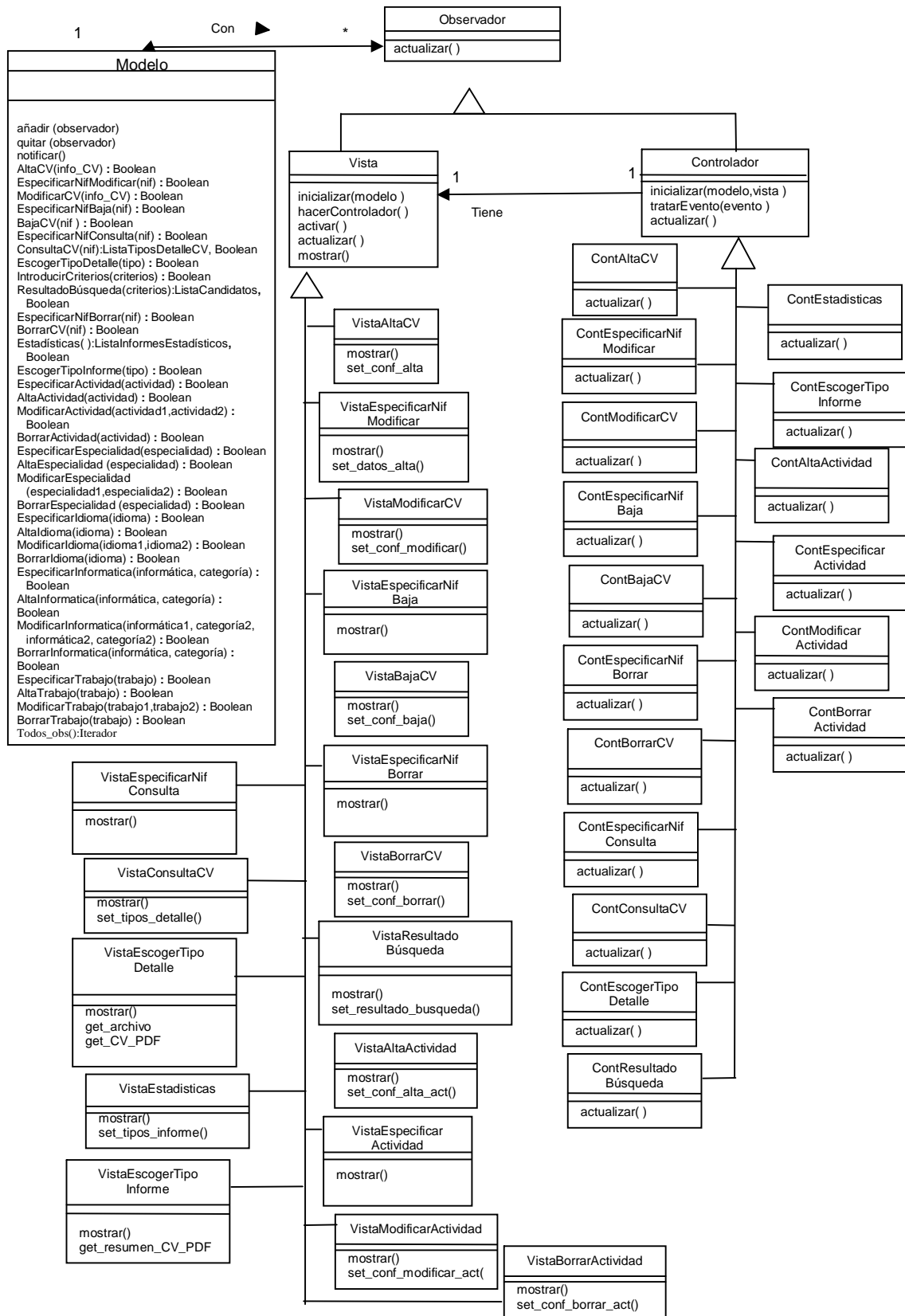
- Permite presentar información del Modelo al usuario. Puede haber diversas vistas de un mismo Modelo
- La información que muestra se puede ver afectado por cambios en el estado del Modelo
- Tiene asociado un Controlador que gestiona, en caso necesario, los eventos de modificación del Modelo
- Puede proporcionar operaciones que permiten que los Controladores gestionen la modificación de la pantalla





### 6.1.2. Aplicación del Patrón Modelo-Vista-Controlador

En la siguiente figura se ve el resultado de la aplicación del MVC sobre nuestro sistema.



En el diagrama anterior faltarían las Vistas y los controladores de EspecificarEspecialidad, AltaEspecialidad, ModificarEspecialidad, BorrarEspecialidad, EspecificarIdioma,AltaIdioma, ModificarIdioma, BorrarIdioma, EspecificarInformatica, AltaInformatica, ModificarInformatica, BorrarInformatica, EspecificarTrabajo, AltaTrabajo, ModificarTrabajo y BorrarTrabajo ya que son análogas a EspecificarActividad, AltaActividad, ModificarActividad y BorrarActividad respectivamente.

En la operación tratarEvento(evento) los ‘eventos’ serían cada uno de los servicios del modelo/funcionalidades del sistema.

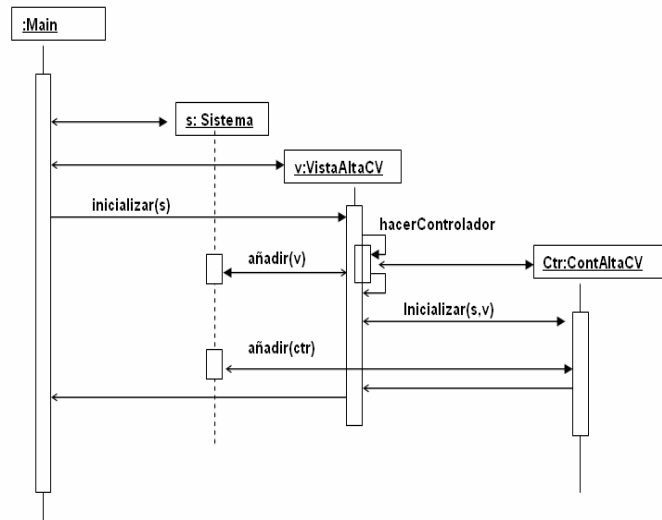
## 6.2. Diagramas de secuencia

Los diagramas de secuencia nos permiten modelar como será el funcionamiento final de las operaciones presentes en nuestro sistema, en otras palabras, nos dan un esquema de alto nivel de cómo se tendría que implementar cada operación que, fácilmente, puede ser traducido a cualquier lenguaje de programación.

