



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Proyecto Fin de Carrera

SISTEMA DE SELECCIÓN DE PERSONAL BASADO EN MODELOS DE DECISIÓN MULTICRITERIO

Alumno: Pedro Barrio Muñoz

Tutores: Prof. D. Luis Martínez López
Prof. D. Pedro José Sánchez Sánchez

Dpto: Informática

Junio, 2013



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

D. Luis Martínez López y D. Pedro José Sánchez Sánchez pertenecientes al Departamento de Informática de la Universidad de Jaén.

INFORMAN

Que la memoria titulada “Sistema de Selección de Personal Basado en Modelos de Decisión Multicriterio” ha sido realizada por **D. Pedro Barrio Muñoz** con DNI 77338594-J bajo nuestra dirección y se presenta como memoria del Proyecto Fin de Carrera para optar al grado de Ingeniería en Informática.

Jaén, 19 de Junio de 2013

Vº Bº

Fdo.: **D. Luis Martínez López**

Fdo.: **D. Pedro J. Sánchez Sánchez**

Agradecimientos

No tendría espacio para dar las gracias a todas las personas que han formado parte de estos años de carrera, que culminan con que este proyecto sea una realidad. Nombraré especialmente a:

A mis padres y mi hermana, por soportar los ratos en los que todo esto se ha hecho cuesta arriba, y han estado a mi lado apoyándome.

A mis compañeros Alberto Montoro, María Cózar y Eugenio Martínez, por esas horas de estudio, por esos ratos por la universidad que hemos compartido, y por todos los quebraderos de cabeza que nos ha tocado pasar.

A esos amigos, que han tenido que aguantar muchos ratos de comentar dudas, y que han sacado tiempo para escucharlas y echar una mano.

A Pilar, por intentar comprender las cosas que le explicaba, y sobre todo, por sus ánimos, y por creer en todo momento en mí, aún cuando ni yo mismo creía que era capaz.

A Francisco J. Galiano, por también brindarme tiempo y darme su consejo sobre las páginas web.

A mis tutores Luis Martínez y Pedro J. Sánchez, por su gran paciencia con mi demora de las entregas, por las reuniones y por su apoyo para poder finalizar este proyecto.

Y para terminar, no podía dejar sin nombrar a mis abuelos, que aunque no lo vayan a leer, sin ellos no sería como soy.

GRACIAS A TODOS

Índice

CAPÍTULO 1: INTRODUCCIÓN	1
1.1 INTRODUCCIÓN AL PROYECTO	3
1.2 PROPÓSITO.....	5
1.3 OBJETIVOS.....	5
1.4 ESTRUCTURA.....	5
CAPÍTULO 2: SELECCIÓN DE PERSONAL Y TOMA DE DECISIONES MULTICRITERIO EN SELECCIÓN DE PERSONAL	9
2.1 DEFINICIÓN DE LOS RECURSOS HUMANOS EN LA EMPRESA Y ÁREAS QUE LO CONFORMAN.....	11
2.2 SELECCIÓN DE PERSONAL	15
2.3 TOMA DE DECISIONES.....	20
2.3.1 <i>Clasificación de los problemas de toma de decisiones</i>	21
2.3.2 <i>Modelado de las opiniones en la toma de decisiones</i>	22
2.3.3 <i>Esquema básico de resolución en problemas de Toma de Decisiones</i>	25
2.3.4 <i>Análisis de decisión y evaluación</i>	26
2.3.5 <i>Toma de Decisiones con Múltiples Criterios</i>	28
2.3.6 <i>TOPSIS</i>	29
2.3.7 <i>Modelo del baremo de Contratación</i>	35
CAPÍTULO 3: TECNOLOGÍAS USADAS EN EL SISTEMA.XML	37
3.1 QUÉ ES XML.....	39
3.2 VENTAJAS DEL XML	40
3.3 DIFERENCIAS CON HTML	41
3.4 XML BIEN FORMADO	42
3.5 ESQUEMA DE DEFINICIÓN DEL XML	44
3.6 USANDO UN FORMULARIO PARA RECOGER DATOS EN XML. INFOPATH.....	45
CAPÍTULO 4: DISEÑO, IMPLEMENTACIÓN Y PRUEBAS	49
4.1 ESPECIFICACIÓN DE LOS REQUERIMIENTOS.....	52
4.1.1 <i>Requerimientos funcionales</i>	54
4.1.2 <i>Requerimientos no funcionales</i>	57
4.2 ANÁLISIS DEL SISTEMA.....	58

4.2.1	<i>Perfil de usuario</i>	58
4.2.2	<i>Casos de uso</i>	60
4.2.3	<i>Casos de uso de nuestra aplicación</i>	64
4.2.4	<i>Escenarios</i>	76
4.3	DISEÑO.....	84
4.3.1	<i>Diagrama de clases</i>	84
4.3.2	<i>Diseño de los Datos</i>	90
4.3.3	<i>Diseño de la interfaz</i>	106
4.4	IMPLEMENTACIÓN.....	122
4.4.1	<i>Tipo de la arquitectura de la aplicación</i>	122
4.4.2	<i>Lenguajes de programación utilizados</i>	123
4.4.3	<i>Herramientas de desarrollo</i>	127
4.5	IMPLANTACIÓN Y PRUEBAS.....	127
4.5.1	<i>Pruebas y validación</i>	128
CAPÍTULO 5: CONCLUSIONES		143
5.1	CONCLUSIONES.....	145
5.2	FUTURAS MEJORAS	146
BIBLIOGRAFÍA		147
ANEXO I. MANUAL DE INSTALACIÓN		153
ANEXO II. MANUAL DE USUARIO		181
ANEXO III. ESQUEMA SEGUIDO PARA LA DEFINICIÓN DE LOS ELEMENTOS XML		221

Capítulo 1:

Introducción

1.1 Introducción al proyecto

Las razones por las que una empresa desea iniciar un proceso de selección de personal pueden ser diversas: se pretende cubrir un puesto de nueva creación, sustituir a un trabajador que se jubila, iniciar una nueva actividad, reemplazar un puesto que ha quedado vacante, etc.

La **selección de personal** es la parte del departamento de Recursos Humanos que se encarga de este fin, y se puede definir como [1]: *“Escoger entre los candidatos reclutados los más adecuados, para ocupar los cargos existentes en la empresa, tratando de mantener o aumentar la eficiencia y el desempeño del personal, así como la eficacia de la organización”*.

De esta manera, la selección busca solucionar dos problemas fundamentales:

- a. Adecuación del hombre al cargo
- b. Eficiencia del hombre en el cargo

También al elegir a la mejor persona posible intentaremos hacerlo sin ninguna discriminación injusta, en contra de ciertos grupos de solicitantes [2].

La selección debe mirarse como un proceso real de comparación entre dos variables: los requisitos del cargo (exigencias que debe cumplir el ocupante del cargo) y el perfil de las características de los candidatos que se presentan. La primera variable les suministra el análisis y la descripción del cargo; la segunda se obtiene mediante la aplicación de técnicas de selección.

Del llamado **análisis del puesto de trabajo** [3] se extrae la definición y características que debe de cumplir un candidato para adecuarse a la plaza. En nuestro caso, nos regimos por el documento [4] creado por el Vicerrectorado de Ordenación académica, Innovación docente y profesorado.

Al fin y al cabo, la decisión de que un aspirante sea o no elegido, no es más que un proceso de **toma de decisiones**, en la que prevalece el componente humano que viene

dado por el responsable último de la selección. Esto introduce una componente subjetiva siempre, ya que la última decisión de la selección, una vez que hemos evaluado al candidato en los distintos aspectos, forma parte del responsable del departamento de Recursos Humanos, tras la realización de la entrevista o entrevistas y de la demostración de los conocimientos y competencias que el candidato ha acreditado en el Currículum Vitae.

Para **facilitar** la toma de decisiones, podemos ver que pueden utilizarse **modelos multicriterio** (AHP, TOPSIS, ELECTRE), que tienen en cuenta varios criterios para evaluar, y producen buenos resultados [5] [6] en el área en la que se centra nuestro estudio.

En nuestro caso, analizaremos la situación actual del proceso de contratación de profesores sustitutos en la Universidad de Jaén. Actualmente se cuenta con un sistema¹ que recoge la información de las personas que desean acceder a la lista para cubrir bajas de PDI en la Universidad, pero que sigue dejando en manos del responsable del departamento el análisis y la evaluación de lo aportado.

Las **motivaciones** que llevan a la realización de este proyecto son proporcionar al experto en la selección una herramienta que permita obtener los datos, en un formato estandarizado, en nuestro caso XML [7], para ser independiente de la fuente de datos desde donde se recojan los perfiles de los aspirantes, y para facilitar la decisión, con la inclusión de un algoritmo multicriterio que dará una valoración más objetiva de qué candidato puede ser más idóneo para un puesto, según los requisitos especificados.

En el **desarrollo del sistema** tendremos en cuenta, en todas las fases, que va a poder ser utilizado para evaluar a candidatos, y puestos, adaptándose a las características de cada situación. Así, tanto los requisitos que se van a tener en cuenta como el modelo que van a ser utilizados, podrán ser cambiados para adaptarlos al proceso de selección.

¹ <http://bcontratacion.ujjaen.es/>

1.2 Propósito

La principal tarea del proyecto que se presenta es diseñar e implementar un sistema basado en un algoritmo de toma de decisión multicriterio, para la selección de PDI en la Universidad de Jaén.

1.3 Objetivos

Para alcanzar el propósito anteriormente descrito, habrá que conseguir los siguientes objetivos:

- Búsqueda de información bibliográfica sobre los distintos modelos de evaluación de personal.
- Identificación de los indicadores utilizados para la evaluación de un candidato a un perfil profesional.
- Análisis del problema y definición de funcionalidades.
- Definición un documento en formato XML para automatizar el proceso de recogida de información a partir de los perfiles de usuario ya existentes.
- Obtención de los datos a partir de un documento externo, pasando los datos a formato XML para poder trabajar cómodamente con ellos y que sean reconocibles por nuestro sistema.
- Desarrollo de un software que establezca un ranking con los candidatos en función de la valoración obtenida, utilizando un modelo de decisión multicriterio (TOPSIS).
- Documentación que permita la adecuación de este sistema de selección de personal a distintos ámbitos.

1.4 Estructura

La memoria de este proyecto se estructura en distintos capítulos que describen las distintas fases para su desarrollo.

En el **Capítulo 2**, veremos cómo se ubica la selección de un candidato revisando el funcionamiento del proceso de selección de personal dentro de una entidad así como las características que tiene la toma de decisiones para elegir a un aspirante u a otro. Se revisarán distintos modelos usados para la evaluación de desempeño, en la que explicaremos dónde se sitúa la idea del proyecto. Se explicará el algoritmo multicriterio utilizado para la realización del proyecto, así como una visión de los aspectos que van a ser tenidos en cuenta para la valoración de los candidatos, obtenidos del baremo de la Universidad de Jaén.

En el **Capítulo 3**, se explicará una de las tecnologías usadas en el sistema. El lenguaje XML. Se verá su importancia para el sistema, y en qué se basa su funcionamiento. Se dará una visión de en qué parte es necesario su uso, dado el esquema actual de baremación en la Universidad de Jaén, y algunas nociones para, en un futuro, poder hacer los cambios necesarios para nuevas necesidades que surjan.

El **Capítulo 4** analiza la resolución del problema desde el punto de vista de la Ingeniería del Software siguiendo las actividades que nos marca para este tipo de desarrollo software. Detallaremos en primer lugar los requerimientos necesarios para el proyecto, tanto los funcionales como los no funcionales. Se pasa en el siguiente punto al análisis del sistema en el que pensamos en qué usuarios van a usar el programa, de forma que el diseño este pensando en ellos, adaptándose a las necesidades generales de los usuarios potenciales. Plantearemos los casos de uso de nuestra aplicación, y los escenarios más comunes que nos podemos encontrar. En la posterior fase de diseño, Se verá cómo vamos a resolver el problema, es decir, el diseño por el que hemos optado. Para ello, reflejaremos los diagrama de clases, el diseño de los datos para la base de datos y los esquemas que determinan la interfaz del sistema. Para finalizar el capítulo, se han incluido las pruebas realizadas para comprobar que el sistema funciona correctamente y que cumple los requisitos que se le exigen. Se incluyen los test realizados, y los resultados de validación que se han obtenido.

En el **Capítulo 5**, expondremos las conclusiones a las que hemos llegado con el desarrollo del proyecto. También explicaremos cuáles son las características mejorables del programa y las limitaciones que hemos encontrado.

Finalmente se adjuntan los anexos con el manual de instalación y de usuario de la aplicación. Además, en otro anexo, se detalla el **esquema XML** a seguir, que será la estructura que deben cumplir los datos de entrada para que sean reconocidos por el sistema.

Capítulo 2:

Selección de Personal y Toma de Decisiones
Multicriterio en Selección de Personal

La selección de personal, es una actividad habitual en las empresas y en cualquier tipo de organización, y puede integrarse en la tarea de la evaluación del desempeño, que tiene como principal objetivo, el estudio y análisis de la capacidad de los empleados en sus actividades habituales en el ejercicio de su trabajo.

El proceso de selección, consiste en distinguir entre diferentes alternativas y escoger una atendiendo a diferentes criterios. Los procesos de toma de decisiones los utilizamos diariamente en nuestra vida cotidiana para seleccionar una solución, entre varias opciones. En el ámbito empresarial, la contratación, realizada por el Departamento de Recursos Humanos, evalúa el rendimiento de los empleados, para mejoras salariales, planes de formación, planes de promoción, contratación, etc., cuya evaluación, lleva asociado un proceso de toma de decisiones por parte del equipo directivo de este departamento.

En este capítulo vamos, en primer lugar, a definir qué son los recursos humanos de la empresa, las partes que lo componen, así como enmarcar el proceso de selección de personal en el área en la que tiene lugar para posteriormente revisar cómo se realiza el proceso de selección de personal. A continuación, describiremos diferentes conceptos básicos y las características propias de los problemas de toma de decisiones y su relación con los procesos de evaluación.

Por último, se revisará el modelo de decisión multicriterio TOPSIS [8] usado en nuestro sistema, analizando su funcionamiento y las fases que lo componen, así como un ejemplo de su utilización.

2.1 Definición de los Recursos Humanos en la empresa y áreas que lo conforman

Una empresa cuenta con diversos tipos de recursos que le permiten funcionar y alcanzar sus metas. Los empleados, trabajadores y colaboradores son quienes conforman lo que se conoce como los recursos humanos de una entidad.

El concepto también se utiliza para nombrar al departamento, la oficina o la persona que se dedica a analizar, elegir, contratar, formar y retener a los trabajadores de una compañía.

Por lo general, la política de recursos humanos (RR.HH.) se centra en lograr que los empleados estén en la misma sintonía que la empresa, haciendo coincidir los anhelos y las aspiraciones de los trabajadores con la estrategia de la compañía.

A grandes rasgos, la función de Recursos Humanos está compuesta por áreas tales como Reclutamiento y Selección, Compensaciones y Retribuciones, Capacitación y Desarrollo y otras diversas operaciones. Dependiendo de la empresa o institución donde la función de Recursos Humanos opere, pueden existir otros grupos que desempeñen distintas responsabilidades que pueden tener que ver con aspectos como por ejemplo la administración de la nómina de los empleados y el manejo de las relaciones con sindicatos entre otras.

Así, las funciones o áreas relacionadas con los RR.HH. se pueden resumir en: [9]

Reclutamiento y Selección

La función de incorporación de nuevo personal radica en determinar el perfil de la persona que necesitamos para cubrir el puesto que estamos ofreciendo, recluir o convocar a los aspirantes que cumplan con dicho perfil, evaluarlos a través de pruebas y entrevistas, seleccionar y contratar al más idóneo (o a los más idóneos), e inducirlo y capacitarlo para que se adapte a su nuevo puesto y a la empresa lo más pronto posible.

Compensaciones y Retribuciones

Consiste en la gestión de todo lo relacionado a las percepciones, prestaciones y beneficios de los trabajadores. Por ejemplo, consiste en determinar y gestionar sus remuneraciones (sueldos y salarios), sus horarios de trabajo, sus vacaciones, sus retenciones (por impuestos, pensiones y seguros), descuentos, etc.

Educación y capacitación

Es educar, capacitar y adiestrar constantemente a los trabajadores:

- *Educación*: consiste en proveer valores y actitudes.
- *Capacitación*: consiste en proveer conocimiento, para ello podemos optar por contratar instructores externos, enviar a nuestros empleados a instituciones especializadas, proporcionarles literatura o textos sobre un determinado tema, etc.
- *Adiestramiento*: consiste en proveer conocimiento práctico, con el fin de que el trabajador adquiera habilidades y experiencia. Se da mientras el trabajador cumple con su trabajo.

Comunicación

La función de comunicación es procurar que en el negocio o empresa exista una comunicación clara, adecuada y eficaz.

La comunicación debe expresarse en un lenguaje claro, simple y comprensible para el receptor. Debe ser lo más precisa posible, sin el uso de adornos lingüísticos ni información de más.

Liderazgo

La función de liderazgo es influir o inducir a los trabajadores para que realicen sus trabajos o tareas, y cumplan los objetivos, de manera eficiente, con entusiasmo y por voluntad propia.

A través de un buen liderazgo logramos influencia en los trabajadores para que nos sigan y acepten nuestras decisiones sin poner dudas ni condiciones.

Motivación

La motivación radica en el acto de animar a los trabajadores, con el fin de que tengan un mejor desempeño en el cumplimiento de los objetivos.

A través de la motivación, logramos un mejor desempeño, una mayor productividad, mayor eficiencia, creatividad, responsabilidad y un mayor compromiso por parte de los trabajadores.

Creación y dirección de equipos de trabajo

Consiste en crear y dirigir grupos o equipos de trabajo, guiarlos hacia el cumplimiento de sus objetivos, motivarlos, mantener la armonía del grupo, y resolver los problemas o discrepancias que puedan suceder.

El crear grupos de trabajo, no solo permite que el trabajador se sienta más seguro, optimista y motivado, sino que al trabajar en equipos se crea un refuerzo entre los trabajadores y, por tanto, la producción del grupo o equipo termina siendo mayor a lo que se lograría con trabajadores trabajando por separado.

Control y evaluación del desempeño

Otra función trata de controlar y evaluar constantemente el desempeño de los trabajadores, así como su compenetración con el puesto y con la empresa.

Para ello podemos hacer uso de técnicas como la asignación de criterios, en donde a cada trabajador le vamos asignando una puntuación en distintos criterios como por ejemplo responsabilidad, puntualidad, productividad, iniciativa, trabajo en equipo, pulcritud en el trabajo, etc., puntuación que se va acumulando, por ejemplo, cada tres o seis meses.

Promoción y manejo de empleados claves

Consiste en mantener en la empresa a los empleados que realmente hacen que ésta crezca, evitando que la competencia se los lleve, y se lleve con ellos, no sólo a sus clientes, sino también los conocimientos internos del negocio.

Para ello debemos reconocer y elogiar sus desempeños o logros, mostrar interés por su bienestar, hacer que se sientan comprometidos con la empresa, o usar otras técnicas de motivación.

De todas funciones o áreas anteriormente mencionadas, el proyecto actual se centra en la función de Reclutamiento y Selección.

2.2 Selección de personal

Una vez que se conocen las áreas del departamento de recursos humanos, veremos a continuación más en profundidad la que se refiere a la selección de personal.

Tradicionalmente, se define [1] como un procedimiento para encontrar a la persona que cubra el puesto adecuado, es decir escoger entre los candidatos reclutados a los más adecuados, para ocupar los cargos existentes en la empresa, tratando de mantener o aumentar la eficiencia y el rendimiento del personal.

La selección del Personal tiene como principales **objetivos** [10]:

- Definir políticas y programas para garantizar la incorporación del personal idóneo a la entidad.
- Seleccionar candidatos adecuados para el acceso, los traslados, las promociones y las actividades de formación y desarrollo.

La selección de personal, es una actividad habitual en las empresas y en cualquier tipo de organización, y puede integrarse en la tarea de la evaluación del desempeño, que tiene como principal objetivo, el estudio y análisis de la capacidad de los empleados en sus actividades habituales en el ejercicio de su trabajo.

El proceso de Selección, consiste en distinguir entre diferentes alternativas, y escoger una atendiendo a diferentes criterios. Los procesos de toma de decisiones, los utilizamos diariamente en nuestra vida cotidiana para seleccionar una solución, entre varias opciones. En el ámbito empresarial, la contratación, realizada por el Departamento de Recursos Humanos, evalúa el rendimiento de los empleados, para mejoras salariales, planes de formación, planes de promoción, contratación, etc., cuya evaluación, lleva asociado un proceso de toma de decisiones por parte del equipo directivo de este departamento.

Fases del proceso de selección de personal

Para conocer más sobre las fases de las que consta el proceso de selección de personal, detallaremos cada una de ellas:

- Análisis del puesto a cubrir

El proceso de selección se inicia cuando se presenta una vacante, que se entiende como la disponibilidad de una tarea a realizar o puesto a desempeñar, que puede ser de nueva creación, o debido a imposibilidad temporal o permanente de la persona que lo venía desempeñando.

- Definición del perfil buscado

Recibida la necesidad de personal, se recurrirá al análisis y evaluación de puestos, con el objetivo de determinar los requerimientos que debe satisfacer la persona para ocupar el puesto eficientemente, así como el salario a pagar. En caso de no existir dicho análisis y evaluación, deberá procederse a su elaboración para poder precisar los requerimientos que se necesitan.

- Reclutamiento
 - Interno [11]

El reclutamiento es interno cuando, al presentarse determinada vacante, la empresa intenta llenarla mediante la reubicación de los empleados, los cuales pueden ser ascendidos (movimiento vertical) o transferidos (movimiento horizontal) o transferidos con promoción (movimiento diagonal).

- Externo [12]

Entre las fuentes de reclutamiento externo tenemos los siguientes:

- Candidatos espontáneos. Son aquellos que se presentan en una empresa para dejar su hoja de vida o envían por correo ordinario o electrónico su "currículum vitae".
- Recomendaciones de los empleados de la empresa. En ocasiones, los empleados de una empresa pueden recomendar a personas para las

vacantes que surjan lo cual se considera ventajoso, ya que en posiciones muy especializadas conocen a colegas que pudieran llenarlas. Además, es muy probable que el recomendado se desempeñe mejor para corresponder a quien lo recomendó.

- c) Anuncios de prensa. Los anuncios en los periódicos sirven para buscar candidatos cuando son posiciones que por su especialidad son difíciles de llenar.
- d) Agencias de empleos tradicionales / agencias de empleo online. Son empresas que actúan como un puente entre las vacantes y los candidatos. Las agencias de empleo online pueden ser muy útiles ya que se pueden obtener grandes cantidades de currículos en poco tiempo y de forma electrónica, lo que facilita su almacenamiento.
- e) Instituciones educativas. Las universidades, los institutos, las academias técnicas y centros educativos, son también una buena fuente de candidatos.

- Evaluación de candidatos

Una vez recibidos los currículos provenientes de presentaciones espontáneas, avisos periodísticos u otras fuentes, se clasifican para examinar los conocimientos y la experiencia de los candidatos; esta tarea puede ser realizada por el cliente o por el departamento de RRHH sobre la base de los requisitos formales que han acordado.

Separados los currículos que reúnen los requisitos formales, se procede a realizar la primera entrevista. El encargado de la selección ya cuenta con el perfil, elaborado con la empresa.

Los candidatos que más se aproximan a él son enviados para el chequeo técnico a fin de continuar con aquellos que reúnen los conocimientos y las destrezas indispensables, cuando el puesto por cubrir lo requiere.

- Entrevista(pruebas)

Se entrevista a los candidatos técnicamente aptos, y se evalúa a los que resultan de mayor interés. Según Rodríguez López [13] en el proceso de selección se pueden dar alguna de las modalidades siguientes de entrevista:

- a) La entrevista individual es la más utilizada, y, como su nombre indica, presenta a un entrevistador con un único entrevistado.
- b) Entrevista en grupo, donde un entrevistador entrevista a varios candidatos a la vez. Suele realizarse ofreciendo al grupo una tarea sobre la que posteriormente habrán de discutir. Su uso está indicado para detectar los factores de interrelación de los candidatos: cooperación, competitividad, liderazgo, etc.
- c) Entrevista de tribunal, es la efectuada por un grupo de entrevistadores con un solo candidato. Este tipo de entrevista es poco eficaz, por lo general, y se utiliza más por imperativos legales, como por ejemplo en un concurso-oposición, y, por razones técnicas, es poco frecuente reunir a un grupo de entrevistadores expertos.
- d) «role-playing», en algunas ocasiones y debido a características especiales del puesto, el entrevistador o grupo de entrevistadores puede plantear a los candidatos que hagan una representación de papeles; los propios entrevistadores pueden asumir igualmente una variedad de roles.

El paso siguiente, una vez realizada la entrevista, es la evaluación psicológica, tras la cual, pasaremos a:

- Decisión final

Superada la etapa de evaluación psicológica, el responsable de RRHH está en condiciones de realizar un profundo examen de cada candidato para presentar a los finalistas previa ponderación comparativa de la muestra. Resulta deseable presentar entre dos y tres candidatos.

La elección final la realiza la empresa. La presentación de los candidatos es precedida por el currículo y el informe de evaluación psicológica de cada uno. Cuando

se trata de un conjunto de personas se presenta un ranking en función de los rasgos de perfil considerados.

Un resumen del proceso de selección en todos sus pasos se puede ver en la Figura 2.1.