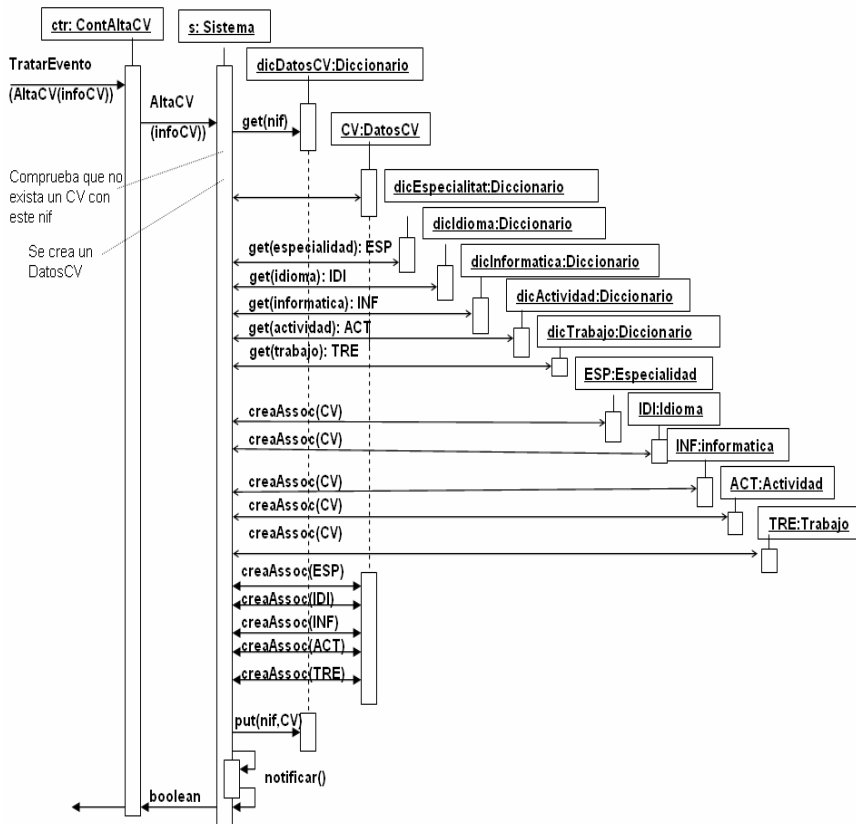
Aplicaremos también los patrones de diseño Diccionario e Iterador para acceder a los objetos almacenado en el sistema. El *Patrón Iterador* aparece con la necesidad de hacer recorridos secuenciales de los elementos de un objeto, es decir, para recorrer listas de objetos que tenemos en nuestro sistema. El *Patrón Diccionario* aparece por la necesidad de acceder a los objetos del sistema de una manera directa y sencilla. El diccionario tiene 4 operaciones básicas, obtener (get), guardar (put), obtener todas las instancias (elements) y eliminar (remove).

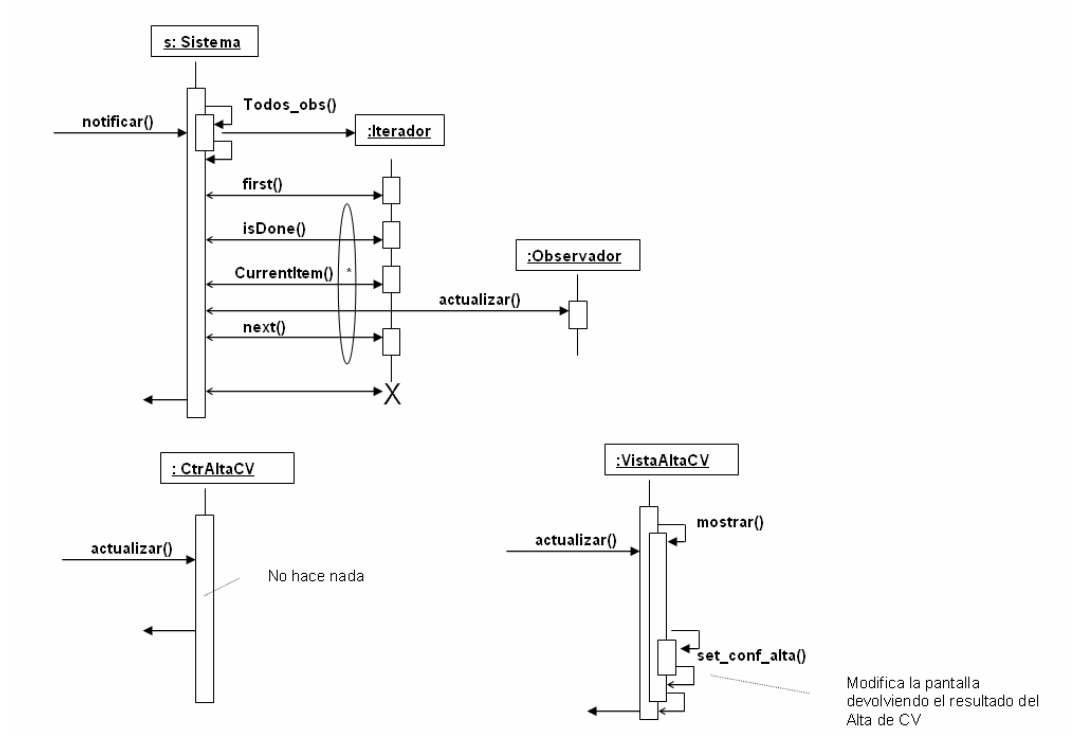
No vamos a mostrar los diagramas de secuencia de cada una de las operaciones. Como muestra tendremos el diagrama de inicialización del Patrón MVC y el diagrama de secuencia de la operación AltaCV.