Figura 2.1 . Proceso de Selección

En nuestro caso, el sistema no servirá para todas las fases del proceso de selección, ya que hay algunas, como la entrevista, que no ha sido objeto de nuestro estudio. Nos centraremos principalmente, en la valoración de los candidatos siguiendo un modelo de toma de decisiones, quedando la responsabilidad tanto del análisis del puesto a cubrir, la definición del perfil buscado, y el reclutamiento, a cargo del departamento de RR.HH. o de las personas responsables de tal fin.

Una vez situada la selección de personal pasaremos a ver el proceso de toma de decisiones, sus características, y explicaremos cómo nos ayudan los métodos de toma de decisiones multicriterio en la selección de personal. Además, explicaremos el método multicriterio TOPSIS, elegido para nuestro caso de estudio, así como un ejemplo de su aplicación.

2.3 Toma de Decisiones

Como se ha descrito anteriormente, la selección de personal, consiste en elegir a un candidato para un puesto específico. Para ello, hay que tomar una decisión. En este apartado, se desgranará en qué consiste la toma de decisiones, para conocer sus elementos básicos, y poder identificar en el problema que se está tratando, dónde encajan los ítems a evaluar, los candidatos, y las valoraciones que se están realizando.

Una decisión, en un sentido amplio, consiste en elegir, de manera consciente, una o varias alternativas de un conjunto de opciones o elecciones posibles, siempre persiguiendo un objetivo concreto. Todos los procesos de toma de decisiones, tienen en común una serie de elementos básicos que son [14]:

- Un conjunto finito de soluciones o de decisiones posibles.
- Un criterio o conjunto finito de criterios a tener en cuenta.
- Un conjunto de valoraciones realizadas por el evaluador o los diferentes evaluadores para cada criterio.
- Un conjunto de circunstancias o elementos que definen el entorno o contexto del problema decisional.

La Teoría de la Decisión ha propuesto múltiples y diversos modelos para resolver problemas de toma de decisiones atendiendo a las diferentes características presentes en cada uno de ellos.

Podemos clasificar los problemas de toma de decisiones atendiendo a varios factores como son:

- Marco o contexto en el que se plantea el problema,
- Número de expertos que participa en el proceso de decisión y
- Número de criterios a evaluar

El contexto en el que vamos a plantear el problema, va a ser de los puntos más importantes que debemos de delimitar correctamente. De él dependerá el tipo de información que vamos a utilizar en el proceso de decisión.

2.3.1 Clasificación de los problemas de toma de decisiones

I. Clasificación según el número de expertos

Esta clasificación hace referencia al número de expertos o evaluadores que toman parte en el problema. Son dos las distinciones que se hacen en la literatura [15] en cuanto a los problemas de toma de decisiones en función del número de expertos participantes:

1. *Unipersonales o individuales.* Las decisiones en este tipo de problemas son tomadas por un único agente o individuo.
2. *Multi-experto.* Las decisiones en este caso son tomadas por un grupo de agentes o individuos.

II. Clasificación según el número de criterios

Atendiendo al número de criterios a evaluar en un problema de toma de decisiones, podemos clasificar a éstos en dos grandes categorías [16]:

1. *Con un solo criterio o atributo.* En este caso los problemas sólo tienen en cuenta un atributo de las alternativas para la toma de decisión.
2. *Con múltiples criterios o atributos.* Los problemas multi-criterio tienen en cuenta al menos dos criterios para la evaluación de cada alternativa. Esta categoría, al ser la que es más interesante para el sistema que se está tratando, se explica con más detalle en el apartado 2.2.5.

III. Clasificación según el contexto o ámbito decisional

Los problemas de toma de decisiones pueden desarrollarse en diferentes contextos o ámbitos de aplicación. Según la literatura clásica existente [17], los problemas decisionales pueden clasificarse atendiendo al contexto de definición en el que se formulan:

1. *Contexto o ambiente de certidumbre.* Una situación o contexto de certidumbre se produce cuando se tiene información precisa sobre todas las situaciones que pueden afectar al problema.

2. *Contexto o ambiente de riesgo.* Un ambiente de riesgo en un problema decisional se puede definir como aquel en el que alguno de los factores o elementos que intervienen en el mismo están sujetos al azar, es decir, tienen asociados una probabilidad.
3. *Contexto o ambiente de incertidumbre.* Una situación de incertidumbre en un problema decisional, se plantea cuando la información disponible sobre las opciones elegibles es vaga o imprecisa. Este último ámbito o contexto es el que más se aproxima a la toma de decisiones características o propias de la vida real y del comportamiento humano, ya que la imprecisión es inherente a los individuos.

La Teoría de la Decisión ha proporcionado distintos modelos decisionales determinísticos y probabilísticos. Los métodos determinísticos ignoran la incertidumbre mientras que los problemas probabilísticos consideran que la incertidumbre puede representarse mediante distribuciones de probabilidad.

2.3.2 Modelado de las opiniones en la toma de decisiones

Un correcto marco de expresión de la información permite que los expertos sean capaces de manifestar sus dictámenes de forma más correcta, lo que afectaría de forma positiva al resultado del proceso decisional.

La naturaleza de las alternativas, la formación y el conocimiento de los evaluadores sobre los diferentes criterios sobre los que evaluar cada alternativa son algunos de los factores que influyen en la forma de modelar la información en cada problema.

Haciendo una breve revisión de los diferentes conceptos tratados en la literatura [18] sobre el modelado de la información, en esta memoria consideraremos los dos aspectos siguientes:

1. **El dominio de expresión de la información.** En el contexto en el que trabajamos se entiende por dominio de expresión al conjunto de valores utilizado por los

diferentes evaluadores o expertos para emitir sus opiniones. La elección de uno u otro dominio de expresión dependerá de varias causas, entre las que podemos destacar:

- a) *La naturaleza de la información: cuantitativa o cualitativa.* En función de la naturaleza de los atributos o criterios a valorar el dominio de expresión de la información puede variar. Así, la consideración de criterios precisos y de naturaleza cuantitativa conlleva la utilización de valoraciones de tipo numérico, mientras que la consideración de criterios de naturaleza cualitativa o subjetiva conlleva la utilización de valoraciones en forma de palabras o términos lingüísticos.

- b) *La utilización de grupos heterogéneos de expertos con diferentes y múltiples grados de conocimiento.* Muchas veces los grupos de evaluadores no proceden del mismo campo de conocimiento, poseen diferente información o no tienen el mismo grado de experiencia. Concretamente, el conocimiento de los evaluadores sobre distintas alternativas atendiendo a diversos criterios puede ser desigual, lo que podrá aconsejar la emisión por parte de los expertos de valoraciones expresadas en diferentes dominios de expresión.

El modelado de la información en los problemas decisionales persigue que los expertos o evaluadores puedan emitir sus valoraciones de la manera más precisa posible, de forma que éstas se adapten a su nivel de conocimiento. Esto tiene como consecuencia un aumento de la garantía de éxito de la decisión adoptada por los mismos. Los dominios de expresión más utilizados en los problemas de toma de decisiones, según la literatura existente [19], son el *dominio numérico*, *dominio intervalar* y el *dominio lingüístico*.

2. **La estructura de representación de la información.** Una vez conocidos los dominios de expresión más comunes en los problemas de toma de decisiones,

otro concepto clave en el modelado de opiniones es la estructura de representación de la información. La importancia de este concepto recae en que las opiniones de los expertos del problema se almacenarán en una estructura de representación sobre la que se operará posteriormente. A continuación se enumeran las estructuras de representación más habituales en este tipo de problemas.

a) Representación de la información mediante *órdenes de preferencia*.

En este caso la información se representa mediante el establecimiento de un ranking de las diferentes alternativas de solución existentes en el problema. La ordenación de las mismas se realiza teniendo en cuenta la idoneidad de cada alternativa como solución al problema.

b) Representación de la información mediante *vectores de valoración*.

Mediante este método de representación los expertos asocian un valor de utilidad a cada alternativa, el cual representará la evaluación del experto sobre la alternativa en un cierto dominio de expresión.

c) Representación de la información mediante *relaciones de preferencia*.

En los modelos clásicos de la teoría de la decisión los individuos representan sus preferencias sobre un conjunto de alternativas mediante relaciones binarias ordinarias.

Podemos definir nuestro proyecto como desde el punto de vista de la toma de decisiones, según el número de expertos, como un sistema multi-experto, si la decisión es encargada al departamento de RRHH, según el número de criterios, como un sistema con múltiples criterios o multi-criterio, según el ámbito decisional, se trabaja bajo un ambiente de incertidumbre, con información cuantitativa, ya que los criterios son valorados con una puntuación concreta, y con un dominio de los datos numérico o intervalar en ciertos casos.

2.3.3 Esquema básico de resolución en problemas de Toma de Decisiones

A pesar de que la teoría de toma de decisiones tiene muchos y diferentes ámbitos de aplicación y por tanto, muchos y diferentes tipos de problemas, existe un esquema básico de resolución común a todos ellos [16]. A continuación se describen brevemente las fases de las que consta dicho esquema (ver Figura 2.2):

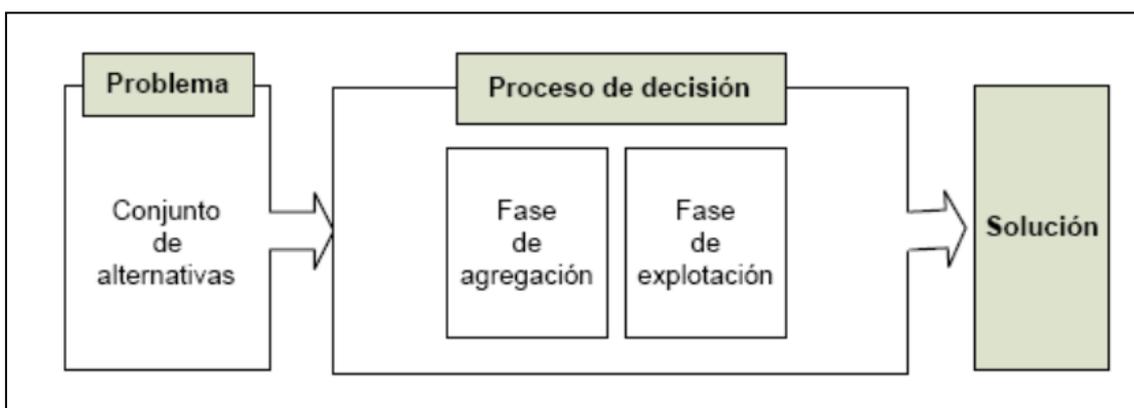


Figura 2.2. Proceso de resolución de un problema de toma de decisiones

Una vez conocemos el problema que queremos resolver, y conocemos el conjunto de alternativas, pasamos al proceso de decisión, que consta de la *fase de agregación*, y *explotación*.

La *fase de agregación* tiene como objetivo añadir la información facilitada por los evaluadores con el fin de obtener una valoración global sobre los diferentes criterios que permita tomar una decisión.

En segundo lugar se realiza una *fase de explotación*. En esta fase las valoraciones globales obtenidas en la fase anterior deben ser ordenadas [20] para obtener el conjunto solución de alternativas.

En la TD (Toma de Decisiones) normalmente se aplica un grado de selección para ordenar las distintas alternativas. Sin embargo, cuando utilizamos los procesos de TD en problemas de “evaluación”, podemos encontrarnos que en ocasiones es necesario ordenar las alternativas que se evalúan o bien, a veces lo que necesitamos es obtener un valor general del elemento o elementos evaluados para saber cuál es su grado de calidad.

Por tanto, en los procesos evaluativos, dependiendo del tipo de evaluación abordada, diseñaremos distintas fases de explotación para obtener el resultado buscado en cada tipo de problema.

2.3.4 Análisis de decisión y evaluación

La teoría de toma de decisiones intenta ayudar a los individuos a discernir entre alternativas complejas de una forma racional, mediante un proceso de análisis objetivo y de estudio de las diferentes alternativas planteadas en el problema.

En la literatura existen referencias [21] que ponen de manifiesto que no todos los agentes decisores realizan la toma de decisiones mediante un análisis racional de las alternativas posibles, sino que toman sus decisiones en función de otros tipo de factores, como pueden ser emocionales, externos, subjetivos, etc.

El esquema general del proceso de decisión mostrado en la Figura 2.2 puede ampliarse con mayor detalle (véase la Figura 2.3). En las primeras fases se lleva a cabo un procedimiento de análisis de las alternativas de forma totalmente racional, denominado *análisis de decisión*, consiste en identificar la decisión, y las alternativas que van a ser tenidas en cuenta, mientras que la última fase del proceso, al ser tomar la decisión final, no está basada necesariamente en factores racionales. De este modo, el análisis de decisión se ajusta perfectamente a las necesidades y objetivos que persiguen los procesos de evaluación, ya que realiza un estudio metódico y razonado de las distintas alternativas planteadas en el problema.

En nuestro caso, solo vamos a usar las fases correspondientes al análisis de la decisión, ya que el sistema no es capaz de distinguir la decisión final que se quiere tomar, ni las alternativas de las que va a constar. Se construirá el modelo, y recopilando la información de los candidatos y, sabiendo la valoración de las alternativas de cada uno, se podrá proponer una selección de las mejores alternativas y proseguir con las demás fases, hasta que el responsable correspondiente tome la decisión.

No obstante, para tener una idea más clara de cada una de las fases del análisis de la decisión, se describirán a continuación.

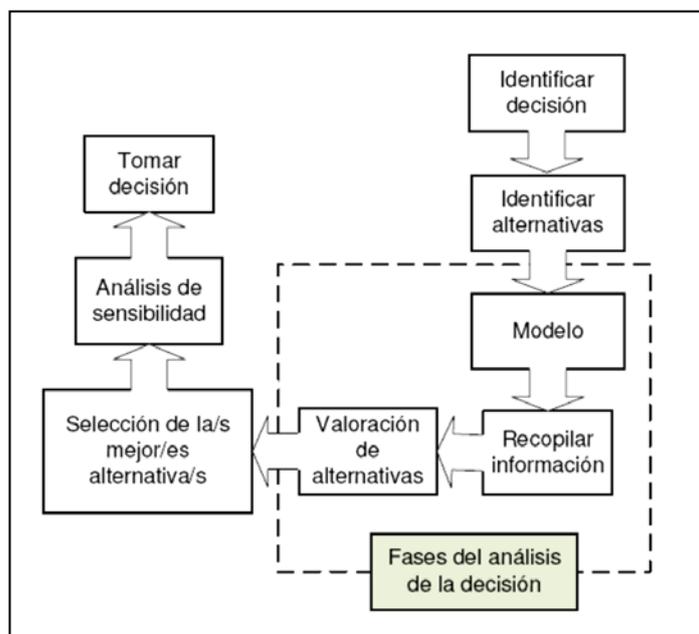


Figura 2.3 Esquema de un proceso de toma de decisión

En relación con la Selección de personal, podemos identificar cada una de las fases, con las seguidas en el proceso de toma de decisiones:

1. Identificar decisión: Consistirá en saber lo que se está buscando. Es decir, concretar la finalidad que va a tener nuestro proceso de selección. En nuestro caso, conocer al candidato ideal para el proceso de selección que se está siguiendo.
2. Identificar alternativas Las alternativas, en el caso de la selección de personal, van a venir dadas por los aspirantes que se presenten a ser elegidos. Serán los elementos entre los que tendremos que escoger al más adecuado.
3. Modelo. En este punto, definiremos los elementos que vamos a tener en cuenta a la hora de hacer la selección. Vendrán determinados por el tipo de plaza y el perfil del trabajador que se quiera contratar. Dependiendo del perfil, tendremos una serie de características o ítem, que van a formar parte del modelo que nos servirá para la toma de la decisión.

4. Recopilar información: En este caso, consistirá en relacionar el modelo que tenemos, con el conjunto de alternativas. Es decir, se verá que información presenta el candidato y se hará su correspondencia con el modelo, que se ha definido anteriormente.
5. Valoración de las alternativas. Consistirá en aplicar un método para hacer que la información que se ha obtenido, pase a tener un valor. Normalmente, la información va a venir acompañada de una importancia o ponderación, que se aplicará para llegar a la valoración de las alternativas.

Así, basándonos en estas fases, vamos a considerar en nuestro sistema los puntos correspondientes al *modelo*, que tendrá los criterios que van a formar parte de la selección además de *recopilar la información* de los aspirantes, dependiendo del modelo, y estableciendo *el valor de las alternativas*, para conseguir dar una puntuación a los aspirantes.

2.3.5 Toma de Decisiones con Múltiples Criterios

En el apartado anterior, se ha definido nuestro sistema como un problema de toma de decisiones con criterios múltiples. En este apartado definiremos de qué se trata este tipo de toma de decisiones.

La mayoría de los problemas de decisiones [22] en el ámbito económico, político, financiero, etc. son casi siempre problemas multicriterio, es decir, problemas en los que se tiene en cuenta varios factores a la vez.

De esta forma, el problema de seleccionar u ordenar un conjunto de alternativas sometidas a una evaluación multicriterio, no resulta ser un problema sencillo desde el punto de vista económico ni matemático. Normalmente, no hay una solución que cumpla todos los criterios a la vez. Es decir, es difícil encontrar la solución óptima. Por ello, la alternativa óptima como tal, se sustituye por la búsqueda de soluciones de compromiso.

Como hemos dicho, la mayoría de los problemas de decisión tienen una naturaleza multicriterio y no tendría sentido la selección de una decisión considerando

solamente un criterio de decisión. El problema es que a veces, algunos criterios entran en conflicto entre sí.

La solución del problema multicriterio, no depende sólo de la naturaleza del mismo, sino también del que toma la decisión. Cada decisor da una importancia relativa diferente a cada uno de los criterios seleccionados de acuerdo con la estructura de preferencias elegida en interactividad con el analista. El analista, tiene un papel importante en el proceso de decisión, ya que su ayuda es valiosísima para el decisor.

La formulación de un problema multicriterio puede expresarse en los siguientes términos:

$$\text{Max. } \{g_1(a), g_2(a), \dots, g_j(a), \dots, g_k(a) / a \in A\}$$

Ecuación 1. Términos de un problema multicriterio

Donde A es un conjunto de alternativas factibles, y $\{g_j(\cdot), j=1, \dots, k\}$ un conjunto de criterios de evaluación. Si bien se expresa como un problema de maximización, lo más normal es que algunos criterios deban de maximizarse y otros minimizarse al mismo tiempo, lo cual no representa ningún obstáculo para su consideración.

2.3.6 TOPSIS

Como modelo para la toma de decisiones con múltiples criterios, se ha escogido TOPSIS, que cumple los requisitos que pedíamos para el desarrollo de este proyecto. A continuación detallamos en qué consiste, qué elementos lo forman, y cómo se realizan los cálculos necesarios para llegar a la posible solución ideal.

Un proceso de toma de decisiones multicriterio (MCDM) como el que se va a usar en este proyecto, con m alternativas que son evaluadas por n atributos, puede verse como un sistema geométrico con m puntos en n-dimensiones en el espacio. Hwang and Yoon [8], desarrollaron “*Technique for Order Preference by Similarity to Ideal Solution*” (TOPSIS) basado en el concepto de que la alternativa elegida debería ser la distancia más corta de la solución positiva ideal y la mayor distancia a la solución ideal negativa.

Soluciones Ideales positiva y solución ideal negativa

Una solución ideal se define como una colección de niveles ideales en todos los criterios. Sin embargo, la solución ideal es inalcanzable o inviable. Como la ideal depende de límites y restricciones, se utiliza la elección racional con un proceso de normalización. Formalmente, la solución positiva ideal se denota como:

$$A^* = (x_1^*, \dots, x_j^*, \dots, x_n^*)$$

Ecuación 2. Solución ideal positiva

Donde x_j^* es el mejor valor para el criterio j-ésimo de todas las alternativas disponibles.

La composición de todas las calificaciones de todos los mejores valores disponibles es la solución positiva ideal, mientras que la solución negativa ideal se compone por todas las calificaciones peores alcanzables. La solución negativa ideal viene dada por:

$$A^- = (x_1^-, \dots, x_j^-, \dots, x_n^-)$$

Ecuación 3. Solución ideal negativa

Donde x_j^- es el peor valor posible para el criterio j-ésimo de todas las alternativas alcanzables

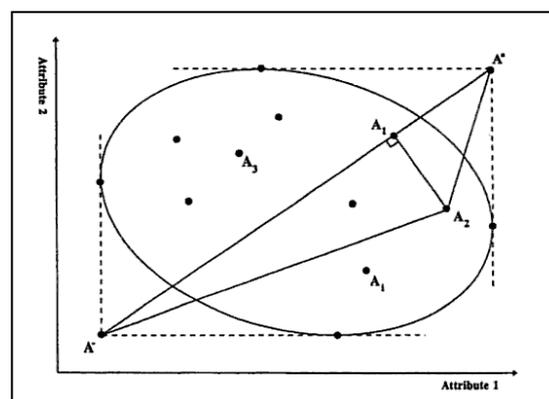


Figura 2.4 Distancia Euclídea al Ideal positivo, y al ideal negativo, en dos dimensiones

Entonces, ¿la alternativa elegida que es la más cercana a la solución ideal positiva de acuerdo con la alternativa elegida que está más lejos de la solución negativa ideal? A menudo, no concuerdan entre sí.

Por ejemplo, si consideramos A_1 y A_2 con respecto a los dos mejores atributos en la Figura anterior (Figura 2.4), A_1 es la más cercana a A^* , pero A_2 es la más lejana a A^- .

TOPSIS define un índice, llamado similitud (o cercanía relativa) a la solución positiva ideal combinando la solución positiva ideal y la lejanía a la solución negativa ideal.

El método se presenta como una serie de sucesivos pasos:

Paso 1: *Calcular las puntuaciones normalizadas:*

El vector de normalización es usado para calcular r_{ij} como:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}, i = 1, \dots, m; j = 1, \dots, n$$

Paso 2: *Calcular las puntuaciones normalizadas con pesos:*

Se calcula como:

$$v_{ij} = w_j r_{ij}, \quad i = 1, \dots, m; j = 1, \dots, n$$

Donde w_j es el peso del atributo j -ésimo.

Paso 3: *Identificar las soluciones ideal-positiva e ideal-negativa*

A^* y A^- están definidas en términos de valores normalizados con pesos:

$$A^* = \{v_1^*, \dots, v_j^*, \dots, v_n^*\} = \{(max_j v_{ij} | j = 1, \dots, n) | i = 1, \dots, m\}$$

$$A^- = \{v_1^-, \dots, v_j^-, \dots, v_n^-\} = \{(min_j v_{ij} | j = 1, \dots, n) | i = 1, \dots, m\}$$

Paso 4: *Calcular la medida de separación*

La separación de cada alternativa de cada solución positiva-ideal A^* , puede ser medida por la distancia Euclídea a la n -dimensión. La separación de cada alternativa con la solución ideal positiva A^* , viene dada por:

$$S_i^* = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^*)^2}, i = 1, \dots, m$$

De manera similar, la separación de la solución negativa ideal A^- , viene dada por:

$$S_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2}, i = 1, \dots, m$$

Paso 5: *Calcular las similitudes con respecto a la solución positiva-ideal:*

$$C_i^* = S_i^- / (S_i^* + S_i^-), i = 1, \dots, m$$

Hay que tener en cuenta que $0 \leq C_i^* \leq 1$ donde $C_i^* = 0$ cuando $A_i = A^-$ y $C_i^* = 1$ cuando $A_i = A^*$.

Paso 6: *Puesto en el orden de preferencia.*

Elegir una alternativa con el máximo C_i^* o tomar el orden de las alternativas de acuerdo al C_i^* en orden descendente.

Ejemplo de TOPSIS usando el modelo para la selección de Personal en la Universidad de Jaén

En este apartado, adaptamos el modelo de TOPSIS a un problema específico. En nuestro caso, lo aplicaremos al baremo que se sigue en la universidad de Jaén para la valoración de los candidatos a la bolsa de profesores sustitutos. Este baremo, está explicado con mayor detalle en el siguiente apartado.

Para el ejemplo que vamos a desarrollar, contamos con los siguientes elementos:

alternativas = {candidato1, candidato2, candidato3, candidato4}

criterios= {Formación, Actividad Docente, Investigaciones, Experiencia}

pesos= {w1, w2, w3, w4} donde: w1= 0,1, w2= 0,4, w3=0,3, w4=0,2

$m = 4$ Alternativas (candidatos) x_{ij} = puntuación de i con
respecto al criterio j $n = 4$ criterios $X = \{x_{ij}\}$ matriz de decisiónMatriz de decisión X

pesos	0,1	0,4	0,3	0,2
	Formación	Actividad Docente	Investigaciones	Experiencia
Candidato 1	7	9	9	8
Candidato 2	8	7	8	7
Candidato 3	9	6	8	9
Candidato 4	6	7	8	6

Tabla 2-I. Ejemplo de 4 candidatos evaluados en 4 categorías

Siguiendo las fases de TOPSIS obtenemos:

Paso 1(a): calculamos $(\sum x_{ij}^2)^{1/2}$ para cada columna:

pesos	0,1	0,4	0,3	0,2
	Formación	Actividad Docente	Investigaciones	Experiencia
Candidato 1	49	81	81	64
Candidato 2	64	49	64	49
Candidato 3	81	36	64	81
Candidato 4	36	49	64	36
$\sum x_{ij}^2$	230	215	273	230
$(\sum x_{ij}^2)^{1/2}$	15,166	14,663	16,523	15,166

Tabla 2-II. Paso 1(a) aplicación TOPSIS al ejemplo

Paso 1(b): dividimos cada columna por $(\sum x_{ij}^2)^{1/2}$ para conseguir r_{ij}

Pesos	0,1	0,4	0,3	0,2
	Formación	Actividad Docente	Investigaciones	Experiencia
Candidato 1	0,462	0,614	0,545	0,528
Candidato 2	0,528	0,477	0,484	0,462
Candidato 3	0,593	0,409	0,484	0,593
Candidato 4	0,396	0,477	0,484	0,396

Tabla 2-III. Paso 1(b) para conseguir r_{ij}

Paso 2: multiplicamos cada columna por w_j para conseguir v_{ij}

	Formación	Actividad Docente	Investigaciones	Experiencia
Candidato 1	0,046	0,246	0,163	0,106
Candidato 2	0,053	0,191	0,145	0,092
Candidato 3	0,059	0,164	0,145	0,119
Candidato 4	0,040	0,191	0,145	0,079

Tabla 2-IV. Paso 2 aplicación TOPSIS

Paso 3: determinamos la soluciones A^* y A^-

	Formación	Actividad Docente	Investigaciones	Experiencia
Candidato 1	0,046	0,246	0,163	0,106
Candidato 2	0,053	0,191	0,145	0,092
Candidato 3	0,059	0,164	0,145	0,119
Candidato 4	0,040	0,191	0,145	0,079
Max:	0,059	0,246	0,163	0,119
Min:	0,040	0,164	0,145	0,079

Tabla 2-V. Obtención A^* y A^- .