### Inicialización del MVC



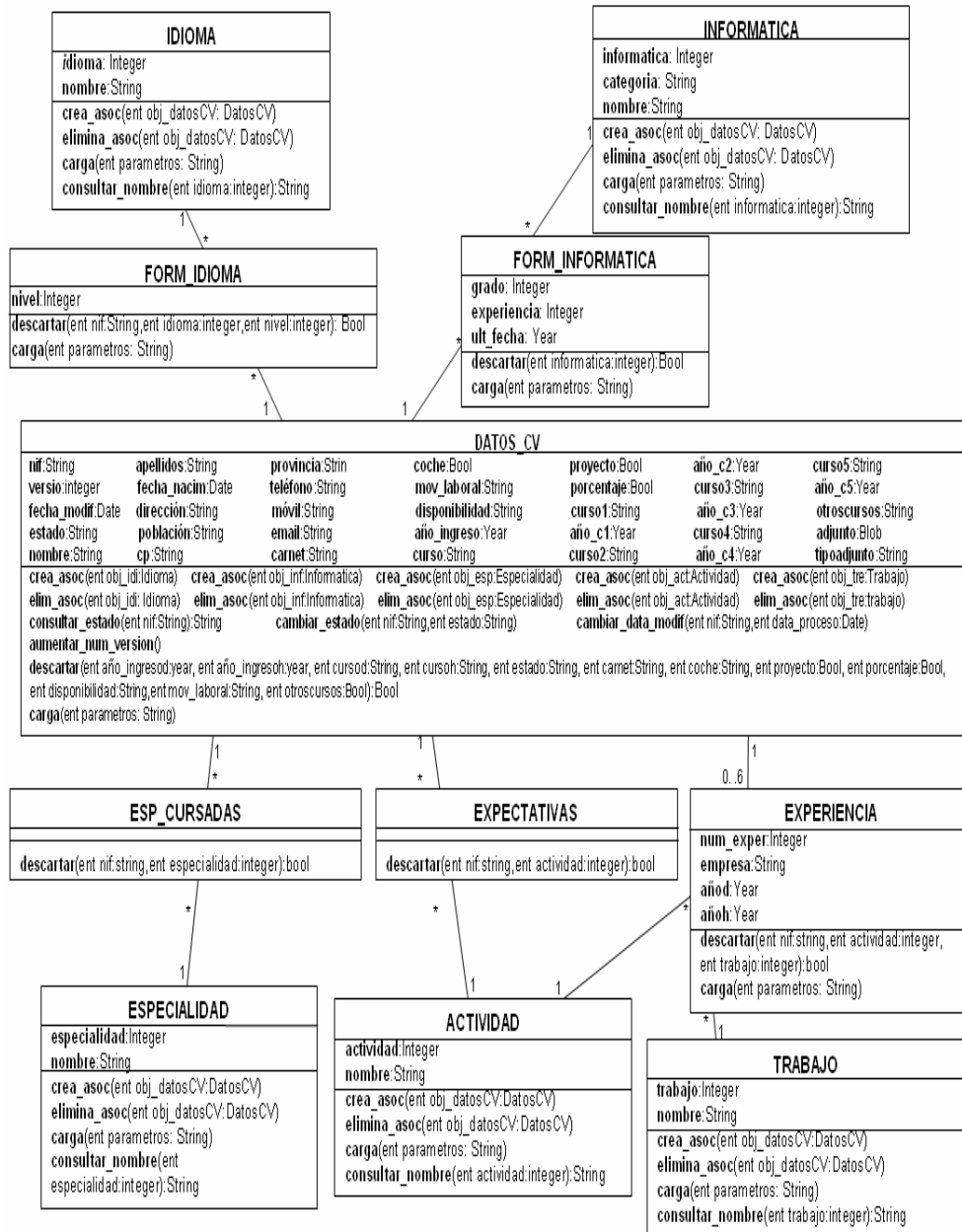
### Gestión de AltaCV





### 6.3. Diagrama de clases de diseño

A continuación mostraremos el diagrama de clases resultante del diseño, con la signatura de todas las operaciones.

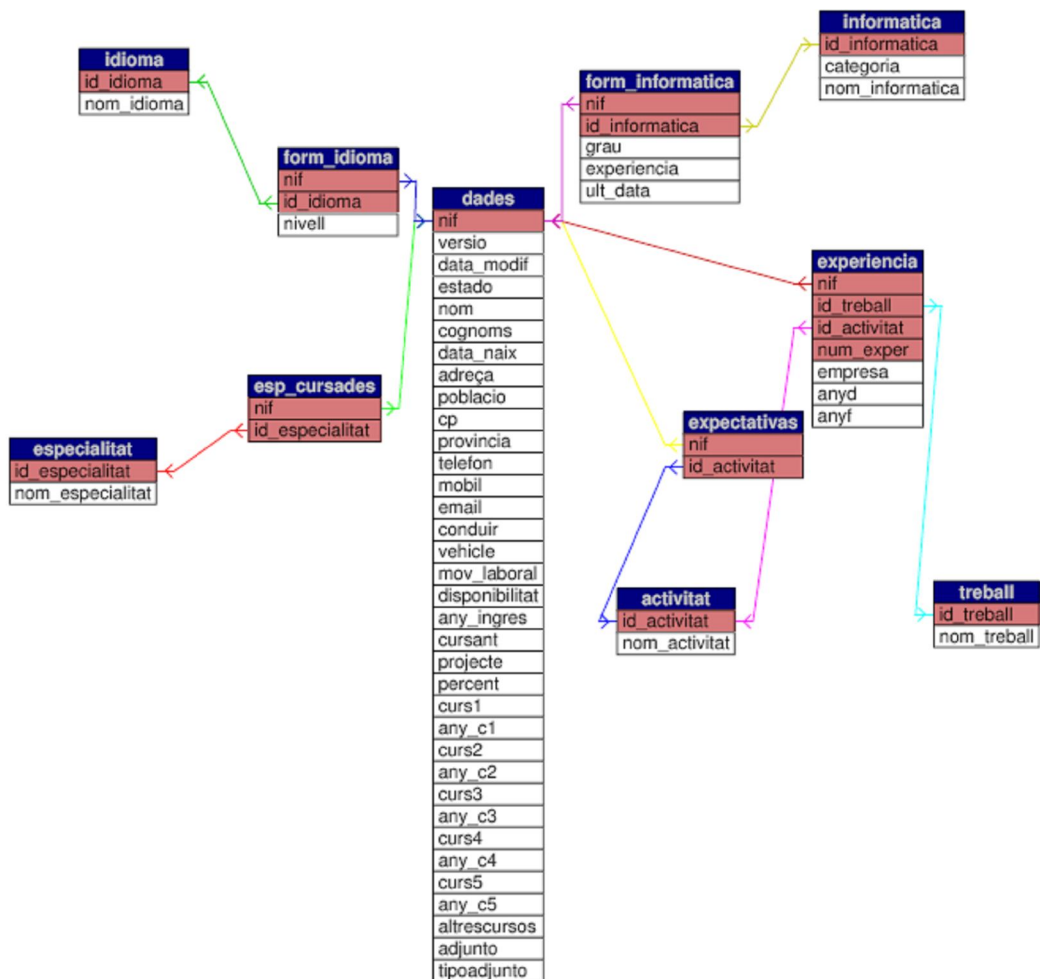


## 6.4. Diseño del Modelo de Datos

El diseño del modelo de datos (diseño de la base de datos) consiste en generar la estructura de tablas que se utilizarán para almacenar nuestros datos de forma permanente.

A continuación mostramos el esquema de la base de datos con todas las relaciones (diseño lógico de la base de datos), así mismo mostraremos las sentencias SQL para la creación de todas las tablas y las relaciones. Tanto el esquema como las sentencias SQL se han realizado con la herramienta phpMyAdmin que se encuentra dentro del paquete Xampp, el funcionamiento del cual explicamos en el apartado de anexos.