Donde: $A^* = \{0,059, 0,246, 0,163, 0,119\}$

$A^- = \{0,040, 0,164, 0,145, 0,079\}$

Paso 4(a): Calculamos las distancias de cada candidato a la solución ideal.

Candidato 1	0,019
Candidato 2	0,064
Candidato 3	0,084
Candidato 4	0,073

Tabla 2-VI. Obtención distancias a la solución ideal.

Paso 4(b): Calculamos las distancias de cada candidato a la solución anti-ideal.

Candidato 1	0,088
Candidato 2	0,033
Candidato 3	0,044
Candidato 4	0,027

Tabla 2-VII. Obtención distancias a la solución anti-ideal.

Paso 5: Calculamos la similitud a la solución ideal

Candidato 1	0,825
Candidato 2	0,342
Candidato 3	0,345
Candidato 4	0,273

Tabla 2-VIII. Similitudes a la solución ideal.

Paso 6: Establecemos el ranking de preferencia.

1. **Candidato 1,**
2. Candidato 3,
3. Candidato 2,
4. Candidato 4

La elección, tras el paso 6, nos indica que el mejor candidato, es decir, la mejor alternativa de las cuatro con las que contábamos, sería la correspondiente al **Candidato 1**.

2.3.7 Modelo del baremo de Contratación

Una vez que se ha descrito TOPSIS, y se ha explicado con un ejemplo basado en el baremo de contratación, pasaremos a ver en qué consiste dicho baremo. El motivo de usarlo, viene dado porque los candidatos que se presentan como PDI en la Universidad de Jaén, son calificados de acuerdo a él. El objetivo de nuestro proyecto, es dar una valoración sobre los candidatos que aspiran a conseguir un puesto de PDI, por lo que debemos de conocer en qué puntos son evaluados dichos candidatos.

Este apartado nos servirá para conocer las alternativas y los criterios que van a ser tenidos en cuenta para aplicar la toma de decisión basada en criterios múltiples.

En la mayoría de los puestos de trabajo, se requieren una serie de aspectos, que conforman el perfil del puesto de trabajo. Para ello, se crean los criterios que van a formar parte de la evaluación del candidato que aspira a ese puesto de trabajo.

En la Universidad de Jaén, así como en la mayoría de las instituciones de carácter público, como la sanidad, educación etc., el baremo se publica para que los aspirantes conozcan en qué ítems van a ser valorados.

El baremo, consiste en una serie de apartados, llamados Categorías, que abarcan aspectos como la Formación Académica, la Actividad docente e investigadora, la experiencia profesional, representación en órganos de gobierno y gestión universitaria, así como un informe del departamento, y una serie de méritos que se consideran preferentes.

Dentro de cada Categoría, se encuentran las Subcategorías, entre las que se localizan los premios obtenidos fin de carrera, titulaciones, doctorados, becas que se le han concedido al aspirante, cursos de postgrado, entre otros.

En el siguiente nivel de detalle, encontramos los Criterios o méritos, que son la base del baremo. Como criterios, se pueden nombrar algunos como la nota media de las asignaturas cursadas, premios nacionales o tiempo de docencia impartido.

Cada criterio o mérito, según el baremo, tiene un peso, que va a hacer que cada mérito tenga una importancia o ponderación en cada uno de los apartados. Así mismo, hay una modulación de acuerdo a la afinidad al área de la plaza. Es decir, se computa el 100% del valor asignado en el baremo las contribuciones propias del área, un 50% las de las áreas afines, y un 20% las de áreas no afines.

El valor final obtenido en cada criterio, vendrá determinado por el valor unitario (que dependerá de qué criterio se esté calculando), es decir, el número de méritos que se tenga en un criterio multiplicado por el peso que tenga ese criterio, y ponderado en función de la afinidad.

Todo esto, nos da la base para saber los datos que tenemos que recoger de los documentos de los aspirantes.

Capítulo 3:

Tecnologías usadas en el Sistema.XML.

El motivo de este capítulo es presentar una breve descripción de la tecnología XML [7]. Conociendo el funcionamiento de esta tecnología, se podrá comprender en qué se ha utilizado en este proyecto y con qué finalidad.

En primer lugar, se definen las tecnologías que, basándose en XML, van a ser usadas para hacer que la información sea legible y usada por el sistema. Se definirá el lenguaje de marcas XML, que es el que se ha escogido para procesar los datos en el sistema. Se explicará lo que es un esquema, y se hablará de en qué consiste el esquema seguido, y se justificará en base al baremo de contratación definido en el capítulo 2.

Para finalizar, se explica un ejemplo del XML válido para el esquema definido, así como para qué se ha usado InfoPath en la generación de un formulario para los aspirantes.

3.1 Qué es XML

Para estructurar todo lo descrito en el baremo anteriormente mencionado en el capítulo 2.2.6, necesitamos de un lenguaje que lo esquematice. Para ello, se va a usar XML, por ser un lenguaje que sigue unos estándares, y que está bastante extendido su uso para la transmisión de información, al ser fácilmente interpretado tanto por personas como por máquinas.

Según W3C [7] se definen las tecnologías XML como: Un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje [23] muy similar a HTML pero su función principal es describir datos más que mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

XML [24] es *una tecnología* sencilla que tiene a su alrededor otras que la complementan y la hacen que tenga posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Entre las tecnologías [25] usadas, se pueden destacar:

XSL : Lenguaje Extensible de Hojas de Estilo, cuyo objetivo principal es mostrar cómo debería estar estructurado el contenido, cómo debería ser diseñado el contenido de origen y cómo debería ser paginado en un medio de presentación como puede ser una ventana de un navegador Web o un dispositivo móvil, o un conjunto de páginas de un catálogo, informe o libro.

XPath: Lenguaje de Rutas XML, es un lenguaje para acceder a partes de un documento XML. Sirve para facilitar la recuperación de información, a través de consultas.

XLink: Lenguaje de Enlace XML, es un lenguaje que permite insertar elementos en documentos XML para crear enlaces entre recursos XML.

XPointer: Lenguaje de Direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido.

XQL: *Lenguaje de Consulta XML, es un lenguaje que facilita la extracción de datos desde documentos XML. Ofrece la posibilidad de realizar consultas flexibles para extraer datos de documentos XML en la Web.*

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

3.2 Ventajas del XML

Las ventajas que tiene XML frente a otras alternativas se pueden resumir en:

Es **extensible**: Después de haber sido diseñado y estar puesto ya en funcionamiento, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.

El analizador es un componente estándar, **no es necesario crear un analizador específico** para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan errores y se acelera el desarrollo de aplicaciones.

Si un tercero decide usar un documento creado en XML, **es sencillo entender** su estructura y procesarla. Además, mejora la compatibilidad entre aplicaciones. Podemos comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos. Es decir, sería posible tener una aplicación en Linux con una base de datos Postgres y comunicarla con otra aplicación en Windows y Base de Datos MS-SQL Server.

Transformamos datos en información, pues se le añade un significado concreto y los asociamos a un contexto, con lo cual tenemos flexibilidad para estructurar documentos.

3.3 Diferencias con HTML

El HTML se preocupa por formatear datos y para ello son las etiquetas que tiene el lenguaje, para formatear la información que se desea mostrar.

El XML se preocupa por estructurar la información que pretende almacenar. La estructura la marca la lógica propia de la información.

El desarrollo del HTML estuvo marcado por la competencia entre los distintos visores del mercado.

El desarrollo del XML está siendo llevado a cabo con rigor, siempre ajustado a lo que marca el estándar que desarrolla el W3C, sin seguir intereses particulares de empresas.

Procesar la información en HTML es inviable, por estar mezclada con los estilos y las etiquetas que formatean la información.

Esto se soluciona en XML ya que se puede procesar la información con mucha facilidad, porque todo está ordenado de una manera lógica. Así mismo, es fácil de aplicar un formato determinado a la información, de cara a que sea interpretado de la manera más clara posible por el usuario final. Se pueden utilizar hojas de estilo, a la forma que se hace con HTML.

3.4 XML bien formado

Como lenguaje de anotación, las sentencias en XML [26] consisten en una serie de etiquetas (llamadas *elementos*) con una serie de modificadores (llamados *atributos*). Las etiquetas pueden estar anidadas unas dentro de otras, pero toda etiqueta que se abra se tiene que cerrar, y siempre en el mismo orden. En caso de que un elemento no tenga pareja (por no tener ningún contenido dentro), se le denomina elemento *vacío* y se indica con un / al final. Los elementos se agrupan en *documentos*, tales como se puede ver en la siguiente

```
<?xml version="1.0" encoding='iso-8859-1' ?>

<micasa>

    <habitacion id='comedor'>

        <mueble>aparador</mueble>

        <mueble>sofá</mueble>

        <puerta a='balcón' />

    </habitacion>

    </micasa>
```

Figura 3.1. Ejemplo de documento XML.

Todos los documentos XML deben estar **bien formados**, y este es el requisito mínimo que deben cumplir los documentos. Esto significa que se debe de cumplir lo siguiente:

- Si no se utiliza DTD (Declaración de Tipo de Documento) [27], el documento debe comenzar con un Declaración de Documento *Standalone* (aislado), tal como la que se ve en la primera línea (Tabla 3-I).
- Todas las etiquetas deben estar equilibradas: esto es, todos los elementos que contengan datos de tipo carácter deben tener etiquetas de principio y fin
- Todos los valores de los atributos deben ir entrecomillados (el carácter comilla simple, puede utilizarse si el valor contiene caracteres comillas dobles, y viceversa): si necesitas ambos, utiliza `'` y `"`. Así es como se hace en el elemento `habitacion`
- Cualquier elemento VACÍO (p.e. aquellos que no tienen etiqueta final como ``, `<HR>`, y `
` y otros de HTML) deben terminar con `'/'` o debes hacerlos no VACÍOS añadiéndoles una etiqueta de fin, tal como se ve en el elemento `puerta` del ejemplo.
- No debe haber etiquetas aisladas (`<` ó `&`) en el texto (p.e. debe darse como `<` y `&`), y la secuencia `]]>` debe darse como `]]>`; si no ocurre esto como final de una sección marcada como CDATA;
- Los elementos deben anidar dentro de sí sus propiedades (no se deben sobreponer etiquetas, como en el resto de SGML);
- Los ficheros bien-formados sin-DTD pueden utilizar atributos en sus elementos, pero éstos deben ser todos del tipo CDATA, por defecto. El tipo CDATA (character DATA) son caracteres.
- Los nombres de las etiquetas pueden ser alfanuméricos, comenzando con una letra, e incluyendo los caracteres “-” y “:”, aunque este último tiene un significado especial.

3.5 Esquema de definición del XML

Un “Schema XML” es algo similar a un **DTD**, es decir, que define qué elementos puede contener un documento XML, cómo están organizados, y qué atributos y de qué tipo pueden ser sus elementos.

Las ventajas de los Schemas con respecto a los DTDs son:

- Usan sintaxis de XML, al contrario que los DTDs,
- Permiten especificar los tipos de datos.
- Son extensibles.

Por ejemplo, un *schema* nos permite definir el tipo del contenido de un elemento o de un atributo, y especificar si debe ser un número entero, o una cadena de texto, o una fecha, etc. Los DTDs no permiten esta funcionalidad.

Un ejemplo de un documento XML con el que trabajará nuestro sistema es el que se muestra en la Figura 3.2. En él, se muestran en primer lugar, los datos personales de un candidato así como la estructura de las categorías y subcategorías, junto con los valores de los criterios. Los distintos campos (Categoría, criterio,..) están extraídos del baremo de contratación, y son los que vamos a usar para recoger los datos de los aspirantes.

```
<?xml version="1.0"?>
<Aspirante xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi: SchemaLocation ="file:///schema/schemaDef.xsd">

<DatosPersonales>
  <DNI> 9999999F </DNI>
  <Apellido1> Ejemplo </Apellido1>
  <Apellido2>Ejemplo</Apellido2>
  <Nombre> Aspirante </Nombre>
  <FechaNacimiento>1983-01-01</FechaNacimiento>
  [...]
</DatosPersonales>
<Categ idCategoria="I." nombreCategoria="Formacion Academica">
  <Subcategoria idSubcategoria="I.1." nombreSubcateg="Titulacion">
```

```

        <Criterio idCriterio="I.1.1." nombreCriterio="Nota Media de las asignaturas
            cursadas en la titulacion A 1, N:2 SB:3 MH:4">
            <valorCriterio> 3 </valorCriterio>
        </Criterio>
        <Criterio idCriterio="I.1.2." nombreCriterio="Tesina">
            <valorCriterio> 1 </valorCriterio>
        </Criterio>
    </Subcategoria>
    <Subcategoria idSubcategoria="I.4." nombreSubcateg="Otras titulaciones universitarias
        afines distitntas a la exigida en la convocatoria">
        <Criterio idCriterio="I.4.1." nombreCriterio="Titulaciones
Universitarias Superiores">
            <valorCriterio>1</valorCriterio>

            </Criterio>
        <Criterio idCriterio="I.4.2." nombreCriterio="Titulaciones Universitarias
de Grado Medio">
            <valorCriterio> 1</valorCriterio>
        </Criterio>
    </Subcategoria>
    [ ... ]
</Categ>
</Aspirante>

```

Figura 3.2. Ejemplo XML. Candidato según baremo UJA

El esquema (usando Schema) que debe de cumplir este XML para que sea válido para nuestro sistema, se puede ver detallado en el Anexo III.

3.6 Usando un formulario para recoger datos en XML. InfoPath.

Según la página oficial, se define InfoPath© [28] como una herramienta para crear formularios y recopilar datos que ayuda a las organizaciones a simplificar los procesos empresariales. InfoPath está pensada para usuarios profesionales avanzados y para programadores, según el tipo de solución basada en formularios que la organización necesite.

Los usuarios profesionales avanzados pueden usar InfoPath sin la necesidad de escribir código para diseñar formularios electrónicos complejos que permitan recopilar información de forma rápida y rentable.

Los programadores pueden crear formularios avanzados para procesos de negocio departamentales y empresariales sin apenas recurrir al uso de código o no usándolo en absoluto.

Una característica que ha sido usada de InfoPath, es que podemos usar un esquema XML, o un documento XML sencillamente, para construir el formulario, y obtener de esa manera, los datos ya formateados con las marcas XML, cosa que simplifica mucho el proceso de lectura y procesamiento de datos del fichero.

Por estas razones, InfoPath se ha usado para contar con una herramienta para recoger los datos de manera estructurada, en la que solamente haya que rellenar los campos (Figura 3.3) y obtengamos el documento XML bien formado y válido. Así, evitamos la ardua tarea de rellenar los campos del XML a mano, con el consiguiente riesgo, de introducir datos inválidos, o alterar la forma que debe tener el documento.

UNIVERSIDAD DE JAÉN
Servicio de Personal y Organización Docente

INSTANCIA-CURRICULUM ACADÉMICO
PARA PARTICIPACIÓN EN CONCURSOS PÚBLICO DE ACCESO PARA CUERPOS DOCENTES UNIVERSITARIOS

Tipo Plaza:
Profesor Adjunto

DATOS PERSONALES

DNI	77338594j
Apellido 1	Barrio
Apellido 2	Muñoz
Nombre	Pedro
Fecha Nacimiento	01/01/1983
Localidad	
Provincia Nacimiento	
Nacionalidad	
Domicilio	
E Mail	
Municipio	
Provincia	
Código Postal	
Teléfono	1.

RELACIÓN DE MÉRITOS QUE DESEA ALEGAR

Categoria	Nombre Categoría	Subcategoría
I.	Formacion Academica	I.1.
		Titulacion
		Criterio
		Nombre Criterio
		I.1.1.
		Nota Media de las asignaturas cursadas en la titulacion A:1, N:2 SB:3 MH:4
		10
		I.1.2.
		Tesina
		si
		I.2.
		Premios fin de carrera
		Criterio
		Nombre Criterio
		I.2.1.
		Premios Nacionales
		10
		I.2.2.
		Premios Internacionales
		20
		I.3.
		Alumno colaborador de un Departamento que incluya el área convocante, becario Erasmus, Intercampus,...
		Criterio
		Nombre Criterio
		I.3.1.
		Meses de becario de alumno colaborador de Departamento
		10
		I.3.2.
		Meses de becario Erasmus, Intercampus
		10
		I.4.
		Otras titulaciones universitarias afines distintas a la exigida en la convocatoria
		Criterio
		Nombre Criterio
		I.4.1.
		Titulaciones Universitarias Superiores
		2
		I.4.2.
		Titulaciones Universitarias de Grado Medio

Figura 3.3 Ejemplo de Documento InfoPath

Capítulo 4:

Diseño, implementación y pruebas

Una vez que hemos presentado el modelo de decisión multicriterio para la selección de personal y el esquema XML, en este capítulo se describirán las fases de la Ingeniería del Software, al estar en un proyecto de desarrollo software.

No existe una definición única y estandarizada para la **Ingeniería del Software**. Pese a ello, las dos que pasamos a exponer a continuación resultarán perfectamente válidas para nuestro cometido.

- La Ingeniería del Software consiste en la construcción de software de calidad con un presupuesto limitado y un plazo de entrega en contextos de cambio continuo [29].
- Es el establecimiento y uso de principios y métodos firmes de ingeniería, para obtener software económico que sea fiable y funcione de manera eficiente en máquinas reales.
- Además, requiere llevar a cabo numerosas actividades, las cuales se pueden agrupar en etapas, o también llamadas fases, que se detallan a continuación [29]:
 - **Especificación de Requerimientos:** Se obtiene un conjunto de propiedades o restricciones definidas con precisión, que un sistema software debe satisfacer.
 - **Análisis de los casos de uso:** Se obtiene un modelo del sistema correcto, completo, consistente, claro y verificable.
 - **Diseño del Sistema:** Se definen los objetivos del proyecto y las estrategias a seguir para conseguirlos.
 - **Implementación:** Se traduce el modelo a código fuente, pudiendo ser la parte más obvia del trabajo de la ingeniería del software. La complejidad y la duración de esta etapa está íntimamente relacionada al o lenguajes de programación utilizados, al diseño previamente realizado.
 - **Prueba:** Verificar y validar el sistema. Básicamente consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación del problema. Para ello se definen una serie de pruebas, que serán comprobadas en el sistema.

4.1 Especificación de los requerimientos

Los requerimientos recogen el conjunto de propiedades o restricciones definidas con precisión, que un sistema software debe satisfacer.

Es importante resaltar las propiedades deseables de un requerimiento:

- **Necesario:** Un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados.
- **Conciso:** Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
- **Completo:** Un requerimiento está completo si no necesita ampliar detalles en su redacción es decir, si se proporciona la información suficiente para su comprensión.
- **Consistente:** Un requerimiento es consistente si no es contradictorio con otro requerimiento.
- **No ambiguo:** Un requerimiento no es ambiguo cuando tiene una sola interpretación.
- **Verificable:** Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

Esta fase del análisis representa el primer acercamiento a las necesidades del usuario, que deberemos conseguir gracias a las entrevistas o informes de la empresa. En caso de aplicaciones dirigidas a un colectivo deberemos definir perfectamente el perfil de usuario y buscar un elemento representativo.

Este paso sentará las bases de las siguientes fases del análisis y posteriormente del diseño. Es muy importante definir todos los requerimientos tanto funcionales como no funcionales de manera que sean los más completos posibles, evitando en cada momento

ambigüedades o dobles interpretaciones. Tenemos que tener en cuenta que un cambio más adelante implicará modificaciones sobre todos los pasos posteriores.

El resultado de esta fase debe ser plasmado en un documento inicial que deberán firmar tanto el desarrollador como el cliente. Esto debe vincular a ambas partes a cumplir con lo pactado. Al fin y al cabo este documento determina exactamente lo que el cliente está pagando y lo que el desarrollador se ha comprometido a satisfacer.

Es lógico que los requerimientos evolucionen a lo largo del tiempo. En caso de que llegue a ser necesario modificar este documento será indispensable que ambas partes acepten el cambio y que asuman las consecuencias del remodelado. Por supuesto, es importante captar los requerimientos básicos en un primer lugar y dejar estos cambios para aquellos secundarios que no vayan a modificar radicalmente el modelo desarrollado.

En nuestro caso no nos encontramos ante un proyecto comercial sino ante un proyecto académico por lo que el propósito es conocido desde el mismo momento de la concepción del mismo:

“Creación de un sistema de ayuda a la selección de personal, para el PDI de la Universidad de Jaén, basado en un modelo de toma de decisión multicriterio, y con interfaz WEB”

Una vez determinado el propósito, el siguiente paso consiste en determinar los requerimientos, que dicho proyecto ha de satisfacer.

Existen dos tipos de requerimientos: funcionales y no funcionales, que se pueden definir como:

- **Requerimientos funcionales:** Aquellos que se refieren expresamente al funcionamiento del sistema.
- **Requerimientos no funcionales:** Son todos aquellos requerimientos no referidos al estricto funcionamiento del sistema, sino a otros factores externos.

En los dos epígrafes siguientes, se pasa a detallar cada uno de ellos.

4.1.1 Requerimientos funcionales

Estos requerimientos se refieren a la funcionalidad que debe proporcionar el programa. Son el punto de partida del análisis y diseño y es imprescindible una completa definición de ellos.

En nuestro caso, los requerimientos funcionales, es decir, la funcionalidad que se espera de nuestro sistema, van a ser los siguientes:

Todos los requerimientos de nuestro sistema, se refieren al administrador, ya que va a ser el único usuario que va a interactuar con el mismo.

a) RF-1: Identificación en el sistema.

Al tratarse de un sistema web, para no permitir accesos indeseados, contaremos con un mecanismo de validación de usuario.

b) RF-2: Preprocesado de los expedientes de los aspirantes.

Los datos que figuran en los expedientes han sido recogidos en un sistema externo, por lo que antes de incorporarlos a nuestro sistema, hemos de hacerle un preprocesado para poder utilizarlos para los cálculos.

El tratamiento de los datos consistirá en extraer la información que es útil para el sistema, que trabajará con ficheros formateados en lenguaje XML.

c) RF-3: Gestión de evaluaciones.

La gestión de evaluaciones se divide en:

- **Alta** de una nueva evaluación. Se dará la posibilidad de dar de alta evaluaciones, indicando un nombre identificativo de la misma. El proceso de creación de un alta estaría terminado con la asignación del formulario de evaluación, así como de los aspirantes y las plazas que van a formar parte de la evaluación. Esta funcionalidad se puede hacer a posteriori, a través de la modificación de la evaluación.

- **Recuperación** de evaluaciones. El sistema permitirá tener acceso a evaluaciones ya finalizadas, es decir, en la que ya se han dado todos los pasos para evaluar a los candidatos. Además, de las finalizadas, se podrá recuperar una evaluación aunque no esté cerrada, cuando sea necesaria alguna modificación.
- **Modificación** de una evaluación. El sistema permitirá cambiar el formulario asignado a la evaluación. Las funciones de añadir y modificar, estarán también presentes para las plazas. Se podrá añadir, modificar y borrar las plazas asignadas a la evaluación. En esta funcionalidad, se incluye además la posibilidad de cambiar el nombre identificativo de la evaluación.

d) **RF-4: Gestión del formulario de evaluación**

La funcionalidad que recoge este requisito es la de:

- **Alta** de un nuevo formulario. El sistema permite el alta de nuevos formularios. Para ello, se requerirá conocer el nombre que se le quiere asignar al mismo. Para que un formulario esté totalmente creado, se necesita añadir los criterios que van a formar parte de él, así como los pesos con los que se pondera cada uno de los mismos. Esta funcionalidad se podrá hacer a posteriori a través de la funcionalidad de *Modificación*.
- **Recuperación** de un formulario. Esta funcionalidad posibilita incluir un formulario anteriormente usado en una nueva evaluación.
- **Modificación** de formulario. Dará la posibilidad de modificar los criterios actuales, (cambiarles el nombre, añadir nuevos, borrarlos), o el peso que se le asignó.

e) **RF-5: Gestión de las plazas ofertadas.**

Para la realización de la evaluación, necesitamos conocer las plazas que se han ofertado, por lo que tendremos que tener la funcionalidad de:

- **Alta** de plazas: Se da la posibilidad de añadir una plaza, indicando el nombre que va a tener la misma. Para que la creación de la plaza sea completa, se añadirán las categorías que la forman, así como el peso que va a tener la categoría dentro de la plaza creada. Esta funcionalidad se podrá realizar posteriormente a través de la modificación.
 - **Recuperación** de plazas. Se dará la posibilidad de consultar una plaza para conocer qué categorías la forman, y el peso que indica cada una.
 - **Modificación**: Tanto las categorías que forman una plaza, como los pesos de cada categoría, podrán ser modificados dentro de cada una de las plazas.
 - **Eliminación**: se permitirá eliminar una plaza entera, o alguna de las categorías que la forman.
- f) **RF-6: Gestión de los elementos del modelo (criterios, categorías, subcategorías).**

Tendremos una colección de criterios, categorías y subcategorías, esquematizadas según la Baremación [4]. En cada uno de ellos, podremos dar altas, bajas y hacer modificaciones.

g) **RF-7: Cálculos y resultados.**

Una vez se tienen las alternativas (aspirantes), los criterios (calculados a partir de las valoraciones conseguidas tras recoger los datos de los aspirantes, y aplicarles el factor de ponderación correspondiente en cada una de las categorías), se procede a aplicar el TDMC y obtener los resultados para la plaza que indiquemos.

El resultado será una lista ordenada de los candidatos, con la puntuación obtenida. Además, se podrán visualizar gráficos, con dichas puntuaciones, de cada uno de los aspirantes, en cada una de las plazas ofertadas.

h) RF-8: Consultas sobre los candidatos.

Para que tengamos una visión clara de los datos y la información que ha presentado cada candidato, el sistema permite ver los datos personales, y los datos profesionales de cada uno. Esto hace, que el administrador no tenga que abrir el fichero para recuperar esa información, que le será útil en caso de que se quiera poner en contacto con la persona o personas que hayan sido seleccionadas para la plaza en cuestión.

4.1.2 Requerimientos no funcionales

Estos requerimientos no hacen referencia a ninguna funcionalidad, sino a propiedades que, de una forma u otra, pueden limitar el sistema. Normalmente son tan importantes como los funcionales y pueden ser críticos para la aceptación del sistema.

Vamos a describir aquellos que hemos considerado especialmente necesarios en esta aplicación:

- **Manipulación directa:** Se debe ofrecer una interfaz amigable que abstraiga la complejidad de los comandos internos del programa. La interacción con el sistema será incremental y cada paso sentará un estado base para el siguiente. Al usar una interfaz Web, con la que todo el mundo está familiarizado, cumplimos este requisito.
- **Compatibilidad.** A través del uso de un navegador web, se hace que sea independiente del sistema operativo, y poniendo cuidado en los distintos estándares que deben de cumplir los navegadores, se hace que sea compatible con cualquier navegador. Se seguirán los estándares que marca W₃C² (<http://www.w3c.es/>).

² El World Wide Web Consortium (W3C) es una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo.

- **Claridad.** Presentaremos la información de la manera más clara para que se puedan interpretar sin problemas. Para ello, se utilizarán gráficos fácilmente reconocibles.
- **Tiempo de respuesta bajo.** Es un aspecto en el que tendremos especial cuidado, ya que al hacer proceso de varios ficheros, puede ser que la velocidad de la aplicación se degrade. También tendremos especial cuidado en la aplicación de los algoritmos, e indicaremos al usuario, si el tiempo de respuesta va a ser más de lo normal, para que no crea que el sistema no está trabajando.

4.2 Análisis del sistema

Una vez conocidas las propiedades que debe cumplir y las restricciones a las que debe someterse el proyecto, llega el momento de analizar el sistema y crear un modelo del mismo que sea correcto, completo, consistente, claro y verificable. Para conseguir esto se estudiarán los perfiles de usuario, se crearán y definirán casos de uso en base a los requerimientos previamente obtenidos. Por último, se describirán ciertos escenarios de acción de dichos casos de uso.

4.2.1 Perfil de usuario

Este paso, lo vamos a realizar una vez obtenemos una versión preliminar de los requerimientos funcionales y no funcionales. Vamos a concretar quienes son los usuarios potenciales de nuestro sistema, así como sus características y unas conclusiones. Así conseguiremos adaptarnos a sus necesidades.

Las fuentes de datos que manejamos para el análisis, son los cuestionarios y entrevistas, observación de los usuarios reales realizando las tareas directamente, o consultando manuales y textos sobre el dominio del problema.

Una vez identificadas las categorías de usuarios que vamos a tener, obtenidas del análisis de los resultados de las fuentes de datos, pasamos a ver las características comunes. Las vamos a clasificar en cuatro grupos [30]:

- **Características psicológicas.** Motivación o actitud frente a los sistemas informáticos, etc.
- **Conocimiento y experiencia.** Habilidad escribiendo a máquina, experiencia en el dominio del problema, experiencia en el uso de sistemas informáticos, hábito de lectura, posibilidad de tener cursos de formación, nivel educativo, conocimiento de idiomas, etc.
- **Características del trabajo y las tareas.** Frecuencia de uso, si el ambiente es ruidoso, si el usuario experimenta interrupciones frecuentes a lo largo de su trabajo, si tiene las manos ocupadas mientras usa el sistema, etc.
- **Características físicas.** Capacidad visual disminuida, etc.

Pasaremos a describir a los usuarios típicos de nuestro sistema, a partir de los perfiles de usuario.

En nuestro caso, sólo existe un usuario, que va a ser el **administrador** del sistema.

Administrador:

En nuestro sistema, es el único tipo de usuario que va a interactuar con él, y el encargado de todo el manejo. A continuación se detallan las características que se han obtenido del análisis del perfil de usuario en nuestro caso:

- **Características psicológicas:** Se caracteriza por tener un nivel informático alto, su nivel de estudios suele ser medio-alto y suele tener un buen hábito de lectura.
- **Conocimiento y experiencia:** Conoce bien el ámbito de aplicación del sistema. Está acostumbrado al lenguaje técnico en el ámbito de la toma de decisiones, y de la selección de personal. Se le pueden presentar los errores y

los mensajes con detalle, ya que sabrá interpretarlos. Sabe esquematizar los componentes (criterios, subcategorías y categorías).

- **Características del trabajo y las tareas:** Las tareas se realizarán en un entorno poco ruidoso y tranquilo.
- **Características físicas:** No se precisa adaptación a ninguna característica física.

4.2.2 Casos de uso

Como parte final del análisis vamos a realizar los diagramas de casos de uso. Estos diagramas se encargarán de plasmar lo descrito en los requerimientos funcionales. Tendremos que desglosar cada requerimiento, indicando los pasos de comunicación entre el usuario y el programa.

El modelo de casos de uso, es una descripción completa de la funcionalidad del sistema y su entorno y se define como el conjunto de todos los casos de uso.

Los casos de uso, al ser parte del análisis nos ayudan a describir qué es lo que el sistema debe hacer. Es decir, describen un uso del sistema y cómo éste interactúa con el usuario.

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia el actor principal sobre el propio sistema.

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Se utilizan para ilustrar cómo reacciona una respuesta a eventos que se producen en el mismo.

Cada una de las partes que conforman el diagrama son:

- **Actores:** Se llama Actor a una entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Representan

papeles (roles) que pueden ser interpretados por personas, periféricos u otros sistemas en incluso, un entorno físico. En el caso de los seres humanos, se pueden ver a los actores como definiciones de rol, por lo que un mismo individuo puede corresponder a uno o más actores. En consecuencia, no necesariamente coinciden con usuarios, si no que cada usuario podría interpretar distintos roles, cada uno representado por un actor (Figura 4.11).

Ejemplo:

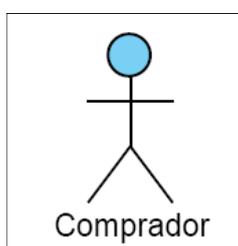


Figura 4.1. Ejemplo de Actor. Casos de Uso.

Suele suceder sin embargo, que es el sistema el que va a tener interés en el tiempo. Es frecuente encontrar que nuestros sistemas deben efectuar operaciones automáticas en determinados momentos; y siendo esto un requisito funcional obvio, resulta de interés desarrollar alguna forma de capturar dicho requisito en el modelo de casos de uso.

- **Relaciones de caso de uso:** Las tres relaciones principales entre los casos de uso son soportadas por el estándar UML, el cual describe notación gráfica.
 - **Inclusión:** Es una forma de interacción. Un caso de uso dado puede "incluir" otro. El primer caso de uso a menudo incluye el comportamiento del caso de uso final (subcaso). Esto es útil para extraer comportamientos verdaderamente comunes desde múltiples casos de uso a una descripción individual.

A un mismo caso de uso se le pueden aplicar múltiples relaciones de inclusión, y el mismo caso de uso proveedor puede

incluirse en múltiples casos base. Además, las inclusiones se pueden anidar (

Figura 4.22). Ejemplo:

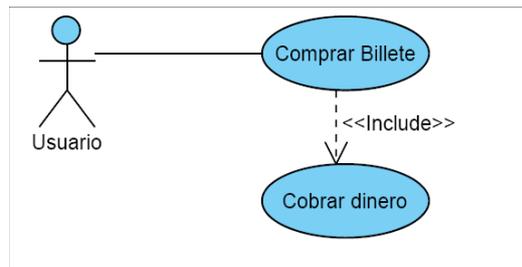


Figura 4.2. Ejemplo de relación Inclusión. Casos de Uso

- o Extensión: Es otra forma de interacción. Un caso de uso dado, puede *extender* a otro. Esta relación indica que el comportamiento del caso de uso extensión puede ser insertado en el caso de uso extendido bajo ciertas condiciones. La notación es una flecha rayada desde el caso de uso extensión al caso de uso extendido, con la etiqueta «extend». Esto puede ser útil para lidiar con casos especiales, o para acomodar nuevos requisitos durante el mantenimiento del sistema y su extensión (Figura 4.33). Ejemplo:

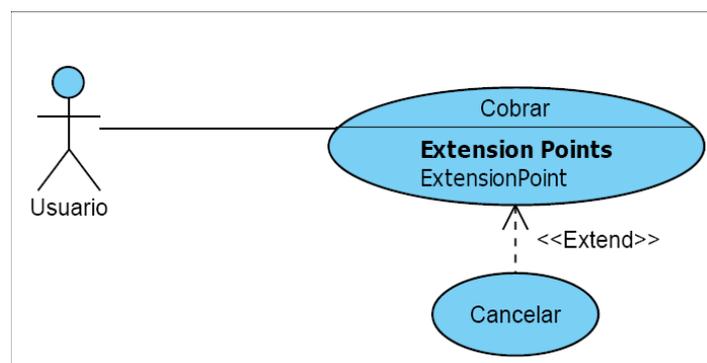


Figura 4.3. Ejemplo de Relación Extensión. Casos de Uso.

- **Generalización:** En la tercera forma de relaciones entre casos de uso, existe una relación generalización/especialización. La notación es una línea sólida terminada en un triángulo dibujado desde el caso de uso especializado al caso de uso general (Figura 4.44). Ejemplo:

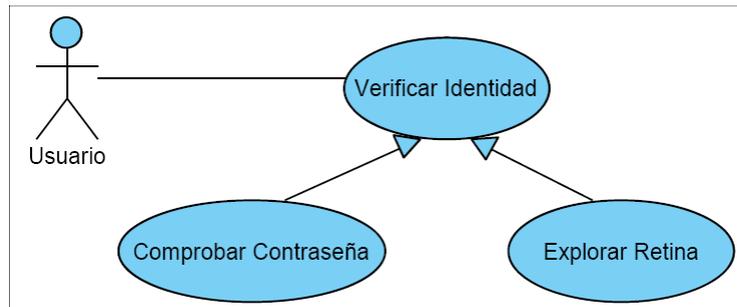


Figura 4.4. Relación de Generalización. Casos de Uso.

El hijo hereda los atributos, operaciones y secuencias de comportamiento del padre, pudiendo agregar atributos y operaciones propios. La capacidad de sustitución para los casos de uso significa que la secuencia de comportamiento de un caso de uso hijo tiene que incluir la secuencia de comportamiento de su padre.

- Otro elemento de los diagramas de casos de uso es el diagrama frontera, que separa a los elementos pertenecientes al sistema de los externos (Figura 4.55).

Ejemplo:

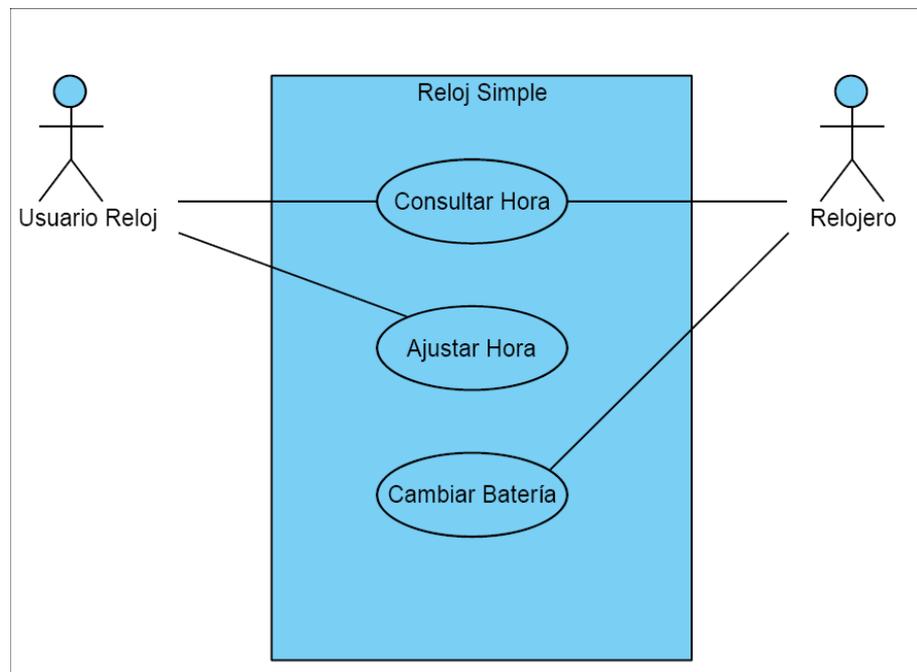


Figura 4.5. Ejemplo de Diagrama Frontera. Casos de Uso.

4.2.3 Casos de uso de nuestra aplicación

El primer paso, será plasmar los casos de uso que tienen entidad suficiente como para ser independientes en el diagrama frontera (Figura 4.66).

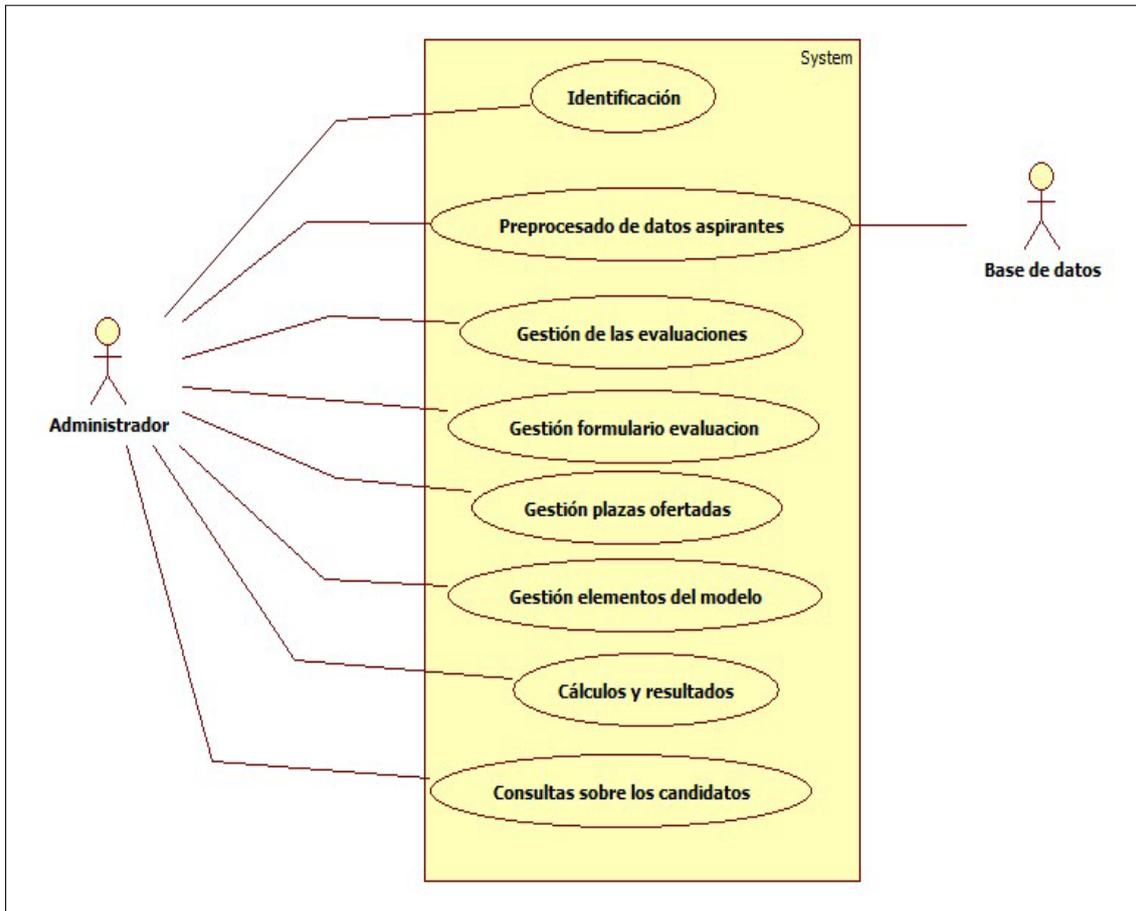


Figura 4.6. Diagrama Frontera del sistema

CU-Identificación

Actores:	Administrador
-----------------	---------------

Condición inicial:	Hay un usuario administrador que desea entrar al sistema.
---------------------------	---

Flujo de eventos:	<ol style="list-style-type: none"> 1. El sistema presenta la pantalla de bienvenida en la que el administrador deberá rellenar su nombre de usuario y contraseña. 2. El sistema comprueba los datos (E1), y permite que el administrador entre al sistema.(E2)
--------------------------	--

Condiciones de salida: El administrador queda identificado en el sistema

Excepciones

E-1.Error de comunicación con la base de datos que impide que el administrador pueda ser validado.

E-2.Administrador no reconocido por el sistema. No dado de alta, o contraseña o identificador inválidos

En nuestro caso, como hay casos de uso que no son lo suficientemente precisos en el diagrama frontera, o que necesitan una mayor explicación, pasaremos a describirlos en los posteriores diagramas.

CU-Preprocesado de los datos de los Aspirantes

Actores: Base de datos, Administrador

Condición inicial: A petición del administrador, se comprueban los nuevos ficheros de aspirantes que han llegado al sistema.

Flujo de eventos:

1. El administrador selecciona una evaluación, para la que se van a leer los datos de los ficheros de los aspirantes.
2. El sistema identifica a qué evaluación pertenece el fichero del aspirante a través de su identificador de fichero (E-1).
3. El sistema identifica la plaza a la que el aspirante se ha presentado por la información que obtiene del interior del fichero (E-2) (E-5).
4. Se recuperan los datos del aspirante (E-3), para conocer los criterios que ha puesto como mérito, y se recogen en la base de datos (E-5).

-
5. El aspirante queda registrado en la base de datos en la evaluación y en la plaza correspondiente, estando a disposición del sistema todos los datos (E-4) (E-5).

Condiciones de salida: El aspirante queda registrado en la base de datos en la evaluación y en la plaza a la que se ha presentado.

Excepciones

E-1. Formato del nombre del fichero inválido. No se procesa el fichero.

E-2. No se encuentra la plaza a la que el candidato ha optado. No se procesa el fichero.

E-3. No hay datos que recuperar correspondiente a los criterios. Los datos serán incompletos.

E-4. Los datos del fichero, no coinciden con los del formulario de evaluación. No se recuperan datos del fichero del aspirante.

E-5. Error en la comunicación al leer o escribir en la Base de Datos

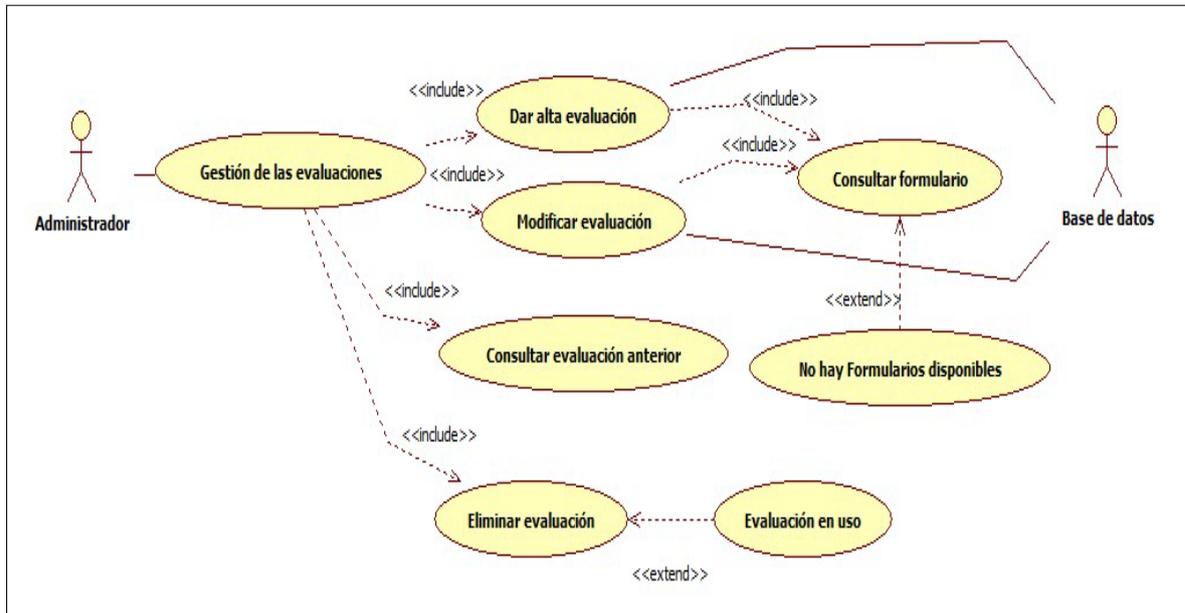


Figura 4.7. Diagrama Caso de uso Gestión de la Evaluación.

CU-Gestión de la Evaluación

Actores: Administrador.
Base de Datos. Encargado de recoger y almacenar los datos de los aspirantes.

Condición inicial: Hay un administrador identificado en el sistema que quiere realizar acciones sobre la Evaluación.

- Flujo de eventos:**
1. El Administrador selecciona la gestión de la evaluación y el sistema le presenta las opciones.
 2. Si el administrador va a **consultar una evaluación pasada**, se le muestran las evaluaciones que hay en el sistema, y podrá seleccionar una. (E-3)
 3. Si quiere **dar de alta** una nueva evaluación, se seguirán los pasos para definir el problema:

1. Para dar de alta un aspirante nuevo, se consulta la base de datos (E-3), en la que se sabe que previamente se han almacenado los datos del aspirante.
2. Podremos usar un formulario ya usado anteriormente a través de **Consultar formulario** (E-3).
4. Para **modificar una evaluación**, podremos elegir un formulario ya existente a través de **Consultar formulario (E-1)** (E-3).
Se podrá añadir nuevos aspirantes, a través del caso de uso de Preprocesado de datos aspirantes.
5. A través de **consultar evaluación anterior**, tenemos acceso a visualizar otras evaluaciones. (E-3)
6. A través de **eliminar evaluación**, se puede (E-3) borrar lo correspondiente a dicha evaluación (E-2).

Condiciones de salida: Los cambios quedan registrados en el sistema y almacenados en la base de datos.

Excepciones

E-1. No hay formularios disponibles. No se puede seguir con el alta, y no se podrá asignar ningún formulario a la evaluación.

E-2. Evaluación en uso. No es posible eliminarla, ya que da lugar a inconsistencia de los datos del sistema.

E-3. Error de comunicación con la base de datos.

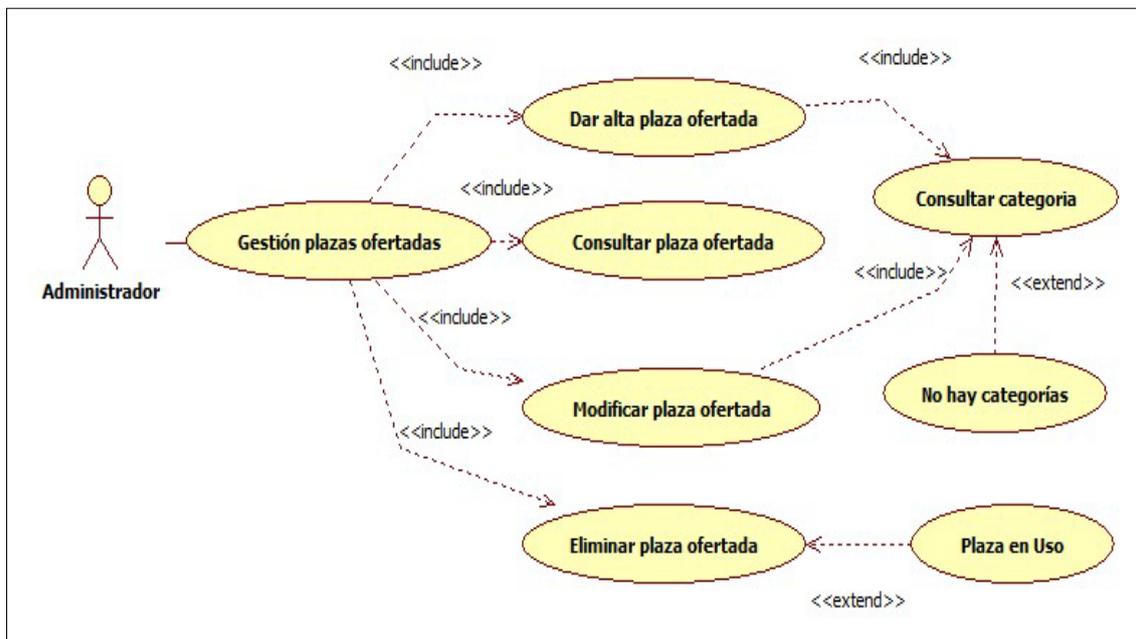


Figura 4.8. Diagrama Caso de uso Gestión de las plazas

CU-Gestión de las Plazas

Actores:	Administrador
Condición inicial:	Hay un administrador identificado en el sistema
Flujo de eventos:	<ol style="list-style-type: none"> 1. Al seleccionar el administrador la Gestión de las plazas, el sistema le presenta las opciones disponibles 2. Podrá dar de alta una plaza ofertada, seleccionando la categoría a través de consultar categoría (E-1), y asignarle un peso (E-3). 3. Podrá recuperar una plaza ofertada, obteniendo el peso asignado (E-3). 4. Modificar una plaza ofertada, seleccionando una nueva categoría para esa plaza, añadiendo más categorías, o quitando categorías. Así mismo, se permite cambiar el peso (E-3).