### DISEÑO LÓGICO DE LA BASE DE DATOS



## SENTENCIAS SQL CREACIÓN TABLAS Y RELACIONES

BEGIN TRANSACTION

```
CREATE TABLE `activitat` (
  `id_activitat` int(10) unsigned NOT NULL auto_increment,
  `nom_activitat` text collate latin1_general_ci NOT NULL,
  PRIMARY KEY (`id_activitat`)
) ENGINE
```

```
CREATE TABLE `dades` (
  `nif` char(9) collate latin1_general_ci NOT NULL,
  `versio` int(10) unsigned NOT NULL,
  `data_modif` date NOT NULL,
  `estado` char(1) collate latin1_general_ci NOT NULL,
  `nom` varchar(20) collate latin1_general_ci NOT NULL,
  `cognoms` varchar(40) collate latin1_general_ci NOT NULL,
  `data_naix` date NOT NULL,
  `adreça` varchar(80) collate latin1_general_ci NOT NULL,
  `poblacio` varchar(40) collate latin1_general_ci NOT NULL,
  `cp` char(5) collate latin1_general_ci NOT NULL,
  `provincia` varchar(40) collate latin1_general_ci NOT NULL,
  `telefon` char(9) collate latin1_general_ci default NULL,
  `mobil` char(9) collate latin1_general_ci default NULL,
  `email` varchar(50) collate latin1_general_ci default NULL,
  `conduir` enum('y','n') collate latin1_general_ci NOT NULL,
  `vehicle` enum('y','n') collate latin1_general_ci NOT NULL,
  `mov_laboral` char(1) collate latin1_general_ci NOT NULL,
  `disponibilitat` char(1) collate latin1_general_ci NOT NULL,
  `any_ingres` year(4) NOT NULL,
  `cursant` char(1) collate latin1_general_ci NOT NULL,
  `projecte` enum('y','n') collate latin1_general_ci NOT NULL,
  `percent` enum('y','n') collate latin1_general_ci NOT NULL,
  `curs1` varchar(60) collate latin1_general_ci default NULL,
  `any_c1` year(4) default NULL,
  `curs2` varchar(60) collate latin1_general_ci default NULL,
  `any_c2` year(4) default NULL,
  `curs3` varchar(60) collate latin1_general_ci default NULL,
  `any_c3` year(4) default NULL,
  `curs4` varchar(60) collate latin1_general_ci default NULL,
  `any_c4` year(4) default NULL,
  `curs5` varchar(60) collate latin1_general_ci default NULL,
  `any_c5` year(4) default NULL,
  `altrescursos` varchar(500) collate latin1_general_ci default NULL,
  `adjunto` mediumblob,
  `tipoadjunt` varchar(50) collate latin1_general_ci default NULL,
  PRIMARY KEY (`nif`)
)
```

```
CREATE TABLE `esp_cursades` (
  `nif` char(9) collate latin1_general_ci NOT NULL,
  `id_especialitat` int(10) unsigned NOT NULL,
  PRIMARY KEY (`nif`,`id_especialitat`)
)
```



```
CREATE TABLE `especialitat` (  
  `id_especialitat` int(10) unsigned NOT NULL auto_increment,  
  `nom_especialitat` varchar(100) collate latin1_general_ci NOT NULL,  
  PRIMARY KEY (`id_especialitat`)  
)  
  
CREATE TABLE `expectativas` (  
  `nif` char(9) collate latin1_general_ci NOT NULL,  
  `id_activitat` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`nif`,`id_activitat`)  
)  
  
CREATE TABLE `experiencia` (  
  `nif` char(9) collate latin1_general_ci NOT NULL,  
  `id_treball` int(10) unsigned NOT NULL,  
  `id_activitat` int(10) unsigned NOT NULL,  
  `num_exper` int(1) NOT NULL COMMENT 'no puede ser mayor que 6',  
  `empresa` varchar(60) collate latin1_general_ci NOT NULL,  
  `anyd` year(4) NOT NULL,  
  `anyf` year(4) NOT NULL,  
  PRIMARY KEY (`nif`,`id_treball`,`id_activitat`,`num_exper`)  
)  
  
CREATE TABLE `form_idioma` (  
  `nif` char(9) collate latin1_general_ci NOT NULL,  
  `id_idioma` int(10) unsigned NOT NULL auto_increment,  
  `nivell` int(1) NOT NULL,  
  PRIMARY KEY (`nif`,`id_idioma`)  
)  
  
CREATE TABLE `form_informatica` (  
  `nif` char(9) collate latin1_general_ci NOT NULL,  
  `id_informatica` int(10) unsigned NOT NULL,  
  `grau` int(2) default NULL,  
  `experiencia` int(3) default NULL,  
  `ult_data` year(4) default NULL,  
  PRIMARY KEY (`nif`,`id_informatica`)  
)  
  
CREATE TABLE `idioma` (  
  `id_idioma` int(10) unsigned NOT NULL auto_increment,  
  `nom_idioma` varchar(30) collate latin1_general_ci NOT NULL,  
  PRIMARY KEY (`id_idioma`)  
)  
  
CREATE TABLE `informatica` (  
  `id_informatica` int(10) unsigned NOT NULL auto_increment,  
  `categoria` varchar(50) collate latin1_general_ci NOT NULL,  
  `nom_informatica` varchar(60) collate latin1_general_ci NOT NULL,  
  PRIMARY KEY (`id_informatica`)  
)  
  
CREATE TABLE `treball` (  
  `id_treball` int(10) unsigned NOT NULL auto_increment,  
  `nom_treball` varchar(80) collate latin1_general_ci NOT NULL,  
  PRIMARY KEY (`id_treball`)  
)  
  
COMMIT TRANSACTION
```

## 6.5. Diseño externo de la capa de presentación. Diseño de formularios

El diseño externo de la capa de presentación define la interacción del usuario con el sistema software. Tiene como finalidad diseñar los elementos (tangibles) que el usuario ve, oye y toca al interactuar con el sistema.

En el diseño externo se definen tanto los **mecanismos de interacción** con los que el usuario pide los servicios al sistema, como los **mecanismos de presentación de la información** con los que se muestran al usuario los resultados de sus peticiones.

El equipo de diseñadores ha de incluir expertos en diversas áreas, desde el usuario final, a los programadores, psicólogos y diseñadores gráficos. Sin embargo, en muchos proyectos no se cuenta con este tipo de diseñadores y el diseño de la interfaz de usuario recae en los informáticos, como es el caso.

Los principios básicos para el diseño de la interacción con el sistema y la presentación de la información son:

- Integridad conceptual en la interacción (secuencia de acciones, abreviaturas,...) y en la presentación (formatos, colores, terminología...)
- Mínimo número de acciones a realizar (selección en vez de introducción de datos, evitar introducción de datos redundantes,...)
- Memoria a corto plazo del usuario mínima (no hace falta recordar códigos complejos, información de otras pantallas...)
- Personalización del mecanismo de interacción (experto, novato...) y de la presentación de la información (formato,...)
- Dar respuesta a cada petición de usuario y mensajes de error claros y explicativos.

El diseño externo realizado en este proyecto es un prototipo previo al que se implantará realmente. Al utilizar un navegador Web como cliente, permite que

el trabajo que se realiza en este prototipo se pueda aprovechar a la hora de la implementación final.

A continuación se detallan la navegabilidad, la estructura de las pantallas y la ampliación de los datos presentados en los formularios.

## NAVEGABILIDAD DE LA APLICACIÓN

Todas las acciones pasan dentro de la misma ventana permitiendo avanzar y retroceder con los botones del navegador en caso necesario. Las únicas acciones que abren una ventana nueva son las ventanas de confirmación y de aviso de errores de validación.

## ESTRUCTURA DE LAS PANTALLAS

Todas las pantallas siguen la misma estructura. Están divididas en un menú, una cabecera y una pantalla principal.

El menú se muestra en la parte izquierda de la pantalla y es un frame independiente donde el usuario puede escoger todos los servicios implementados para su perfil. La cabecera también es un frame independiente.

Una vez escogido un servicio, en la pantalla principal se muestra el formulario donde se rellenan los datos necesarios para realizar dicho servicio. En la parte superior se encuentra la cabecera que le corresponde al formulario, que muestra accesos **directos** a cada sección del formulario. Así mismo en el formulario se encuentran enlaces para ir moviéndose **secuencialmente** dentro de sus distintas secciones. En todo el transcurso de un servicio tanto la cabecera correspondiente como el menú siempre serán visibles.