5. Eliminar una plaza ofertada en su totalidad (E-2), con todas las categorías que contenga (E-3).

Condiciones de salida: Los cambios quedan registrados en el sistema, y en la tabla de la base de datos correspondiente.

Excepciones **E-1.** No hay categorías. No se podrá recuperar ni asignar nuevas categorías al formulario.

E-2. No es posible eliminar la plaza, por estar siendo usada en algún formulario. Así se podrá mantener un histórico.

E-3. Error de comunicación con la base de datos.

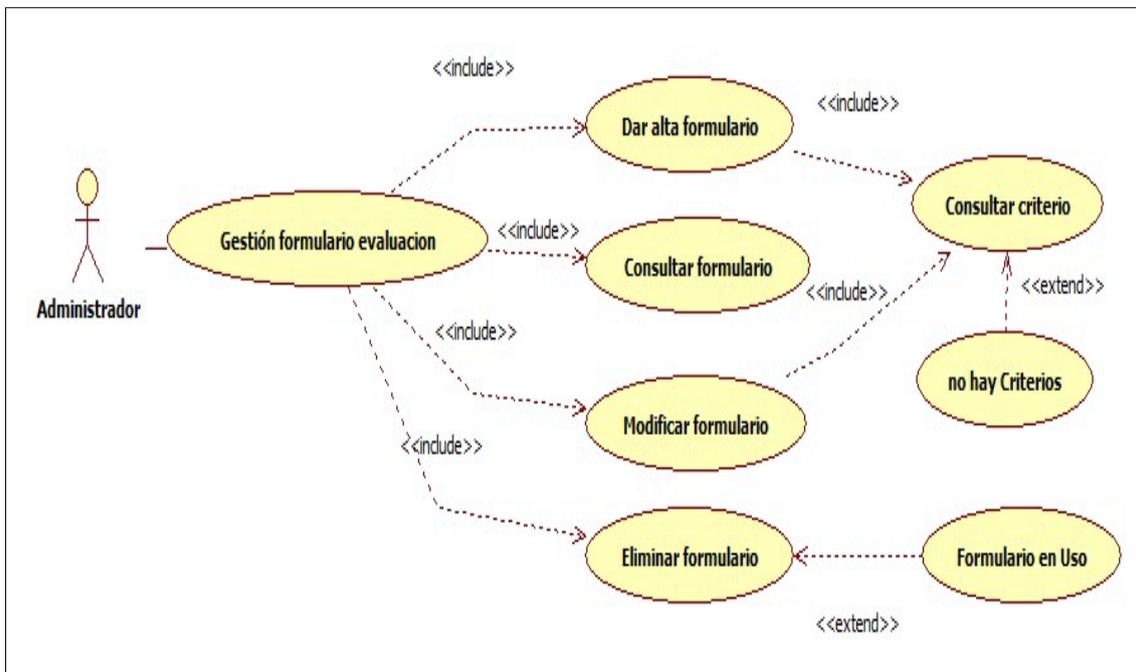


Figura 4.9. Diagrama Caso de uso Gestión de los Formularios

CU-Gestión de los Formularios

Actores:	Administrador
-----------------	---------------

Condición inicial:	Hay un administrador identificado en el sistema
---------------------------	---

Flujo de eventos:	<ol style="list-style-type: none">1. El administrador quiere gestionar los formularios y tras seleccionar la opción, el sistema le presenta las opciones.2. Las opciones son: Dar de alta un formulario nuevo, consultar un formulario anterior, modificar un formulario, o eliminarlo (E-1) (E-2) (E-3).3. El funcionamiento es el mismo que el descrito en el CU-Gestión de plazas
--------------------------	--

Condiciones de salida:	Los cambios quedan registrados en el sistema, y en la tabla de la base de datos correspondiente.
-------------------------------	--

Excepciones	<p>E-1. No hay Criterios. No se podrá recuperar ni asignar nuevos criterios al formulario de evaluación.</p> <p>E-2. No es posible eliminar el formulario, por estar siendo usado. Así se podrá mantener un histórico.</p> <p>E-3. Error de comunicación con la base de datos.</p>
--------------------	---

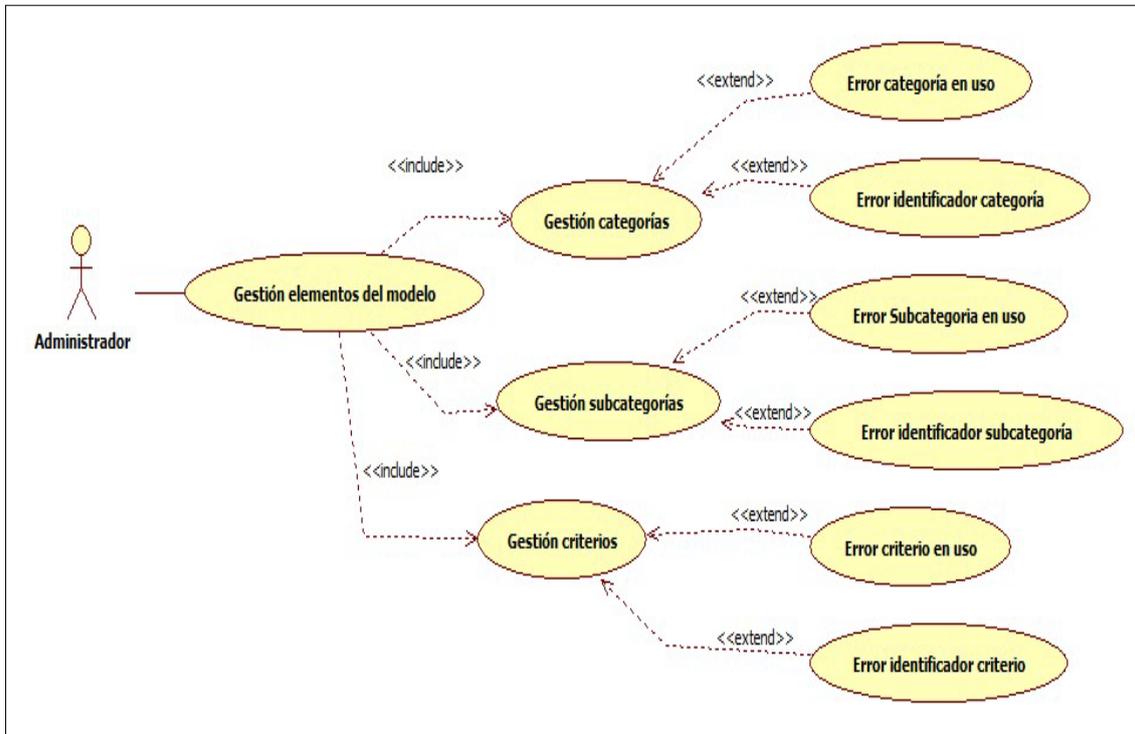


Figura 4.10. Diagrama caso de uso Gestión de los elementos del Modelo de Evaluación Diagrama

CU-Gestión del Modelo de Evaluación

Actores: Administrador

Condición inicial: Hay un administrador identificado en el sistema

- Flujo de eventos:**
1. El administrador quiere gestionar los elementos que van a formar parte del modelo de la evaluación. Serán los elementos que van a ser usados para la creación de las plazas y de la evaluación de los candidatos a través del formulario.
 2. Puede elegir entre las gestión de los criterios, de las categorías y de las subcategorías.
 3. Las opciones de los criterios serán dar de alta un nuevo criterio (E-1), que deberá ir en una subcategoría, que el administrador tendrá que escoger antes de añadirlo. Podrá

eliminar criterios (E-2) (E-7), y modificar el criterio (E-7).

4. Las opciones para las subcategorías, igual que para los criterios, pero el administrador solo tendrá que seleccionar la categoría en la que va la subcategoría. (E-3 y E-4)
5. Las opciones para las categorías, iguales que para subcategorías, excepto que en este caso, no se tendrá que escoger dónde se incluye la categoría, ya que es la del nivel más alto en la jerarquía. (E-5 y E-6).

Condiciones de salida: Los cambios quedan registrados en el sistema.

Excepciones

E-1. Error identificador categoría. Al dar de alta una categoría, el identificador no cumple el formato. "L." siendo L una letra. El administrador será avisado de este error, y del formato a seguir.

E-2. Error categoría en uso. La categoría ya está siendo usada en la evaluación. No es posible eliminarla para evitar incongruencias y mantener el histórico. El administrador será avisado de este error.

E-3. Error subcategoría en uso. . La subcategoría ya está siendo usada en la evaluación. No es posible eliminarla para evitar incongruencias y mantener el histórico. El administrador será avisado de este error.

E-4. Error identificador Subcategoría. Al dar de alta una subcategoría, el identificador no cumple el formato. "L.D." siendo L una letra y D un dígito. El administrador será avisado de este error, y del formato a seguir.

E-5. Error criterio en uso. El criterio ya está siendo usado en la evaluación. No es posible eliminarlo para evitar

incongruencias y mantener el histórico. El administrador será avisado de este error.

E-6. Error identificador criterio. Al dar de alta una categoría, el identificador no cumple el formato. “L.D.D.” siendo L una letra y D un dígito. El administrador será avisado de este error, y del formato a seguir.

E-7. Error de comunicación con la base de datos.

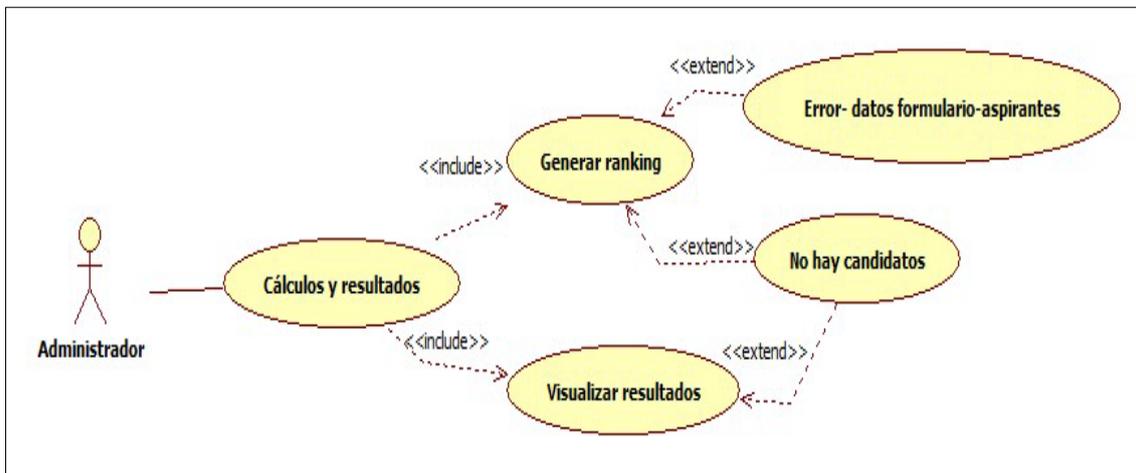


Figura 4.11. Diagrama caso de uso. Resultados

CU-Resultados

Actores: Administrador

Condición inicial: Hay un administrador identificado en el sistema

- Flujo de eventos:**
1. El administrador va a la opción de cálculos y resultados.
 2. El sistema le presenta las opciones de generar el ranking, o de visualizar los resultados.
 3. Para generar el ranking, verá de los candidatos que están en la evaluación actual (E-3), y posteriormente, el algoritmo a

utilizar para el cálculo (E-1).

4. Podrá elegir visualizar los resultados, bien a modo de ranking, con los nombres de los candidatos, o bien a modo de gráfico comparativo (E-2).

Condiciones de salida: Se realizan los cálculos para generar el ranking. Se quedan almacenados en la base de datos.

Excepciones E-1. Los datos de los aspirantes no coinciden con los del formulario de evaluación. No es posible realizar cálculos al no disponer de estos datos.

E-2. No hay candidatos disponibles para la evaluación. No se podrán mostrar datos de valoraciones, ni ranking.

E-3. Error de comunicación con la base de datos.

4.2.4 Escenarios

Un caso de uso es una representación abstracta, una abstracción, de una funcionalidad del sistema a realizar. La representación concreta de un caso de uso se realiza mediante la creación de uno o más escenarios que muestren todas las interacciones posibles entre el sistema y sus usuarios.

Los escenarios son historias ficticias que describen posibles interacciones persona-aplicación Web. Permiten a los diseñadores anticiparse a los problemas. Aunque son historias ficticias deben hacerse lo más detalladas posibles, así por ejemplo, los personajes deben tener nombres, motivaciones para usar la interfaz, deben encontrarse en entornos reales con las restricciones que ello conlleva, etc. De esta manera, se facilita a los diseñadores la discusión sobre la interfaz ya que a las personas nos cuesta más trabajo discutir sobre una situación abstracta.

Esta forma de proceder fuerza a los diseñadores a considerar el rango de usuarios que va a usar el sistema y el rango de actividades por las que lo van a usar. Los escenarios permiten hacer diferentes combinaciones de usuarios y actividades de forma que se tengan en cuenta todas las posibilidades.

Un escenario está formado por los siguientes elementos:

- Un nombre único y unívoco
- Una descripción
- Los actores participantes
- El flujo de eventos

Como se ha indicado, para cada caso de uso puede haber varios escenarios. Para nuestro proyecto se han creado y descrito una cantidad importante de casos de uso. Por lo que, no vamos a definir todos los escenarios de cada uno de ellos sino que, vamos a definir unos pocos que puedan servir como ejemplo de las principales funcionalidades que el sistema va a desarrollar:

Escenario 1: Modificación del Modelo de Evaluación

Nombre: Modificar el nombre de un criterio en la evaluación.

Descripción: El administrador, recibe un nuevo documento de evaluación y debe añadir en la descripción del criterio I.3.2, la beca Faro para la evaluación *2012-junio*.

Actores participantes: Administrador.

Flujo de eventos:

1. El administrador *admin* entra al sistema.
2. Al entrar, elige la evaluación sobre la que desea trabajar. En este caso, la de 2012-junio.
3. Elige del menú la opción de la *Gestión del modelo* de Evaluación.
4. Sabe que tiene que cambiar la denominación del criterio I.3.2, en el que se ha añadido, la beca Faro, como beca que se va a baremar.
5. Con la denominación del criterio, se va a la Categoría I, pulsa *ver subcategorías*.
6. Elige la subcategoría I.3., y elige “*Ver criterios*”.
7. Elige el criterio I.3.2, y escoge “*Modificar*”.
8. Modifica el nombre del campo, escribiendo *Faro* y elige “*Guardar*”.
9. El sistema, recoge el cambio en el nombre de ese criterio.

Escenario 2: Creación del Formulario

Nombre: Añadir dos criterios y eliminar uno del formulario de evaluación

Descripción: El administrador, va a añadir el criterio I.3.2 con nombre “Meses de becario Erasmus, Intercampus”, con peso 0.9 y va a borrar el correspondiente a II.3.2 “Créditos impartidos en cursos de formación”, ya existente en el formulario 2012-junio.

Actores participantes: Administrador.

Flujo de eventos:

1. El administrador *admin* entra al sistema.
2. Al entrar, elige la evaluación sobre la que desea trabajar. En este caso, la correspondiente a *2012-junio*.
3. Elige del menú la opción de la *Gestión del Formulario de Evaluación*.
4. Sabe que tiene que añadir el criterio “*I.3.2 Créditos impartidos en cursos de formación*” con peso *0.9*.
5. Elige de la lista el criterio.
6. Escribe el peso *0.9*, y le da a Añadir.
7. En la lista, aparece el nuevo criterio, con el peso indicado.
8. El administrador pulsa en la lista inferior, *eliminar y* borra el criterio *II.3.2*, que ya no se va a usar.
9. Desaparece el criterio *II.3.2* de la lista.
10. Los cambios quedan grabados en el sistema.

Escenario 3: Gestión de la Evaluación. Alta de una plaza

Nombre: Gestión de la evaluación. Alta de una plaza

Descripción: El administrador, quiere dar de alta una nueva plaza (profesor sustituto) para la evaluación Curso 2012.

Actores participantes: Administrador.

Flujo de eventos:

1. El administrador *admin* entra al sistema.
2. Va al apartado de “*Gestión del formulario*” de evaluación.
3. Selecciona la evaluación “*Curso 2012*” y cambia a esa evaluación.
4. Elige la opción de crear nueva plaza, y le asigna el nombre de “*profesor sustituto*”.
5. Para terminar el proceso de alta, selecciona las categoría “*I. Formación Académica*”, le da el peso 0,9 y la *Añade*.
6. Repite el paso 5, con “*II. Actividad docente*” con un peso de 0,5. Y con “*IV. Experiencia profesional de los últimos 12 años*”.
7. El sistema recoge los cambios producidos y registra la nueva plaza de “*profesor sustituto*”.

Escenario 4: Recuperación de los aspirantes de una evaluación pasada

Nombre: Recuperar los aspirantes de una evaluación pasada

Descripción: El administrador, quiere ver los aspirantes que formaron parte de la evaluación 2011-junio, en la plaza de profesor titular, para notificar al aspirante Roberto que ha sido seleccionado.

Actores participantes: Administrador.

Flujo de eventos:

1. El administrador *admin* entra al sistema.
2. Al entrar, elige la evaluación sobre la que desea trabajar. En este caso, 2011-junio.
3. Elige del menú la opción de “*Aspirantes*”.
4. Selecciona la plaza de profesor titular, ya que sabe que ahí se encuentra el aspirante.
5. Los aspirantes se pueden ir pasando con las flechas de navegación.
6. Al llegar al aspirante Roberto con el que quiere ponerse en contacto, puede ver que está el número de teléfono 6xx xxx xxx y el correo electrónico Roberto@roberto.com que facilitó en su formulario de registro.

Escenario 5: Evaluación de los candidatos

Nombre: Evaluación de los candidatos

Descripción: El administrador, quiere hacer los cálculos para cerrar la evaluación de 2012-junio.

Actores participantes: Administrador.

Flujo de eventos:

1. El administrador *admin* entra en el sistema.
2. Selecciona la evaluación 2012-Junio.
3. El administrador va al apartado de Evaluación, y tras seleccionar la plaza de *Profesor titular*, para la que quiere realizar la evaluación,
4. El sistema le muestra los candidatos que ha recogido para esa plaza.
5. El administrador refresca para que el sistema compruebe si se han introducido nuevos ficheros de aspirantes en el sistema, correspondientes a esa evaluación, y para esa plaza.
6. El administrador pulsa el botón calcular.
7. El sistema le indica que se han realizado los cálculos y que tendrá los resultados disponibles en el apartado de *Resultados*.
8. Se muestra el apartado de *Resultados* en el que están las puntuaciones de los aspirantes y el Ranking ordenado según la puntuación obtenida en cada uno de los apartados baremados.

Escenario 6: Visualización de los resultados

Nombre: Visualización de resultados

Descripción: El administrador, quiere comparar las gráficas de dos aspirantes que han formado parte del proceso de selección en el 2012-junio. El administrador deberá primero proceder a realizarlos para visualizar las gráficas.

Actores participantes: Administrador.

Flujo de eventos:

1. El administrador *admin* entra al sistema.
2. Al entrar, elige la evaluación sobre la que desea trabajar. En este caso, la 2012-junio.
3. Al entrar, elige del menú la opción de *Resultados*.
4. El sistema le indica, que no ha hecho todavía los cálculos, por lo que no hay gráficas para mostrar.
5. El administrador va a la pestaña *Evaluación*, y ve que para esa evaluación, se han recogido los datos de los aspirantes Antonio Pérez, Isabel García, e Irene López.
6. El administrador realiza la evaluación con el botón "*calcular*".
7. El sistema le indica que se han realizado los cálculos.
8. El administrador pasa al apartado de *Resultados*.
9. El sistema le muestra los 3 aspirantes.
10. El administrador selecciona Antonio Pérez e Isabel García, y le da a "*ver Gráficas*".
11. El sistema le muestra la gráfica de barras con las puntuaciones obtenidas por Antonio Pérez e Isabel García.

4.3 Diseño

- **Diagrama de clases:** muestran la estructura del sistema.
- **Diseño de los datos:** muestra la estructura de los datos.
- **Diseño de la interfaz:** se define la apariencia visual de la aplicación.

4.3.1 Diagrama de clases.

Los diagramas de clases se utilizan para mostrar la estructura estática del sistema modelado. Pueden estar contenidos por clases, interfaces, paquetes, relaciones e incluso instancias, como objetos o enlaces.

Son una potente herramienta de diseño, ayudando a los desarrolladores a planificar y establecer la estructura del sistema y subsistema antes de escribir ningún código. Esto permite asegurar que el sistema está bien diseñado desde el principio.

Son utilizados en la fase de diseño prácticamente en la totalidad de los sistemas que utilizan UML para su modelado.

Los diagramas de clases tienen los siguientes componentes:

- Clases: son los componentes fundamentales de los diagramas de clase. Su notación general es un rectángulo dividido en tres secciones, mostrando la primera el nombre de la clase, la siguiente los atributos y la última las operaciones.
- Relaciones: es una conexión semántica entre elementos. Existen cuatro tipos principales de relaciones:
 - **Generalización:** es una relación de especialización.
 - **Asociación:** es una relación estructural. Existen dos subtipos, la agregación y la composición.
 - **Realización:** es una relación contractual, en la cual una clase especifica un contrato que otra clase garantiza que cumplirá (por ejemplo, una interface).
 - **Dependencia:** Es una relación de uso.

Una vez descritos brevemente los componentes de un diagrama de clases, vamos a proceder a analizar los propios del sistema.

Diagrama completo de clases

Antes de mostrar el diagrama de clases, vemos apropiado explicar como va a ser el funcionamiento o la lógica de negocio del sistema para ver más claramente el paso de información o comunicación entre las distintas capas diseñadas. A la hora de diseñar una aplicación con una interfaz gráfica de usuario es importante seguir el esquema de diseño Modelo-Vista-Controlador (MVC). En este esquema se definen tres roles bien diferenciados, como se puede ver en la Figura 4.123.

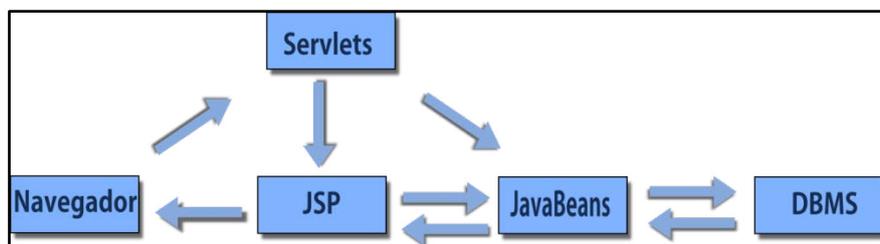


Figura 4.13 Esquema de la Arquitectura MVC

- **Modelo:** el modelo es la representación de la información de un problema.
- **Vista:** una vista es una posible visualización de la información contenida en un modelo. Un modelo puede tener varias vistas definidas.
- **Controlador:** el controlador se encarga de coordinar la interacción entre las vistas y los modelos. Cada vez que un modelo cambia internamente, actualiza sus vistas.

El mensaje más importante que proporciona este esquema de diseño es que siempre hay que separar la representación del problema con la interfaz de usuario u otra forma de visualización de la información del modelo (informe impreso, gráfico, etc.).

A continuación se presenta el diagrama simplificado de la estructura de paquetes de nuestra aplicación. Dada la complejidad, se ha preferido la simplificación a nivel de paquetes, para dar una idea general del modelo seguido (ver Figura 4.14).