La estructura específica de todas las pantallas se detalla mejor en el Manual de usuario, que puede consultarse en el CD que se adjunta con la memoria.

A continuación se muestra la pantalla alta de CV:

[AJUNTAR ARCHIU](#) [DADES PERSONALS](#) [FORMACIÓ](#) [IDIOMES](#) [INFORMÀTICA](#) [EXPERIÈNCIA PROFESSIONAL](#) [EXPECTATIVES PROFESSIONALS](#) [ALTA CURRICULUM](#)

[Tornar](#)

- [Alta curriculum](#)
- [Modificació curriculum](#)
- [Consultar el meu curriculum](#)
- [Baixa curriculum](#)

**1. ADJUNTAR ARCHIVO**

Ajuntar curriculum i/o carta de presentació:

[Seguir](#)

**2. DADES PERSONALS**

Nom :

Cognoms :

NIF :

Data naixement (aaaa-mm-dd) :

Adreça :

Població :

C.P. :

Província :

Telèfon :

Mòbil :

Correu electrònic :

## AMPLIACIÓN DE LOS DATOS DE LOS FORMULARIOS

Todos los formularios del sistema son dinámicos, de manera que los datos que se muestran en las listas desplegables y tablas provienen de la base de datos. Para modificar, añadir o eliminar la información que se muestra en los formularios se ha realizado una interfaz para el usuario administrador con la cual se aísla el acceso a la base de datos manteniéndose la integridad y la coherencia de los datos.

A continuación se muestra la pantalla de inserción de una especialidad nueva en la base de datos:

The screenshot shows a web interface for adding a new specialty. The page has a light blue background. On the left side, there is a navigation menu with the following items:

- [Tornar](#)
- Operacions sobre formularis:
  - [Alta especialitat](#)
  - [Alta idioma](#)
  - [Alta coneixement informàtic](#)
  - [Alta activitat econòmica](#)
  - [Alta tipus treball](#)
  - [Modificar especialitat](#)
  - [Modificar idioma](#)
  - [Modificar coneixement informàtic](#)
  - [Modificar activitat econòmica](#)
  - [Modificar tipus treball](#)
  - [Esborrar especialitat](#)
  - [Esborrar idioma](#)
  - [Esborrar coneixement informàtic](#)
  - [Esborrar activitat econòmica](#)
  - [Esborrar tipus treball](#)
- Operacions sobre currículums:
  - [Cerca curriculum](#)
  - [Esborrar curriculum](#)
- Estadístiques:
  - [Resum](#)

The main content area on the right is titled 'ALTA DE NOVA ESPECIALITAT'. It contains a single text input field labeled 'Nom especialitat:' and a button labeled 'Enviar'.

## **7. IMPLEMENTACIÓN**

En la fase de implementación se crea el código correspondiente al resultado de la fase de diseño, siguiendo los patrones y la arquitectura escogida además de los requerimientos no funcionales.

En este apartado de la memoria se resumirán los puntos principales de la implementación sin necesidad de ver el código.

### **7.1. Tecnología y lenguajes utilizados**

#### **7.1.1. Apache**

Apache es un servidor HTTP de código abierto disponible para diferentes tipos de plataformas. Entre las cuales están UNIX y WINDOWS. Comenzó su desarrollo en 1995 y principalmente se basaba en el código popular NCSA HTTPd, más tarde fue modificado por completo.

Tiene capacidad para servir páginas tanto de contenido estático, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información.

El Apache http Server está codificado y mantenido por desarrolladores de la comunidad Open Source y supervisado por Apache Software Foundation.

#### **7.1.2. Lenguajes HTML y CSS , PHP y JavaScript**

##### **HTML y CSS**

HTML acrónimo de *HiperText Markup Lenguaje* es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

Gracias a Internet y a los navegadores del tipo Internet Explorer, Opera, Firefox o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

CSS son las siglas de *Cascading Style Sheets*. Las hojas de estilo en cascada son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML.

El modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial, la forma de presentación que le aplicaremos a una Web entera, a un documento HTML, a una porción del documento o a una etiqueta en concreto. Con CSS conseguimos separar la capa de código del documento de su presentación lo que proporciona potencia a la programación

En este proyecto se implementan unas hojas de estilo muy simples, que se aplican solo sobre algunas etiquetas en concreto. La presentación de los documentos HTML usados en este proyecto no es uno de los objetivos principales del mismo, ya que debe poder adaptarse a la presentación, ya definida o no, del sistema donde el módulo se integre finalmente.

## **PHP**

PHP *'Hypertext Preprocessor'* es un lenguaje de programación interpretado de alto nivel, multiplataforma, cuyos programas se ejecutan en la parte del servidor, y cuyo código se escribe incrustado con el código HTML.

Dado que es multiplataforma funciona tanto para Unix, Windows y MAC OS, de forma que el código desarrollado se puede ejecutar en cualquiera de ellos sin modificaciones.

Al ejecutarse en el servidor, será necesario pues, que tengamos un servidor Web para poder comprobar nuestros programas. La opción que se tomó en la generación del proyecto fue Apache.

En ningún caso se envía código PHP al navegador, por lo que todas las operaciones realizadas son transparentes para el usuario, al que le parecerá que está visitando páginas HTML que cualquier navegador puede interpretar.

PHP tiene muchas ventajas, entre ellas:

- Software libre, que usa la licencia GPL (GNU Licencia Publica General, que regula los derechos de autor de los programas de código abierto) lo que facilita su distribución y uso
- Muy sencillo de aprender. Similar en sintaxis a C y a PERL
- Se puede incrustar código PHP con etiquetas HTML.
- Excelente soporte de acceso a base de datos.
- Toda la comprobación de la información enviada por el cliente se hace en el servidor.
- PHP proporciona unos tiempos de ejecución óptimos

Los principales usos del PHP son:

- Programación de páginas Web dinámicas en servidores, habitualmente en combinación con MySQL, aunque cuenta con un soporte para muchas otras bases de datos
- Creación de aplicaciones gráficas ‘stand-alone’ que no usan navegador web

## **JAVASCRIPT**

JavaScript es un lenguaje de programación interpretado que permite la ejecución de código dentro de las páginas HTML con una sintaxis semejante a la del Lenguaje Java. A diferencia de PHP y al igual que HTML, todo el código JavaScript se ejecuta en el cliente.

Gracias a los programas escritos en este lenguaje y embebidos en las páginas HTML, se pueden conseguir interesantes efectos en las páginas Web, como comprobar la validez de la entrada de formularios, abrir y cerrar ventanas, cambiar dinámicamente el aspecto y los contenidos de una página, cálculos



matemáticos sencillos y mucho más sin necesidad de realizar ninguna transmisión de datos por la red.

En nuestra aplicación todo el código javascript utilizado es completamente compatible en todos los navegadores.