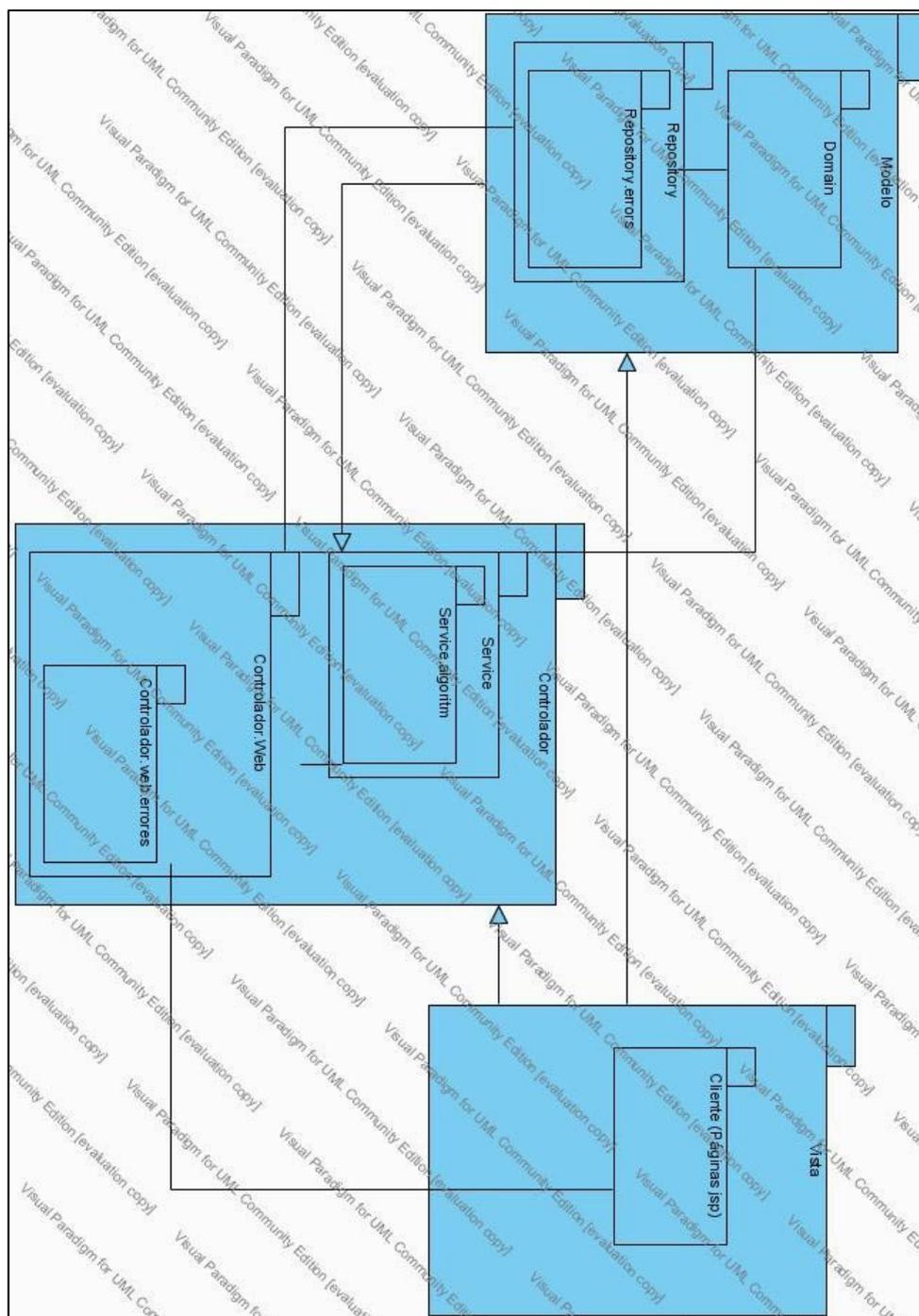


Figura 4.14 Diagrama de clases simplificado del sistema.

A continuación, se detallan el paquete del Modelo (Figura 4.15), del controlador Web (Figura 4.16) y un ejemplo de la comunicación entre los distintos paquetes. (Figura 4.17)

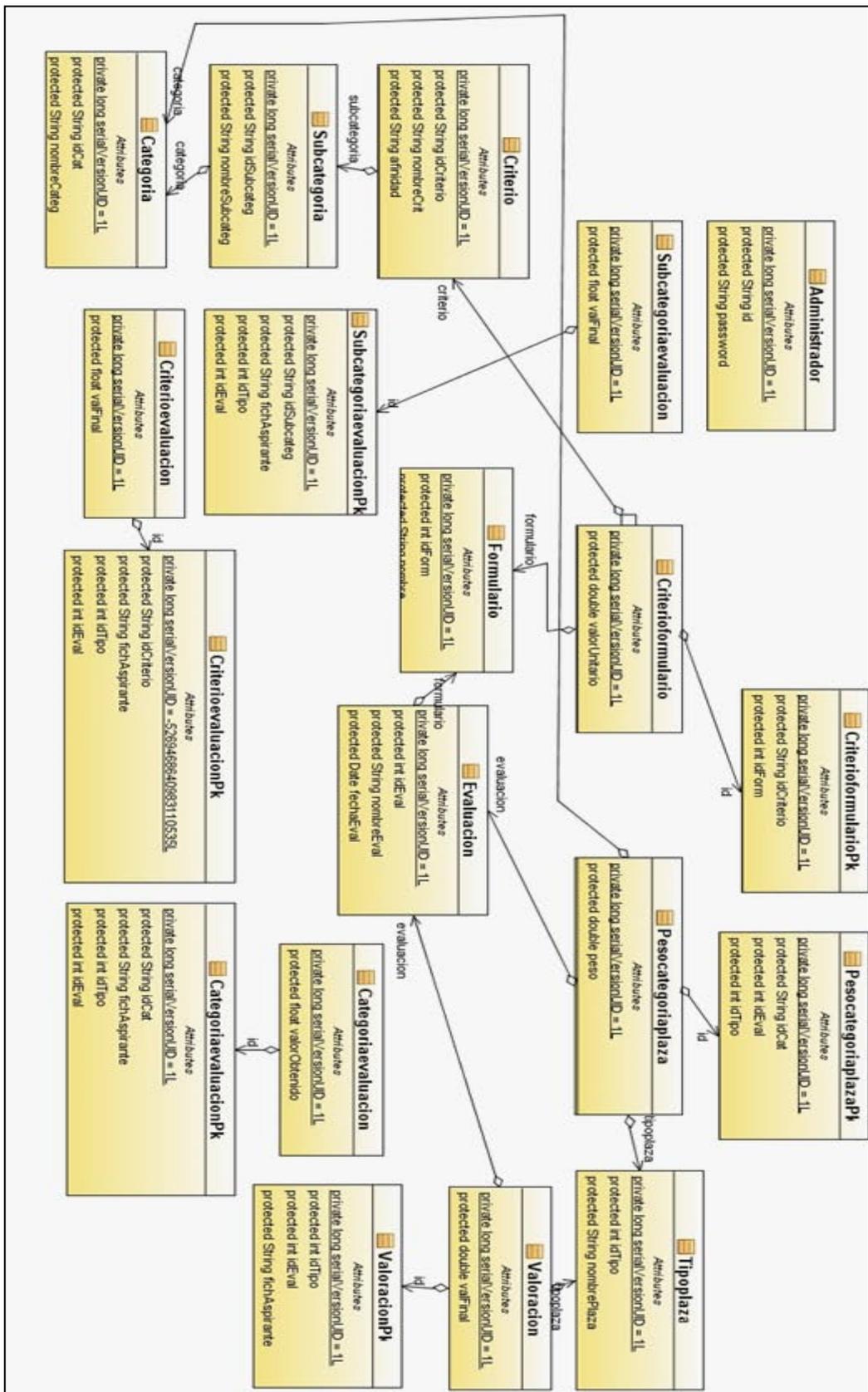


Figura 4.15 Diagrama Modelo

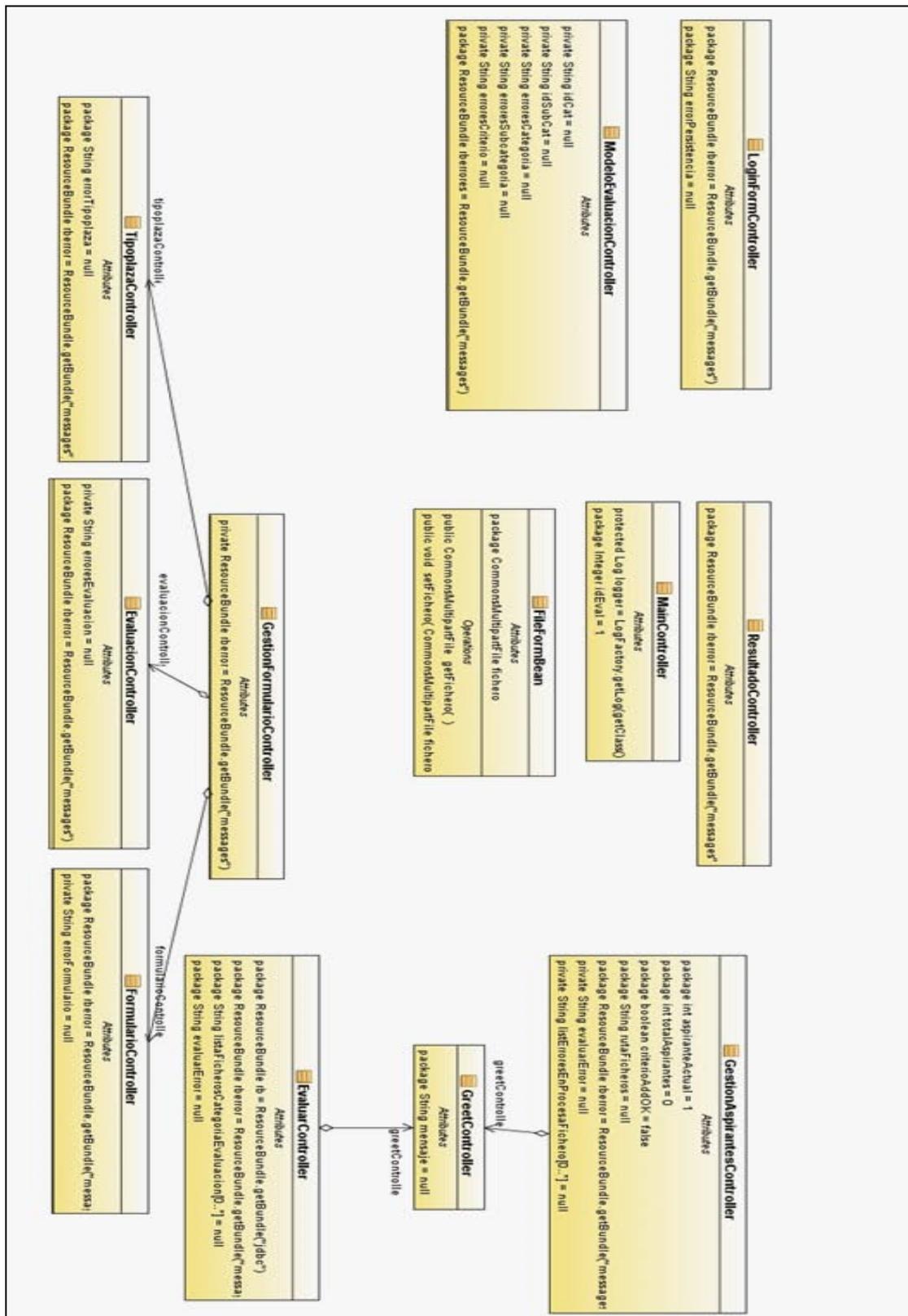


Figura 4.16. Controlador Web

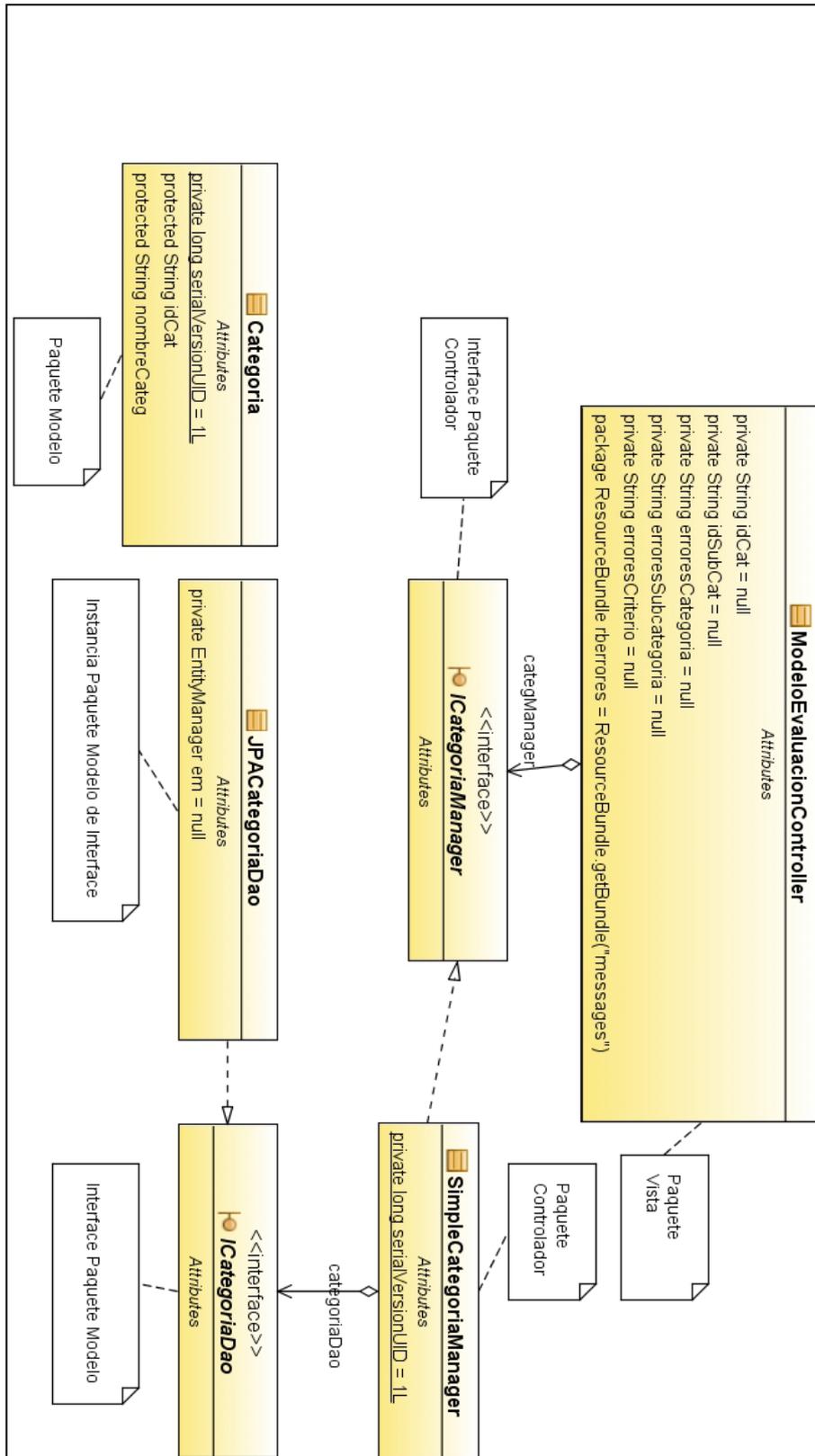


Figura 4.17 Diagrama Ejemplo de comunicación entre paquetes.

4.3.2 Diseño de los Datos

El objetivo de esta fase de diseño es determinar la estructura que tienen cada uno de los elementos de información del sistema, es decir, la estructura de los datos sobre los que se va a trabajar.

Haremos las siguientes **consideraciones** iniciales:

Nuestra base de datos, servirá de apoyo para el cálculo, y recuperación de los datos de las evaluaciones que se efectúen, qué elementos son los que han sido tenidos en cuenta en esa evaluación, y recoger las puntuaciones de cada uno de los aspirantes que se han presentado a la evaluación.

De los candidatos, no vamos a tener ningún dato en la base de datos del tipo nombre, apellidos, etc., ya que contamos con el fichero en formato legible para el sistema, en nuestro caso XML, en el estarán recogidos. Lo único que necesitamos, es el nombre del fichero en el que se podrán recuperar esos datos. En cada evaluación, se recogerá el aspirante que se ha presentado, y a la plaza en concreto que se ha presentado.

Los demás datos que necesitamos, son para el análisis y la muestra de datos A saber, la valoración que ha obtenido cada candidato de manera global, y en cada uno de los puntos.

Se tiene en cuenta, que podremos recuperar una evaluación pasada, los aspirantes que se presentaron, y qué formulario se utilizó, así como las valoraciones pasadas de los aspirantes.

Como consideración final, al ser un sistema web, debemos de restringir el acceso a las personas autorizadas, por lo que se crea la figura del Administrador.

Teniendo en cuenta las consideraciones anteriores, determinamos que los datos con los que vamos a trabajar y que necesitamos almacenar son los siguientes:

- o Del **administrador**, se almacenara su identificador, y su contraseña.

- Las **categorías**. Para las categorías, tendremos su nombre, y un identificador, que tendrá la función de clave principal, para poder editar posteriormente el nombre, si se desea.
- Las **subcategoría**, que estarán compuestas por un nombre y un identificador que será la clave principal. Tendremos que saber a que categoría pertenece cada subcategoría.
- Los **criterios**, de los que tendremos como información un identificador, que funcionara como clave principal, el nombre, un valor unitario, que será por el que posteriormente multiplicaremos para hacer los cálculos, la afinidad, que también participara en la modulación de acuerdo a si el área es propia o no del área de conocimiento. Deberemos recoger a que subcategoría pertenece cada criterio.
- **Tipo de plaza**, en el que tendremos la ponderación que se hace de cada bloque según el tipo de plaza. Almacenaremos un id, por el motivo de poder cambiar el nombre del tipo de plaza, si así fuera necesario.
- En **formulario**, se recogerán los criterios que van a ser tenidos en cuenta en el proceso de selección. Deberemos de tener en cuenta, que van a poder ser recuperados posteriormente, y por lo tanto, se almacenarán de manera histórica.
- La **evaluación**, nos servirá para relacionar los formularios correspondientes, con los tipos de plaza que van a ser ofertadas en esa evaluación.
- Con **valoración**, reflejamos la valoración final que han obtenido los aspirantes que se han presentado a una plaza,. También, servirá para relacionar las valoraciones que se hayan obtenido en cada apartado (en cada criterio).

Una vez hemos descrito los elementos de información, se debe obtener su representación en forma de tablas de la base de datos [31]. Para realizar el diseño adecuado de los datos utilizaremos el modelo Entidad-Relación.

- **El modelo Entidad-Relación**

Es conocido también por sus iniciales E-R. Es una técnica de modelado de datos que utiliza diagramas entidad-relación. Es la técnica para modelar datos más extendida, aunque no la única.

Un diagrama entidad-relación está compuesto básicamente por tres elementos básicos:

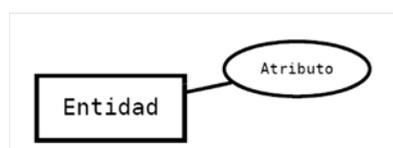
- **Entidades:** Objetos sobre los que tenemos información. Pueden ser personas, cosas, etc. En el diagrama se representan por rectángulos, y cada una tiene un nombre que la identifique de manera única del resto de entidades. Cada ejemplar concreto de una entidad se llama instancia.



- **Relaciones:** Interdependencias entre una o más entidades. Se representan mediante rombos etiquetados con un verbo en el interior.
- Si la relación se establece entre una entidad consigo misma, se denomina reflexiva, si es entre dos entidades, binaria, ternaria si es entre tres, y múltiple cuando es entre más de tres.



- **Atributos:** Son las características propias de una entidad o relación. Se suelen representar mediante elipses etiquetados en el interior con un nombre.



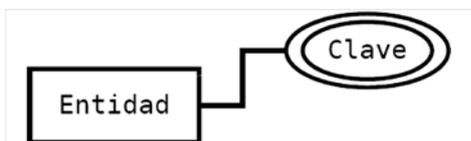
En los diagramas entidad- relación hay que tener en cuenta además de lo ya mencionado otros aspectos como son:

Cardinalidad de las relaciones: Hay tres tipos de cardinalidades:

- *Uno a uno*: Una instancia de la entidad A se relaciona solamente con una instancia de la entidad B. Se representan como 1:1.
- *Uno a muchos*: Cada instancia de la entidad A se relaciona con varias instancias de la entidad B. Se representan como 1:*
- *Muchos a muchos*: Cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B. Se representan como *:*

Entidades débiles: Decimos que una entidad es débil cuando no se puede identificar unívocamente sólo con sus atributos sino que necesitan estar relacionadas con otras entidades para existir. Su representación es con dos rectángulos concéntricos de distinto tamaño, con el nombre en el interior del más pequeño.

Claves: Cada entidad de un diagrama entidad-relación debe tener una clave, que debe estar formada por uno o varios de sus atributos. Dicha clave es la que distingue unívocamente una instancia de la citada entidad del resto.



Una vez que hemos descrito todos los elementos del diagrama Entidad-Relación que vamos a utilizar, describiremos los pasos que vamos a seguir para llegar a obtener las tablas.

Los pasos son:

1. Convertir el enunciado del problema en un Esquema Conceptual del mismo.
2. Convertir este Esquema Conceptual (o EC) en uno más refinado, conocido como Esquema Conceptual Modificado (ECM).
3. Obtener las tablas de la base de datos a partir del Esquema Conceptual Modificado.

Normalización:

La normalización es un proceso que consiste en imponer a las tablas ciertas restricciones mediante una serie de transformaciones consecutivas. Con ello nos aseguramos de que las tablas contengan los atributos necesarios y suficientes para describir la realidad de la entidad que representan, separando aquellos que pueden contener información cuya relevancia permite la creación de otra nueva tabla.

Para asegurar la normalización, Codd estableció tres formas normales, las cuales hacen que toda base de datos que las cumple se considere normalizada.

Estas formas normales son:

- *Primera forma Normal (FN₁):* Una tabla está en FN₁ si todos los atributos no clave dependen funcionalmente de la clave, o lo que es lo mismo, no existen grupos repetitivos para un valor de clave.
- *Segunda forma Normal (FN₂):* Una tabla está en FN₂ si está en FN₁ y además todos los atributos que no pertenecen a la clave dependen funcionalmente de ella de forma completa. De esta definición se puede concluir que una tabla en FN₁ y cuya clave está compuesta por un único atributo ya está en FN₂.
- *Tercera forma Normal (FN₃):* Una tabla está en FN₃ si está en FN₂ y además no existen atributos no clave que dependan transitivamente de la clave.

- **Esquema Conceptual**

Las relaciones que vamos a establecer son las siguientes:

R₁: *Pertenece*, relaciona cada categoría con las distintas subcategorías que la componen. La cardinalidad es de (1:N) ya que cada categoría estará formada por varios apartados, representados por las subcategorías.

R2: *Forma parte de*, establece que cada criterio, pertenecerá a una subcategoría, y que dentro de cada una de ellas, habrá distintos apartados, representados por los criterios. Por ese motivo, la cardinalidad es de (1:N).

R3: *Es tenido en cuenta*, nos dice, que criterios van a formar parte del formulario de evaluación. Asociada a esta relación de (N:N), está lo que se denomina valor unitario, que es la importancia que tiene cada criterio en ese formulario. Está asociada a la relación, porque podrá ir cambiando dependiendo de las características de los formularios.

R4: Es una relación ternaria entre Evaluación, tipo de plaza de (N:N), y valoración. La relación establecida entre Evaluación y TipoPlaza nos sirve para definir los tipos de plaza que habrá en esa evaluación, y también, para saber, a partir de la relación establecida con Valoración, qué candidatos se han presentado a esa plaza, en esa evaluación.

R5: Es una relación ternaria que relaciona categoría con TipoPlaza para almacenar los pesos que va a tener cada categoría dentro de cada plaza. Para tener constancia de en qué evaluación ha sido, esta entidad forma parte también de la relación. Asociamos el peso en la relación, ya que así hacemos que pueda cambiar el peso que toma cada categoría en cada tipo de plaza.

R6: *Obtiene en categoría*, determina el valor obtenido por un candidato, en una categoría determinada. Asocia de (N:N), categoría y valoración. Sirve para hacer los históricos de las valoraciones, y evitar recalcularlos cada vez que se recupere una evaluación pasada. Ayuda además, a obtener los datos para las gráficas.

R7: *Obtiene en subcategoría*, tiene la misma misión que R6.

R8: *Obtiene en criterio*, con la misma misión que R6.

Con las relaciones y las entidades ya definidas, obtenemos el Esquema conceptual siguiente (Figura 4.18):

- **Esquema conceptual modificado:**

Para hacer la transformación del Esquema conceptual anterior al Esquema Conceptual Modificado, se deben hacer los siguientes cambios:

- Eliminar todas las entidades débiles.
- Transformar las relaciones de muchos a muchos.
- Eliminar las relaciones que contengan atributos del Esquema Conceptual.

Siguiendo los pasos anteriores, obtenemos nuevas entidades, que surgen de eliminar las relaciones de (N: N), y de añadir los atributos que estaban asociados a la relación. Las relaciones ternarias, también han sido simplificadas.

Las entidades creadas son:

- **CategTipoPlazaEval**, que recogerá el peso que va a tener cada categoría, en cada tipo de plaza, y además, en qué evaluación se le ha dado ese peso, para poder recuperar esos datos de manera histórica.
- **CriterioFormulario**, será la importancia o valor unitario, que ha tomado cada criterio en un formulario. Será necesaria para calcular la valoración que obtiene cada candidato.
- **CriterioEvaluacion**, recoge la valoración que ha obtenido cada candidato en el apartado de los criterios. Se relaciona con la Evaluación para poder recuperar este dato de manera histórica.
- **SubcategoriaEvaluacion**, lo mismo que el anterior, para subcategoría.
- **CategoríaEvaluacion**, por el mismo motivo que CriterioEvaluación, para categoría.
- **EvalTiPlazValora**, surge de dividir la relación ternaria, y recoge las claves de evaluación TipoPlaza y Valoracion.

Con los cambios antes mencionados, y tras incluirlos en el esquema conceptual modificado, el resultado es el que se presenta en la siguiente Figura 4.19:

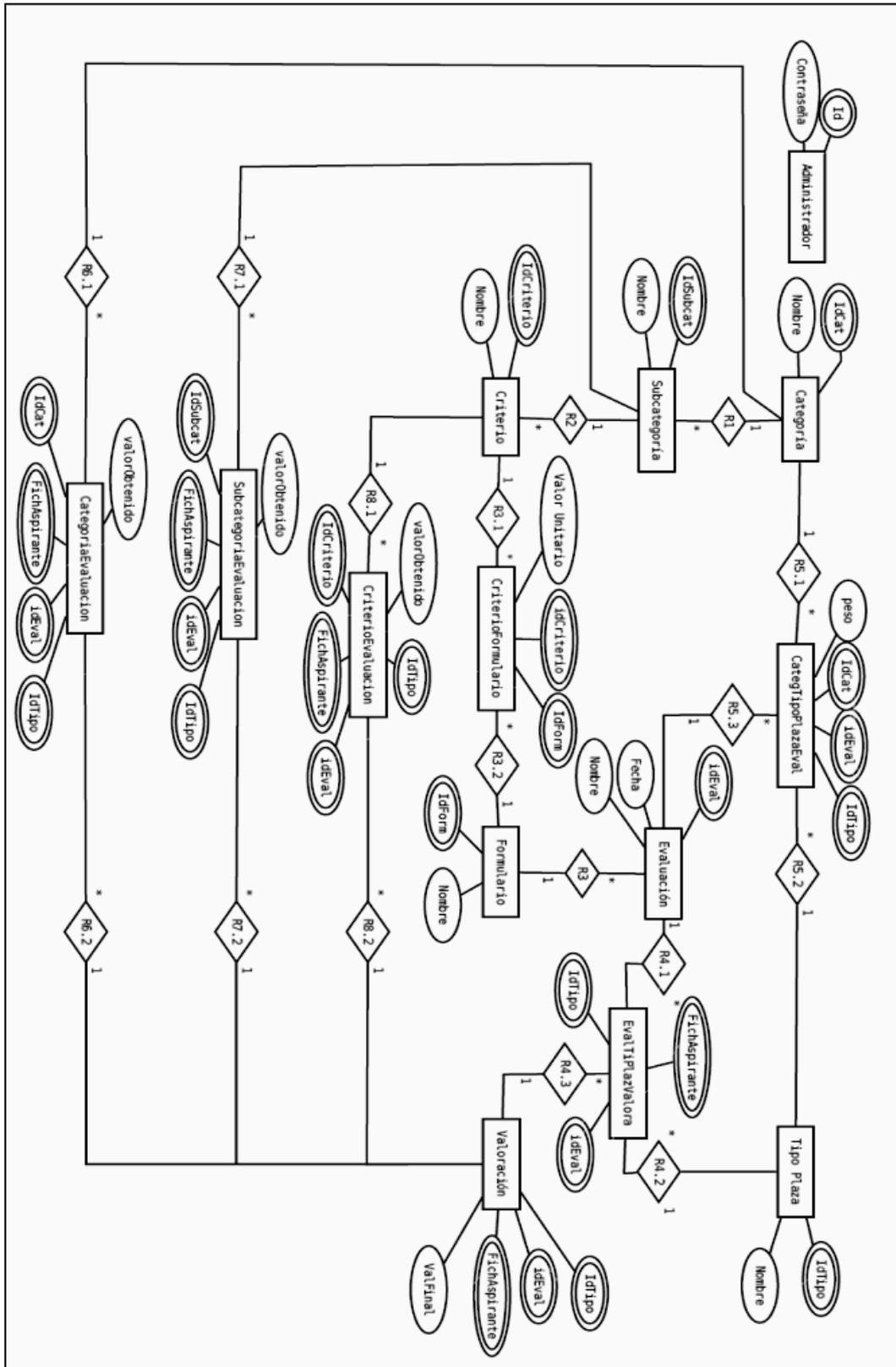


Figura 4.19. Esquema Conceptual Modificado

En este Esquema Conceptual Modificado, se observa que la entidad **EvalTiPlazValora**, solamente almacena las claves, que están también en valoración, ya que se ha recogido en valoración, las claves de las entidades que formaban parte de la relación anterior, al ser una entidad débil.

Nos quedaríamos únicamente con Valoración, eliminando la entidad EvalTiPlazValora, y eliminando una de las relaciones que se habían establecido anteriormente entre esta entidad y Valoración, que ahora ya no serían necesarias.

La entidad Valoración seguiría recogiendo los mismos atributos que tenía anteriormente.

Después de esta transformación, el Esquema Conceptual Modificado, queda simplificado, y se muestra en la siguiente Figura 4.20.

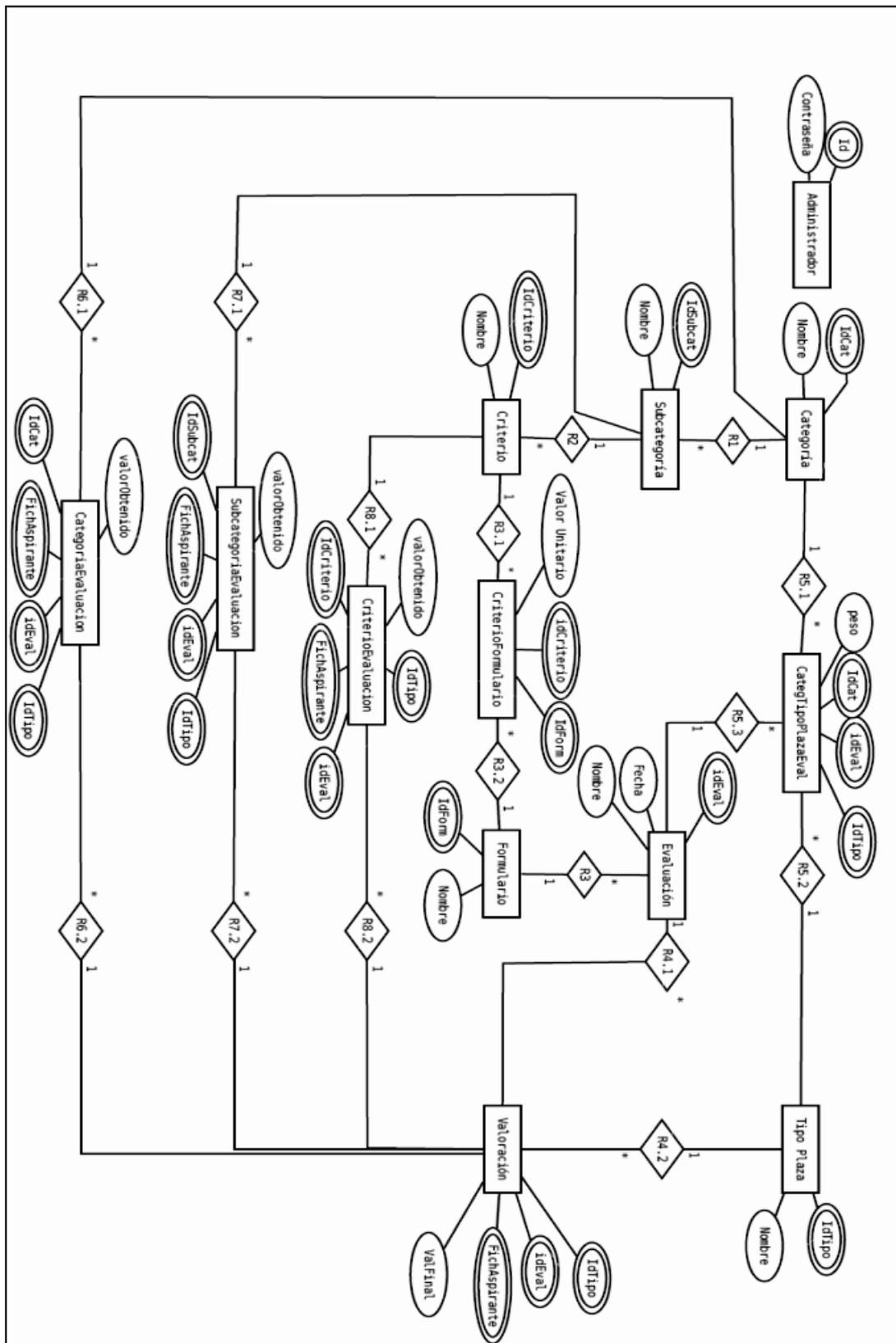


Figura 4.20 Esquema conceptual Modificado y Simplificado

- **Tablas del Sistema**

Tomando como base el ECM anterior, obtenemos un total de ocho tablas en la base de datos. Las reglas que hemos seguido para obtener las tablas han sido las siguientes:

- Cada entidad del ECM se convierte en una tabla.
- Los atributos de una entidad pasan a ser los campos de la tabla respectiva.
- Cada relación pasa a ser clave foránea.

Así, obtenemos estas tablas:

1. La entidad Administrador, pasa a ser la tabla ADMINISTRADOR
2. La entidad Categoría pasa a ser la tabla CATEGORIA.
3. La entidad Subcategoría pasa a ser la tabla SUBCATEGORIA.
4. La entidad Criterio pasa a ser la tabla CRITERIO.
5. La entidad Tipo_Plaza pasa a ser la tabla TIPOPLAZA.
6. La entidad Evaluacion pasa a ser la tabla EVALUACION.
7. La entidad Formulario pasa a ser la tabla FORMULARIO.
8. La entidad Valoracion pasa a ser la tabla VALORACION.
9. La entidad CategTipoPlazaEval pasa a ser la tabla PESOCATEGORIAPLAZA
10. La entidad CriterioFormulario pasa a ser la tabla CRITERIOFORMULARIO.
11. La entidad CriterioEvaluacion pasa a ser la tabla CRITERIOEVALUACION.
12. La entidad SubcatEvaluacion pasa a ser la tabla SUBCATEGORIAEVALUACION
13. La entidad CategoriaEvaluacion pasa a ser la tabla CATEGORIAEVALUACION.

Para tener una visión clara de cómo queda finalmente las tablas, vamos a indicar los campos que compondrían cada una de las tablas que componen nuestros datos:

1. ADMINISTRADOR		
CAMPO	TIPO	DESCRIPCIÓN
Id	varchar(30)	Nombre con el que se identifica al administrador, para que tenga acceso al sistema
Password	varchar(30)	Contraseña, para el acceso al sistema

Tabla 4-I. Campos de la tabla Administrador

2. CATEGORIA		
CAMPO	TIPO	DESCRIPCIÓN
IdCat	varchar(11)	Identificador de categoría con formato I., II. ,etc.
NombreCateg	varchar(120)	Nombre de la categoría

Tabla 4-II. Campos de la tabla Categoría

3. SUBCATEGORIA			
CAMPO	TIPO	ENLACES A	DESCRIPCIÓN
idSubcateg	varchar(11)		Identificador numérico de la subcategoría
Nombre	varchar(120)		Nombre de la subcategoría
idCat	varchar(11)	categoría -> idCat	Identificador de la categoría a la que pertenece

Tabla 4-III. Campos de la tabla Subcategoría

4. CRITERIO			
CAMPO	TIPO	ENLACES A	DESCRIPCIÓN
idCriterio	varchar(11)		Clave principal de criterio, con el formato I.1.1, II.2.1. ,etc.
NombreCrit	varchar(150)		Nombre del criterio
Afinidad	varchar(2)		Valor que será N, S o SA, para determinar si sometemos a modulación
idSubcat	varchar(11)	subcategoría-> idSubcateg	Identificador de la subcategoría a la que pertenece el criterio

Tabla 4-IV. Campos de la tabla Criterio

5. TIPOPLAZA		
CAMPO	TIPO	DESCRIPCIÓN
idTipo	int(11)	Identificador numérico del tipo de plaza
Nombre	varchar(50)	Nombre del puesto que se oferta

Tabla 4-V. Campos de la tabla TipoPlaza

6. EVALUACION			
CAMPO	TIPO	ENLACES A	DESCRIPCIÓN
idEval	int(11)		Identificador de la evaluación
Nombre	varchar(50)		Nombre que se le asigna a la evaluación.
Fecha	timestamp		Fecha en la que se ha producido la evaluación
idForm	int(11)	formulario -> idForm	Identificación del formulario

Tabla 4-VI. Campos de la tabla Evaluacion

7. FORMULARIO		
CAMPO	TIPO	DESCRIPCIÓN
idForm	int(11)	Identificación del formulario
Nombre	varchar(30)	Nombre que se asigna al formulario

Tabla 4-VII. Campos de la tabla Formulario

8. VALORACION			
CAMPO	TIPO	ENLACES A	DESCRIPCIÓN
idTipo	int(11)	tipoplaza -> idTipo	Identificador numérico del tipo de plaza
idEval	int(11)	evaluacion -> idEval	Identificador de la evaluación
fichAspirante	varchar(120)		Fichero en el que está la información con los datos del aspirante. Servirá para recuperar datos como el nombre, apellidos, publicaciones, etc.
ValFinal	decimal(6,3)		Recoge la valoración final que ha obtenido el aspirante

Tabla 4-VIII. Campos de la tabla Valoración

9. PESOCATEGORIAPLAZA			
CAMPO	TIPO	ENLACES A	DESCRIPCIÓN
idCat	varchar(11)	categoria -> idCat	Identificador numérico de la categoría
idEval	int(11)	evaluacion -> idEval	Identificador de la evaluación
idTipo	int(11)	tipoplaza -> idTipo	Identificador numérico del tipo de plaza
peso	decimal(6,3)		Importancia o factor de ponderación, que tiene esa categoría para la plaza indicada

Tabla 4-IX. Campos de la tabla PesoCategoríaPlaza

10. CRITERIOFORMULARIO		
CAMPO	TIPO	DESCRIPCIÓN
idCriterio	varchar(11)	Identificador numérico del criterio
idForm	int(11)	Identificación del formulario
valorUnitario	Decimal(6,3)	Ponderación o peso que tiene el criterio en el formulario indicado

Tabla 4-X. Campos de la tabla CriterioFormulario

11. CRITERIOEVALUACION			
CAMPO	TIPO	ENLACES A	DESCRIPCIÓN
idCriterio	varchar(11)	criterio-> idCriterio	Clave principal de la tabla criterio
fichAspirante	varchar(120)		Fichero en el que está la información con los datos del aspirante. Servirá para recuperar datos como el nombre, apellidos, publicaciones, etc.
idTipo	int(11)		Identificador numérico del tipo de plaza
idEval	int(11)		Identificador de la evaluación
valFinal	decimal(6,3)		Valor Obtenido en ese Criterio

Tabla 4-XI. Campos de la tabla CriterioEvaluacion

12. SUBCATEGORIAEVALUACION			
CAMPO	TIPO	ENLACES A	DESCRIPCIÓN
idSubcateg	varchar(11)	subcategoria -> idSubcateg	Identificador de la subcategoría a la que pertenece el criterio
fichAspirante	varchar(120)		Fichero en el que está la información con los datos del aspirante. Servirá para recuperar datos como el nombre, apellidos, publicaciones, etc.
idTipo	int(11)		Identificador numérico del tipo de plaza
idEval	int(11)		Identificador de la evaluación
valFinal	decimal(6,3)		Valor Obtenido en esa Subcategoría

Tabla 4-XII. Campos de la tabla SubcategoríaEvaluacion

13. CATEGORIAEVALUACION			
CAMPO	TIPO	ENLACES A	DESCRIPCIÓN
idCat	varchar(11)	categoria -> idCat	Identificador numérico de la categoría
fichAspirante	varchar(120)		Fichero en el que está la información con los datos del aspirante. Servirá para recuperar datos como el nombre, apellidos, publicaciones, etc.
idTipo	int(11)		Identificador numérico del tipo de plaza
idEval	int(11)		Identificador de la evaluación
valFinal	decimal(6,3)		Valor Obtenido en esa Categoría

Tabla 4-XIII. Campos de la tabla CategoríaEvaluacion

4.3.3 Diseño de la interfaz

En esta etapa del diseño del sistema software se define cuál va a ser la apariencia visual de la aplicación, es decir, se define la interfaz visual entre el usuario y la aplicación. Sin duda, llevar a cabo un buen diseño de la interfaz resulta decisivo debido a que esta debe ser atractiva para el usuario de la aplicación pero al mismo tiempo debe resultar fácil de entender y de trabajar sobre ella [30].

Esta importancia se acrecienta aun más en el caso de nuestro proyecto ya que la interfaz de nuestro proyecto es Web y la usabilidad en el entorno Web es un tema candente, sujeto a cierta controversia [32]. Esto es así porque para las aplicaciones con interfaces Web no existe una guía de estilo estándar como existe, por ejemplo, para el desarrollo de interfaces para aplicaciones de escritorio en sistemas operativos como Windows y que resulten atractivas y familiares al mismo tiempo. Cada programador, desarrollador o diseñador Web debe definir, pues, su propia guía de estilo y procurar que, en base a ella, la interfaz resultante alcance un nivel digno de atractivo visual, familiaridad y facilidad de uso.

En este apartado definiremos nuestra guía de estilo de la aplicación, describiremos y analizaremos las metáforas y revisaremos el diseño de pantallas empleados para el desarrollo de la interfaz de nuestro prototipo de aplicación software.

Guía De estilo

Antes de entrar de lleno en el diseño de la interfaz de usuario, es conveniente definir el estilo de la misma. Esto es de suma importancia cuando el diseño va a ser compartido entre varios diseñadores, ya que ayuda a mantener la coherencia interna de la interfaz.

Sin embargo, a pesar de lo que pueda parecer en un principio, también es de gran utilidad definir una guía de estilo cuando sólo hay un diseñador encargado de la interfaz.

Esto es debido a varias razones:

- En ocasiones es posible que mantener la coherencia y consistencia de una interfaz, si esta es muy grande o muy ambiciosa, sea complicado incluso si sólo hay un diseñador.
- El diseñador primitivo puede, por las más diversas razones, abandonar el diseño. Por tanto, es de gran utilidad para sus sustitutos contar con una guía de estilo predefinida para no tener que empezar desde cero de nuevo. Esto es también aplicable cuando no es el diseñador original el que se encarga de la actualización o el mantenimiento de la interfaz.

Quedando demostrada la utilidad del uso de guías de estilo, pasamos a definir las reglas y normas que contendrá la guía de estilo de nuestra interfaz

Fuentes

- Tipo: font-family: Verdana,Arial,Helvetica,sans-serif
- Color: #4D767A para encabezados y textos principales
- Negrita
- Tamaños:
 - 35px para encabezado.
 - 18 para Nombre de página
 - 16 resto del texto

Colores

- El color predominante será el azul, acompañado de un gris suave (25%) para los menús y un azul muy oscuro para la cabecera del sitio.
- Contenedor: #FoFoFo
- Color-fondo: #C2D7F8

Tamaños:

Los tamaños de las distintas partes de las pantallas, se pueden ver en la

Figura 4.21.

Los márgenes que no están indicados, corresponden todos a 10px. Tampoco se ha incluido el tamaño de la zona “Contenido”, ya que al ser variable, no se le puede asignar a priori un tamaño.

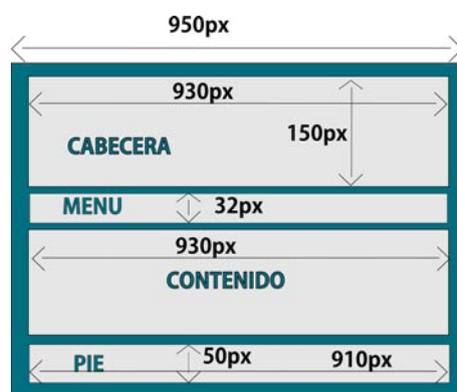


Figura 4.21. Tamaños de las distintas partes de la pantalla.

Con esta guía de estilo, se puede tener una idea aproximada de la apariencia final de las pantallas sistema, así como la adecuación a los navegadores y las resoluciones más utilizadas.

Pantallas de la aplicación

En este apartado, vamos a definir la estructura de nuestra interfaz con el usuario, ya que la parte del servidor no tendrá interfaz ninguna, pues se ejecutará automáticamente sin intervención manual.

Mediante el diseño de prototipos se pretende tener un esbozo de lo que será la interfaz de usuario de nuestra aplicación. Dichos prototipos no expresan el diseño final, simplemente dan una idea de lo que será nuestro sistema para el usuario final. Estos prototipos serán, por tanto, susceptibles de cambio durante el proceso de implementación.

En nuestro caso, optamos por el **diseño de pantallas** como técnica de definición de prototipos de la interfaz.

A continuación se mostrarán las pantallas que se han construido:

Universidad de Jaén
Sistema de Selección de Personal. PDI

PÁGINA DE ENTRADA AL SISTEMA

Introduzca su usuario y contraseña para empezar

Admin:

Password:

Entrar

Figura 4.22. Pantalla principal de Acceso

Universidad de Jaén

Sistema de Selección de Personal. PDI

Admin: Admin [salir](#)

[Evaluar Aspirantes](#) [Resultados](#) [Gestión de los elementos de Evaluación](#) [Gestión Formulario](#)

Evaluación actual: Evaluacion 2012 [Cambiar](#) [nueva](#)

DATOS DE LOS ASPIRANTES

Plaza: [Cambiar](#) [nueva](#)

/

Datos Personales

Fichero: *Pedro Ejemplo*
Nombre: *Pedro*
Apellidos: *Ejemplo*
Dirección: *c\ Extremadura, 18, Toledo, España*
Teléfono: *666 666 666*

Méritos

Criterios	Valoración
Nota Media	9,1
Tesina	1
Premios nacionales	3
...	...

Para actualizar los nuevos aspirantes que se hayan podido añadir, pulse en actualizar [Actualizar](#)

Añadir nuevo Aspirante: [Calcular Particiones](#)

Seleccione el fichero [Examinar](#) [Subir Fichero](#)

Figura 4.23. Pantalla de Evaluar Aspirantes



Figura 4.24. Página de Resultados

Universidad de Jaén

Sistema de Selección de Personal. PDI

Admin: Admin [salir](#)

[Evaluar Aspirantes](#) [Resultados](#) [Gestión de los elementos de Evaluación](#) [Gestión Formulario](#)

GESTIÓN DE LOS ELEMENTOS DE EVALUACIÓN

Categoría:

Nueva **Modificar**

Subcategoría:

Nueva **Modificar**

Criterio:

Nueva **Modificar**

Figura 4.25. Página de la Gestión de los Elementos de Evaluación

Universidad de Jaén

Sistema de Selección de Personal. PDI

Admin: Admin [salir](#)

Evaluar Aspirantes

Resultados

Gestión de los elementos de Evaluación

Gestión Formulario

Evaluación actual: Cambiar Borrar

Nueva
 Modificar

Formulario actual: Ver Asignar Borrar

Nueva
 Modificar

GESTIÓN DEL FORMULARIO

Criterio: ▼

Peso: Añadir

Criterios Añadidos

Criterio	Peso	
I.1.1. Nota media de la titulación	0.3	Eliminar
I.3.1. Meses de becario de alumno colaborador de departamento	0.9	Eliminar
II.3.2. Créditos impartidos en cursos de formación	0.04	Eliminar

Plaza Cambiar

Nueva
 Modificar

Categoría ▼

Peso: Añadir

Categorías añadidas a *Ejemplo Plaza*

Categoría	Peso	
I. Formación académica	0.3	Eliminar
III. Actividad investigadora	0.04	Eliminar

Figura 4.26. Página de la Gestión del Formulario

Caminos de navegación

Hasta este momento tenemos un diseño visual de la interfaz estática, es decir, cada pantalla diseñada individualmente, pero no tenemos una idea de si en el conjunto de la interacción, la acción va a transcurrir de forma comprensible para el usuario. Para ello vamos a diseñar la interfaz en movimiento y comprobar que es usable.

Para estudiar los caminos de interacción se empleará una herramienta llamada *storyboard*, que consiste en mostrar, a modo de secuencia, las distintas pantallas por las que se va pasando al realizar el usuario una determinada acción sobre la aplicación. Mediante flechas se ayuda a entender que es lo que ha desencadenado el paso de una pantalla a otra.

La técnica del *Storyboarding* nos resultará altamente útil para describir los escenarios de situaciones concretas que ayuden a entender partes del sistema. Con los *storyboards* se consigue dotar al escenario descrito en lenguaje natural de la componente gráfica que facilita la comprensión y el detalle. Además, podemos analizar las pantallas de este escenario para ver si los pasos que se dan son los correctos y si la acción se entiende bien, de forma que podría ser necesario añadir más pantallas intermedias o, por el contrario, podría convenir fusionar algunas pantallas. Los *storyboards* nos facilitan esta tarea ya que nos permiten ver juntas todas las pantallas asociadas a una tarea o escenario

No se han desarrollado para todas las acciones de nuestro sistema por lo que los que se muestran a continuación, son los que se han considerado más importantes.

Una vez diseñados deberían ser validados para comprobar que realmente la aplicación es usable.