### **7.1.3. MySQL**

MySQL es un gestor de bases de datos de código abierto, lo que quiere decir que es accesible para cualquiera, para usarlo o modificarlo. Podemos descargar MySQL desde Internet y usarlo sin ningún coste, de ese modo cualquiera puede inclinarse a estudiar el código fuente y modificarlo en base a sus necesidades. MySQL usa el GPL para definir qué podemos y no podemos hacer con el software en diferentes situaciones.

MySQL es muy rápido, fiable, robusto, multihebra, multiusuario, multiproceso, fácil de usar, con capacidad para manejar tanto grandes volúmenes de datos como pequeños, disponible para varias plataformas, acceso seguro mediante privilegios y password, soporta ODBC, basado en ANSI SQL 92, arquitectura cliente/servidor, etc.

MySQL es un Sistema de Gestión de Base de Datos Relacional. El modelo relacional se caracteriza, a muy grandes rasgos, por disponer de toda la información contenida en tablas, y las relaciones entre datos deben ser representadas explícitamente en esos mismos datos. Esto añade velocidad y flexibilidad.

Su sencillez de instalación y su gran compatibilidad con lenguajes de programación de servidores Web como PHP, hacen de esta herramienta el soporte perfecto a páginas web dinámicas. De hecho, en los últimos tiempos han surgido paquetes instaladores que montan, sobre cualquier plataforma, un servidor Apache, PHP y MySQL sin dificultad. En este proyecto se ha usado uno de estos paquetes, el paquete XAMPP.

## **7.2. Exportación a PDF**

El formato PDF aporta un alto grado de encapsulación y versatilidad. La gran desventaja que existe es que para crear archivos PDF se necesita el programa Adobe Acrobat, que no es un software libre. Pero la problemática es mayor ya que en un principio parece imposible crear PDF's on-line.

Como crear ficheros PDF on-line, es la función de la librería libpdf. Con ella podremos crear ficheros PDF y enviarlos, si necesidad de poseer el programa Adobe Acrobat. Los datos que se incrustan en un fichero PDF no pueden ser modificados lo que ofrece un alto grado de seguridad.

## **7.3. Seguridad**

Hay ocasiones en las que se hace necesario recibir/enviar información sensible desde/a un servidor de Web. Resulta evidente que no interesa que esa información sea accesible a toda la Red, sino sólo a un pequeño número autorizado de usuarios o llegue solo al servidor. Por lo tanto, no vale con restringir el acceso mediante claves de acceso o procedimientos similares, además la información que viaja hacia esos usuarios o al servidor debe ir cifrada.

En conclusión es necesario asegurar mediante algún mecanismo la intimidad y la integridad en las sesiones con el servidor Web. En el proyecto solo se implementa el módulo de gestión de los CV pero explicaremos que habría que tener en cuenta para gestionar la seguridad.

### 7.3.1. Autenticación y privilegios

Para acceder a las partes restringidas del sistema haría falta que el usuario se identificase mediante un nombre de usuario y una contraseña. Se necesitaría que en el servidor se guardasen los datos de todos los usuarios y los privilegios que éste tiene.

En el sistema se añadiría una capa de seguridad propia en forma de cookies y basada en un algoritmo de clave privada y clave pública.

Al final de la validación el usuario tendría una cookie con un valor que se podría interpretar como la clave pública del servidor además de una lista de privilegios o acciones que podría realizar y por el otro lado el servidor tendría guardada la clave privada en una variable de sesión.

La lista de privilegios se obtendría de la lista de privilegios del perfil del usuario. Esta lista también se guardaría a nivel de sesión, lo que permitiría saber en cualquier momento las acciones que podría realizar el usuario.

A partir de este momento cada vez que el usuario accediese a una zona restringida, se validarían las claves además de comprobarse que el usuario tiene privilegio para realizar la acción.

La variable de sesión caducaría pasado un tiempo sin actividad por parte del usuario, si una vez pasado este tiempo de *sesión* y el usuario volviese a hacer algún acceso a una zona restringida, el sistema respondería que la sesión ha caducado y que es necesario volver a identificarse.

Para garantizar que todos los datos que intercambian los usuarios con el servidor son confidenciales se puede poner bajo SSL (https) todo el portal. Así si se rompiera la seguridad SSL, aún quedaría la seguridad añadida por el sistema.

### 7.3.2. CSS y HTTPS

**SSL** *Secure Sockets Layer* es un protocolo criptográfico que proporciona comunicaciones seguras (autenticación y privacidad de la información entre extremos) en Internet. Habitualmente, sólo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar; la autenticación mutua requiere un despliegue de infraestructura de claves públicas *PKI* para los clientes. CSS permite a las aplicaciones cliente-servidor comunicarse de una forma diseñada para prevenir escuchas, la falsificación de la identidad del remitente y mantener la integridad del mensaje

El protocolo **HTTPS** es la versión segura del protocolo HTTP. El sistema HTTPS utiliza un cifrado basado en las SSL para crear un canal cifrado más apropiado para el tráfico de información sensible que el protocolo HTTP. Cabe mencionar que el uso del protocolo HTTPS no impide que se pueda utilizar HTTP. Es aquí, cuando nuestro navegador nos advertirá sobre la carga de elementos no seguros (HTTP), estando conectados a un entorno seguro (HTTPS). Los protocolos HTTPS son utilizados por navegadores como: Internet Explorer, Mozilla Firefox, Opera, ... entre otros. Es utilizado para asegurar páginas Web principalmente entidades bancarias, tiendas en línea, y cualquier tipo de servicio que requiera el envío de datos personales o contraseñas.

Para poder usar el protocolo HTTPS en nuestro sistema deberemos hacer los siguientes pasos:

- Comprobar que nuestro servidor tenga instalado SSL
- Obtener un certificado digital. Decidirnos por una autoridad de certificación (AC). Entre ellas se encuentran AT&T, GTE, MCI, Thawte o VeriSign.

El proceso a seguir obtener un certificado digital de VeriSign sería:

- Conectarse con la página web de VeriSign para la solicitud de certificados, donde se dan instrucciones paso a paso de lo que se debe hacer, previo registro.
- Generar una Solicitud de Firma de Certificado. Una Solicitud de Firma de Certificado (CSR) es un fichero que se debe enviar a la autoridad de certificación para que lo firme y lo devuelva en la forma de un certificado. Básicamente, contiene la clave pública que debe ser firmada por la AC una vez haya verificado la veracidad de los datos
- Envía la Solicitud de Firma de Certificado a VeriSign
- La autenticación tarda algunos días mientras VeriSign comprueba los datos. Si toda la información que enviaste es correcta, el recibirá el Certificado de Servidor Seguro en unos pocos días.
- Instalar el Certificado de Servidor

## 8. PRUEBAS

La corrección de un programa es un aspecto básico e indispensable. Las implicaciones de un fallo de software pueden ser fatales. Por lo tanto es necesario diseñar y ejecutar juegos de prueba que, de forma sistemática, detecten diferentes tipos de error en el menor tiempo y esfuerzo posibles.

Los tipos de juegos de prueba realizados son:

- Pruebas unitarias: prueba por cada componente elemental
- Pruebas de integración: prueba de las interacciones entre componentes
- Pruebas del sistema: prueba global del sistema como una unidad de ejecución
- Pruebas de validación: para asegurar que se satisfacen los requerimientos desde el punto de vista del usuario
- Pruebas de regresión: para asegurar que cambios realizados no introducen nuevos errores en componentes ya probados

Se han realizado estos juegos para las capas de la arquitectura MVC

### TESTS CAPA VISTA

Para la capa Vista se hicieron las pruebas con datos reales para:

- validar la corrección de los datos introducidos
- validar la corrección de la longitud de los campos de la interfaz gráfica
- validar la corrección de todas las acciones
- validar la corrección de los resultados esperados

### TESTS CAPA CONTROLADOR

Para la capa Controlador se hicieron pruebas para comprobar que los datos que recibe la capa de vista son correctos. Estas pruebas son importantes porque es necesario comprobar que las acciones son correctas sin la implementación previa de la capa de vista.

## TEST CAPA MODELO

Para la capa Modelo se comprobó que para cada operación se devolvían los datos adecuados así como que la inserción, modificación, consulta y eliminación de los datos en la base de datos se realizaba correctamente.

## 9. INSTALACIÓN Y MANTENIMIENTO

Para el funcionamiento de la herramienta serán necesarios varios elementos:

- servidor Web que admita páginas PHP (Apache)
- servidor de MYSQL.

Todo esto se encuentra en el paquete XAMPP que trae ya pre-configurado APACHE, PHP, MySQL, phpMyAdmin para la familia Windows (ver anexos).

Una vez tengamos estos elementos debemos realizar los pasos siguientes:

- Instalar la aplicación copiando en el servidor todos los ficheros proporcionados en el CD ajunto a la memoria, en un directorio válido para apache como C:\apachefriends\xampp\htdocs. No debe modificarse la estructura interna, el re de los archivos y de directorios no puede por tanto modificarse. Los directorios son:
  - JS: Se almacenan los diferentes scripts en lenguaje javascript que se encuentran enlazados con las páginas php.
  - COMP: En él se almacenarían las páginas HTML utilizadas así como los archivos php, que dan forma a la aplicación.
  - IMG: Se almacenan todas las imágenes necesarias para la aplicación
  - El fichero principal se llama 'index.html' y accediendo a él entramos en la aplicación web. Se debe encontrar en el directorio raíz
- Instalar la base de datos.

Mediante phpMyAdmin crearemos la base de datos con nombre 'curriculum'. En el directorio SQL se encuentran los archivos sql necesarios para la creación y carga de las tablas de la BD. Se tiene que usar la función importar del phpMyAdmin (ver anexos).



## 10. CONCLUSIÓN Y FUTURO DE LA APLICACIÓN

En este apartado se evalúan los resultados del proyecto en varios aspectos. Mostraremos una atención principal a los objetivos cubiertos y a futuras ampliaciones.

### OBJETIVOS CUBIERTOS

Los objetivos de este proyecto se han alcanzado con bastante éxito. Se han cubierto en todas sus etapas las funcionalidades descritas en el análisis de requerimientos, a pesar de que la planificación temporal real haya tenido una desviación de un 17,5% respecto de la planificación inicial, debido al aprendizaje de la tecnología utilizada y de los problemas surgidos durante la implementación.

La aplicación diseñada está preparada para poder ser ampliada, modificada y personalizada fácilmente, gracias a la modularidad aplicada a las capas y servicios.

### FUTURO DE LA APLICACIÓN

El futuro de la aplicación se centraría en las ampliaciones que se lleven a cabo. Las más destacables serían:

- Ampliación de los datos. Dentro de los CV se podría tener la necesidad de guardar más datos. Por ejemplo dentro de los conocimientos del candidato sólo se han tenido en cuenta los idiomas y los conocimientos de informática. Estos datos podrían ampliarse a otros campos.
- Generalización o especificación de los datos. La aplicación puede gestionar desde curriculums muy generales hasta los más específicos. Se podría gestionar la información de cualquiera de las categorías profesionales (por ejemplo ingenieros industriales, médicos, abogados etc...) pudiéndose configurar los formularios de entrada de datos según unos criterios previos escogidos por el usuario.
- Ampliación de las estadísticas del sistema.
- Posibilidad de escoger el idioma de la aplicación.

## 11.BIBLIOGRAFIA

- Dolors Costal - M.Ribera Sancho – Ernest Teniente  
**“Enginyeria del software: Especificació”**  
*Edicions UPC, Febrero 2000*
- Cristina Gómez – Enric Mayol – Antoni Olivé – Ernest Teniente  
**“Enginyeria del software: Disseny I”**  
*Edicions UPC, Septiembre 2003*
- E.Rodríguez – T.Urpí  
**“Fitxers i Bases de Dades (FBD) Transparències de l’assignatura”**  
*Febrero 2001*
- Nuria Altés  
**“Redacció Tècnica i de Projectes a l’Enginyeria. Dossier de l’assignatura”**  
*Febrero 2005*
- Swing Sæther Bakken, Alexander Aulbach, Egon Schimd, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Zeev Suraski, Andrei Zmievski y Jouni Ahto  
**“Manual de PHP”**  
*Este manual es © Copyright 1997, 1998, 1999, 2000, 2001 del Grupo de documentación de PHP. Este manual puede ser redistribuido bajo los términos de la "GNU General Public License" publicada por la "Free Software Foundation"; tanto bajo la versión 2 de esta licencia o bajo versiones posteriores.*
- Fermín Vilà  
**“Manual Práctico JavaScript”**  
*2004*

- Cristian Cantón Ferrer  
**“Curso PHP-SQL. Aplicaciones dinámicas web con conexiones a bases de datos. Dossier del curso”**  
*IAESTE TELECOMUNICACIONES, Marzo-Abril 2001*
- Manuel Puig Sánchez  
**“Desarrollo de módulo de gestión comercial en Web”**  
*PFC d'Enginyeria Informàtica. Facultat d'Informàtica de Barcelona.*  
*Gener 2006*

## 11.1. Bibliografía digital

### ESTUDIO PREVIO

- <http://www.infojobs.net>
- <http://www.laboris.net>
- <http://www.trabajofacil.com>
- <http://www.hispavista.com>
- <http://www.bolsadetrabajo.com>
- <http://www.monster.es>
- <http://www.oficinaempleo.com>
- <http://www.adecco.es>
- <http://www.primerempleo.es>
- <http://www.tecnoempleo.com>
- <http://www.tecnjobs.net>
- <http://www.cooiar.net>
- <http://www.eic.es>
- <http://www.fib.es>
- <http://www.educoweb.com/trabajo>

### PHP

- <http://www.php.net>
- <http://www.webestilo.com/php>
- <http://www.desarrolloweb.com>
- <http://www.forosdelweb.com>
- <http://www.webtaller.com/foros>
- <http://www.soleup.eup.uva.es/mediawiki/index.php>
- <http://www.programación.net/php>
- <http://www.php-hispano.net/foros/php>
- <http://www.programacion.com/foros>
- <http://www.geneura.ugr.es/~maribel/php/temario.html>

- <http://www.webexperto.com/articulos/cat/11/php>
- <http://www.gembeta.com>

## XAMPP

- <http://www.apachefriend.org>

## PROCESOS DE INGENIERIA DEL SOFTWARE

- <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>
- <http://www.monografias.com/trabajos5/desof>
- <http://www.microsoft.com/spanish/MSDN/estudiantes/infsoft>

## ARQUITECTURA CLIENTE/SERVIDOR

- <http://www.csi.es>

## SSL

- <http://www.verisign.es>

## PHPMYAdmin

- <http://www.sync.es>
- <http://www.phpmyadmin.net>

## PDFLIB

- <http://www.pdflib.com>

## JAVASCRIPT

- <http://www.wikilearning.com>

## OTRAS FUENTES

- <http://www.wikipedia.es>

## 12. ANEXOS

### 12.1. Programas utilizados

#### 12.1.1. Paquete XAMPP

**XAMPP** es un paquete que permite instalar varios tipos de servidores en un sistema fácil y rápidamente.



XAMPP incluye el servidor web Apache, los servidores de bases de datos MySQL y SQLite, sus respectivos gestores phpMyAdmin y phpSQLiteAdmin, el intérprete del lenguaje homónimo PHP con los extras incluidos en PEAR, el intérprete del lenguaje Perl, servidores de FTP como ProFTPD ó FileZilla FTP Server, las estadísticas Webalizer y OpenSSL, eAccelerator, Freetype2, libjpeg, libpng, zlib, Ming, etc. entre otros.

Existen versiones para Linux, Windows, MacOS X y Solaris, cuyos paquetes difieren según la disponibilidad de los diversos programas en cada plataforma. En este proyecto instalamos Xampp para Windows.

#### **XAMPP para Windows** (*versión 1.4.15*)

Contiene: Apache HTTPD 2.0.54, MySQL 4.1.13, PHP 5.0.4 + 4.4.0 + PEAR + Switch, MiniPerl 5.8.7, mod\_ssl 2.0.54, Openssl 0.9.8, PHPMYAdmin 2.6.3 pl1, XAMPP Control Panel 2.1, eAccelerator 0.9.4, Webalizer 2.01-10, Mercury Mail Transport System für Win32 und NetWare Systems v4.01a, FileZilla FTP Server 0.9.8c, SQLite 2.8.15, ADODB 4.65, Zend Optimizer 2.5.10a y XAMPP Security.

Se puede el paquete Xampp para Windows en Internet gratuitamente en <http://www.apachefriends.org/en/xampp-windows.html>, donde también se encuentran las instrucciones necesarias para la instalación y configuración.

### **12.1.2. Editplus**

EditPlus (antes conocido como Edit++) es un editor de texto 32-bit para Windows (Win95/98/NT/ME/2000/XP) . No sólo es un buen reemplazo para el bloc de notas de Windows, sino que además ofrece varias potentes funciones para autores y programadores de páginas web.

Algunas de las funciones más importantes es el resalto de errores de sintaxis, una barra de herramientas de HTML, herramientas definibles por el usuario, y archivos de ayuda; potente rehacer y deshacer, soporte para arrastrar y soltar, búsqueda y reemplazo de palabras, corrector ortográfico, teclas rápidas, y muchas cosas más, entre las que destaca las funciones de FTP incluidas en esta última versión y la posibilidad de ver páginas web.

EditPlus también incluye resalto de errores sintéticos para HTML, ASP, JavaScript, VBScript, Perl, Java, PHP, CSS, y C/C++. La versión de prueba es funcional durante un periodo de 30 días.

### **12.1.3. Gant Project**

Gantt Project es una herramienta OpenSource para la gestión de proyectos. Permite generar diagramas de Gantt, donde se enumeran las tareas, su duración y dependencias así como los recursos tanto materiales como humanos necesarios para realizar proyectos.

### **12.1.4. PhpMyAdmin**

PhpMyAdmin puede gestionar tanto un servidor MySQL completo como una única Bd.

Actualmente phpMyAdmin tiene las siguientes funcionalidades:

- mostrar y borrar bases de datos, tablas, vistas, campos e índice
- crear, copiar, borrar, renombrar y modificar bases de datos, tablas, campos e índices

- ejecutar, editar y guardar cualquier sentencia SQL
- cargar ficheros de texto en tablas
- exportar datos en varios formatos: CSV, XML, PDF, ISO/IEC 26300 - OpenDocument Text y Spreadsheet, Word, Excel y L<sup>A</sup>T<sub>E</sub>X
- administrar y mantener diversos servidores
- gestionar usuarios y privilegios MySql
- revisar la integridad referencial en tablas MyISAM
- crear gráficos PDF del esquema de la base de datos
- transformar datos almacenados en cualquier formato usando un conjunto de funciones predefinidas, como mostrar datos tipo BLOB como un imagen o un enlace de descarga

Existen en Internet muchas referencias sobre el uso de phpMyAdmin. Para este proyecto se ha utilizado el manual de uso de phpMyAdmin en formato PDF que se encuentra en <http://www.sync.es>, donde se pueden encontrar explicadas las funcionalidades más relevantes de la aplicación.

## 12.2. Glosario

En este apartado se incluyen definiciones de conceptos y acrónimos usados en la memoria.

**BD - Base de datos.** Conjunto ordenado de datos que permite administrarlos, consultarlos, realizar informes sobre los mismos, etc. De este modo la información registrada en una sola ocasión se puede localizar y clasificar en sus más diferentes aspectos y de la forma más rápida posible.

**Cookie.** Pequeño archivo que se genera en el disco duro del usuario desde una página web. Un archivo de esta clase puede registrar las actividades del usuario en la página visitada. Su uso es controvertido, puesto que implica un registro de datos en el ordenador del usuario.

**FRAMES – Marcos.** Técnica del navegador, que permite dividir la ventana de visualización en zonas independientes entre sí, las cuales contienen diferentes documentos. De aspecto muy agradable, pero discutido porque en los navegadores más antiguos no es posible su visualización correcta.

**FTP – File Transfer Protocol.** *Protocolo de transferencia de archivos.* Protocolo que regula la transferencia de archivos vía Internet. Permite enviar y recibir archivos en línea. También se llaman así a los programas que transmiten y reciben archivos según el protocolo FTP.

**GNU –** Serie de programas desarrollada y comercializada por la Free Software Foundation. Estos programas están ampliamente divulgados entre los programadores UNIX.

**HTTP - Hypertext Transmisión Protocol.** *Protocolo de transmisión de hipertexto.* Protocolo que se utiliza para transmitir páginas web.

**PDF - Portable Document Format.** *Formato de Documento Portátil.* Es una forma de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.

**PFC – Proyecto Final de Carrera**

**SGBD – Sistema Gestor de Bases de Datos.** Son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan

**UML – Unified Modelling Language.** *Lenguaje Unificado de Modelado.* Es el lenguaje de modelado de sistemas más conocido en la actualidad.

**UP – Unified Process.** *Proceso Unificado.* Método iterativo de diseño de software que describe como desarrollar software de forma eficaz, utilizando técnicas probadas en la industria.



**XML – eXtensible Markup Language.** *Lenguaje de marcado ampliable o extensible.* Al igual que el HTML, se basa en documentos de texto plano en los que se utilizan etiquetas para delimitar los elementos de un documento.