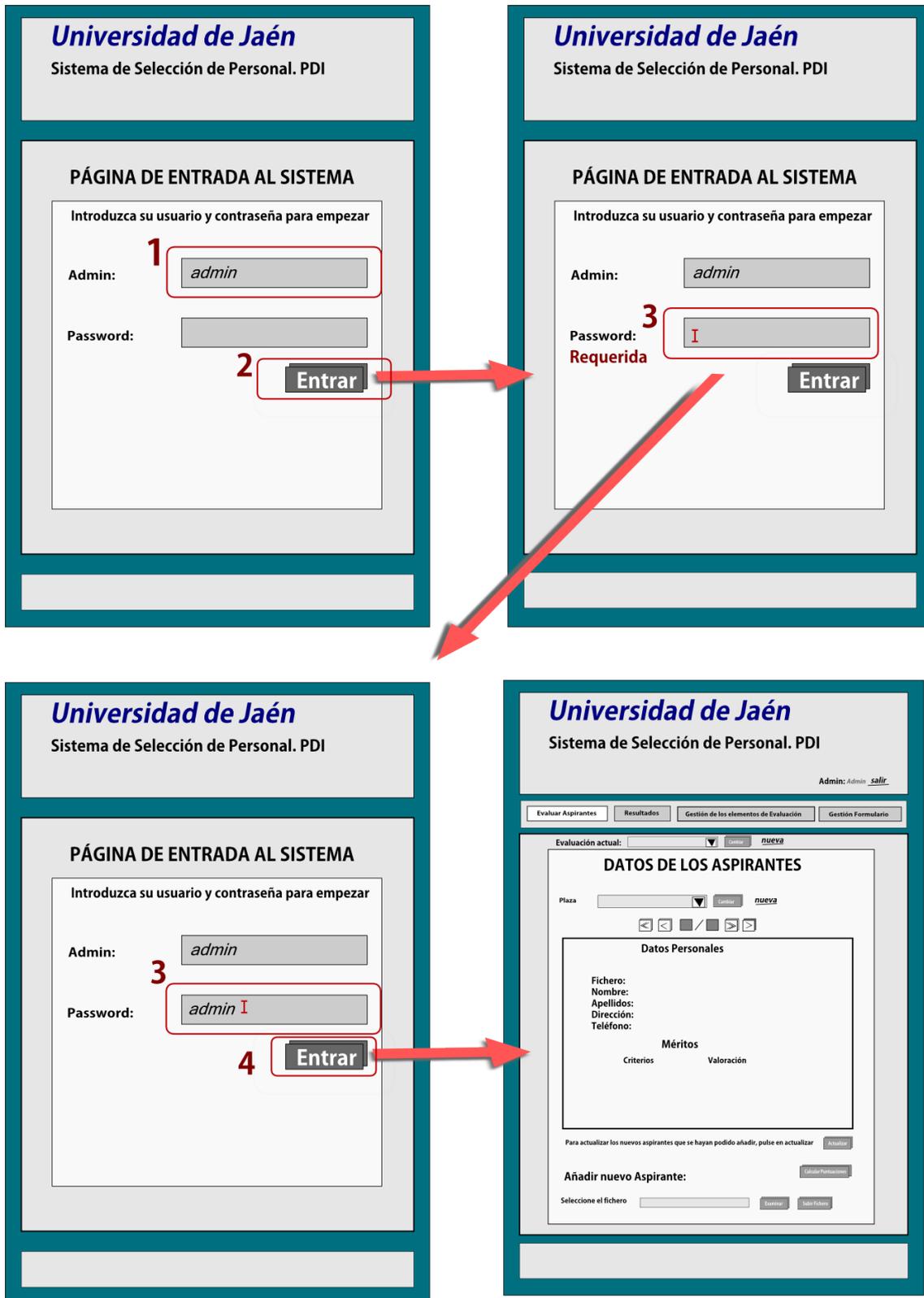


Figura 4.27. StoryBoard-Login en el Sistema

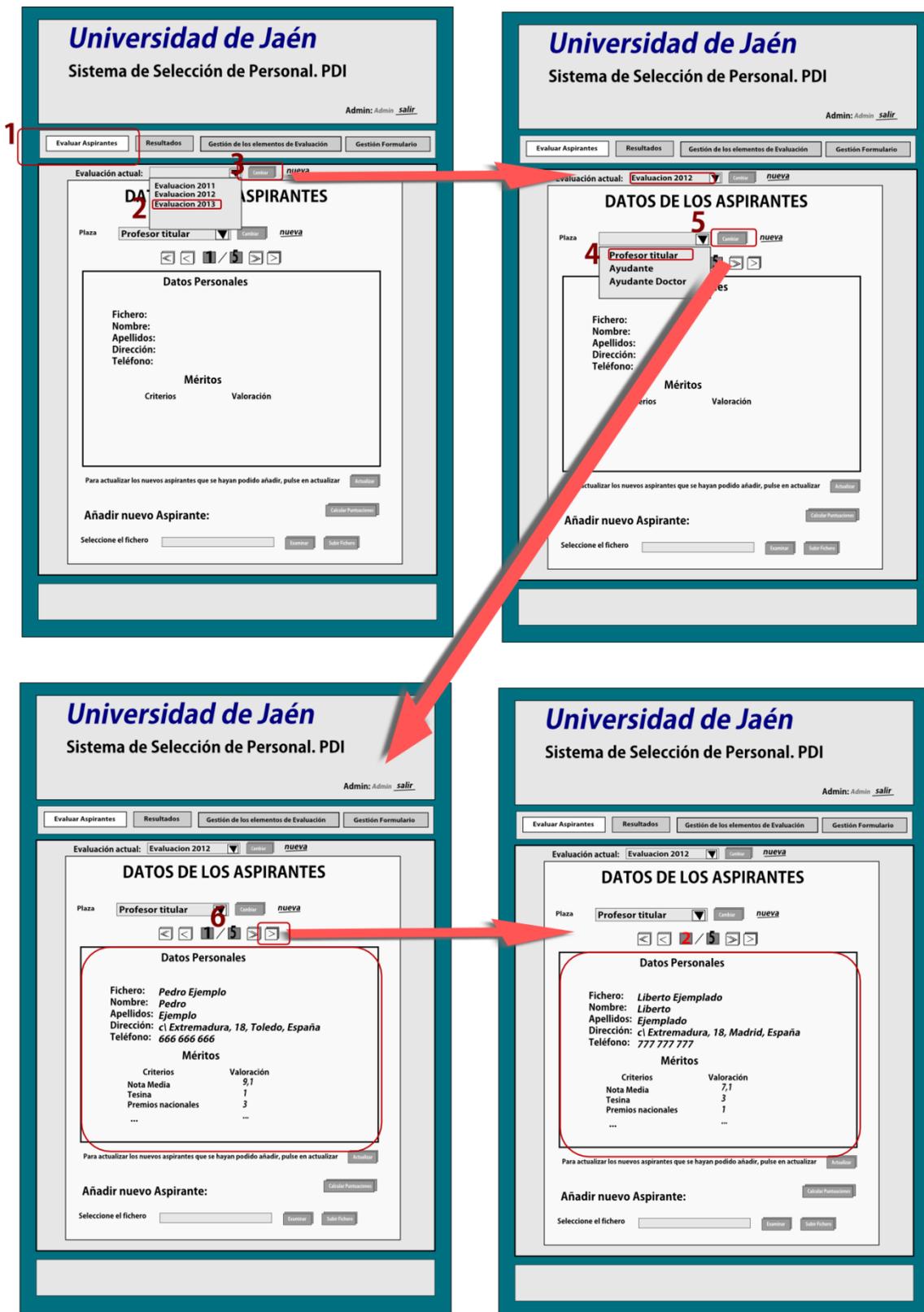


Figura 4.28. Storyboard-Recuperar los aspirantes a una evaluación anterior

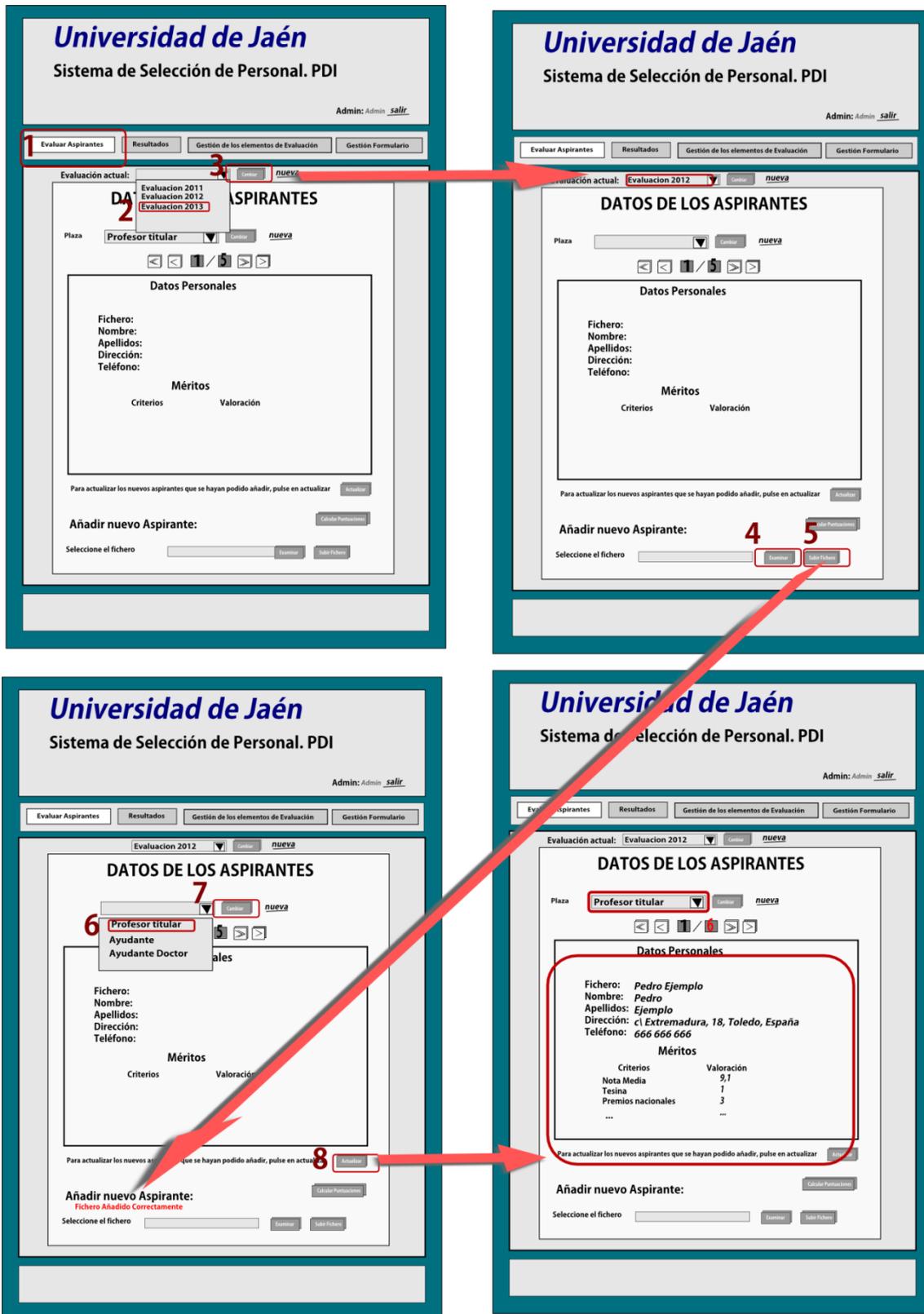


Figura 4.29. Storyboard-Añadir nuevo Aspirante a Evaluación

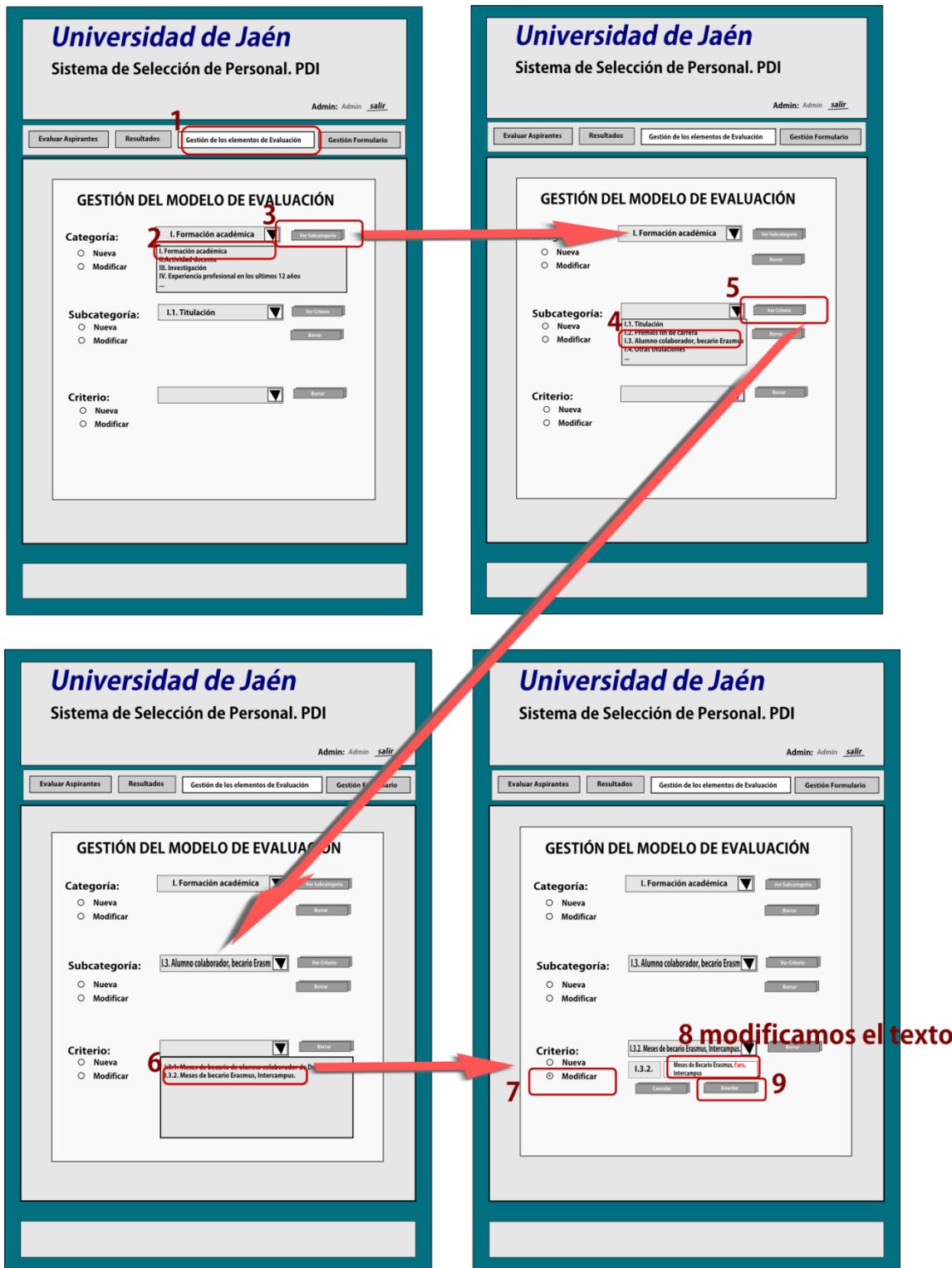


Figura 4.30. Storyboard-Modificar un criterio de los elementos de la Evaluación

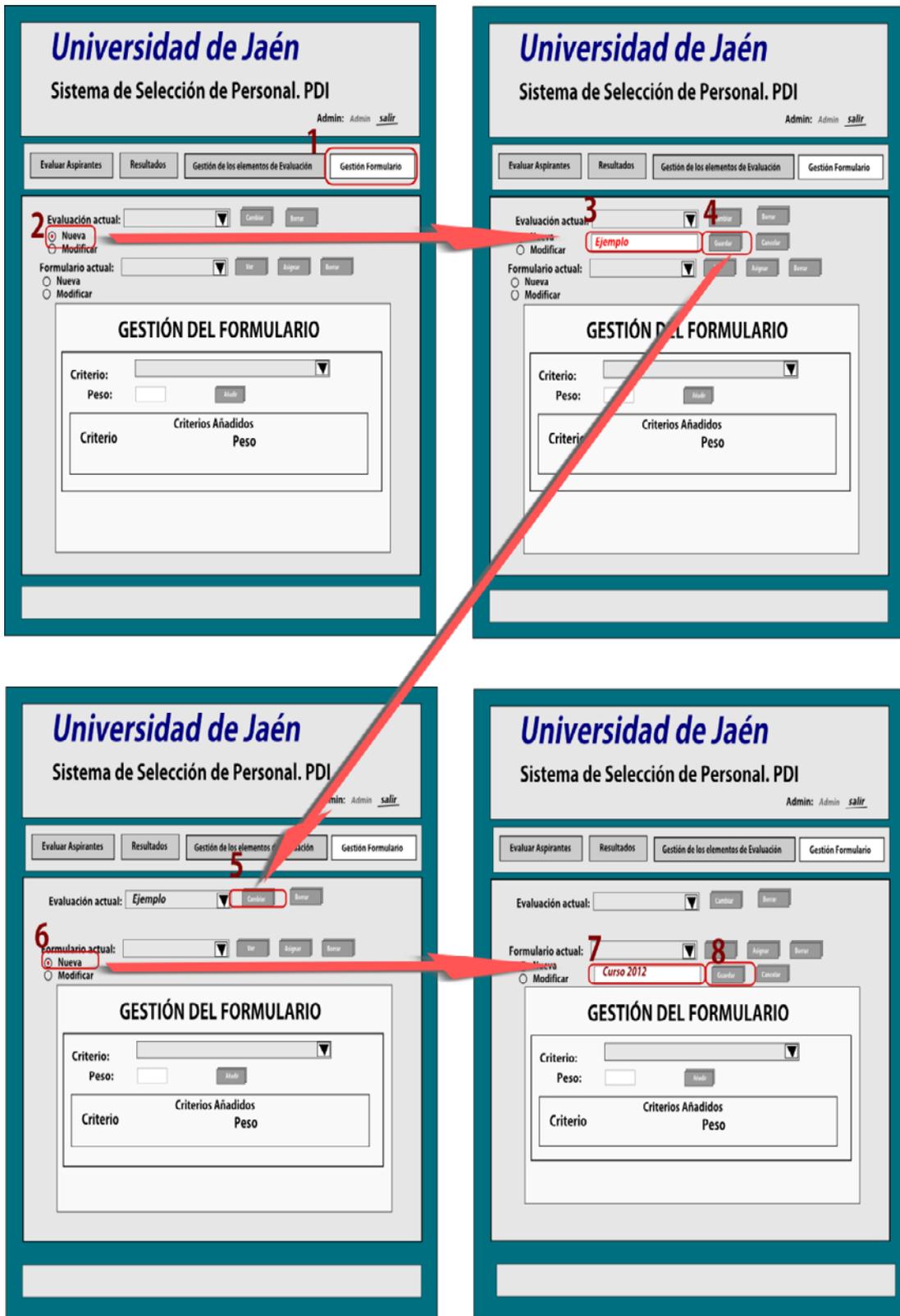


Figura 4.31. Storyboard-Creación de una Evaluación y un Formulario de Evaluación

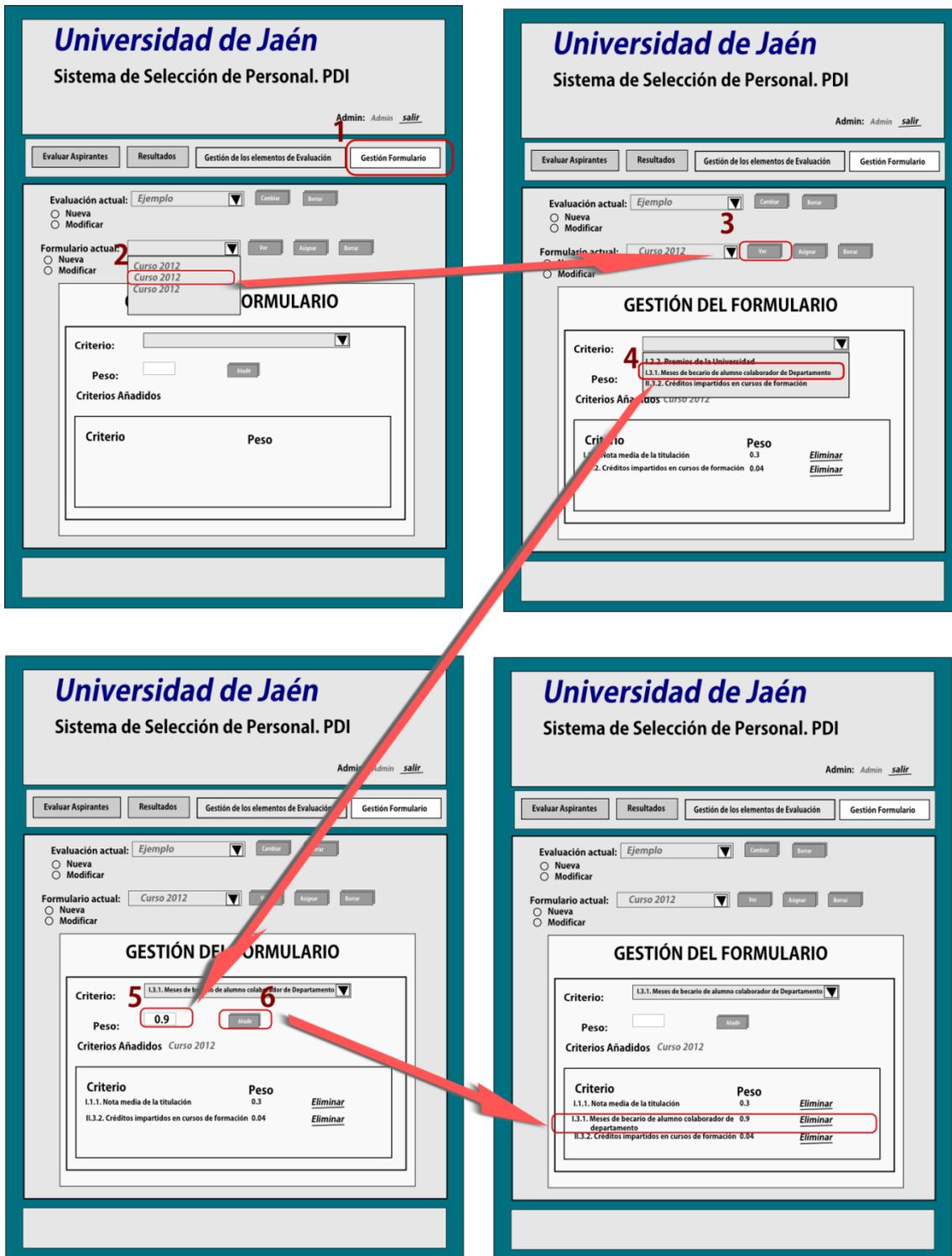


Figura 4.32. Storyboard- Añadir un nuevo criterio a un Formulario de Evaluación.

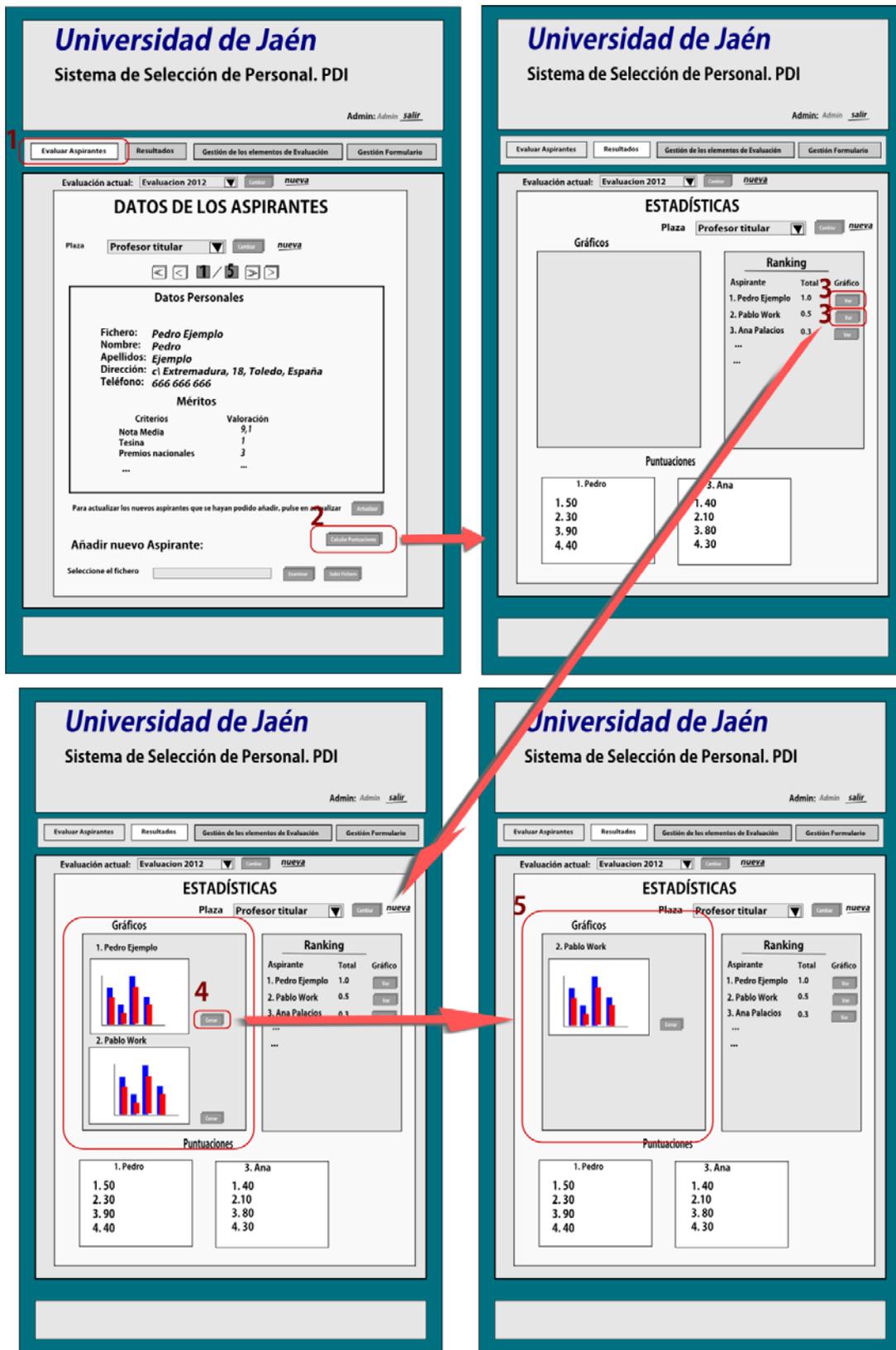


Figura 4.33. Storyboard- Realizar cálculos y ver gráficos

4.4 Implementación

La implementación es la actividad final de la Ingeniería del Software. Aquella en la que el modelo obtenido en las actividades anteriores se debe transformar en código fuente. Para ello se debe ser cuidadoso en la elección del lenguaje de programación empleado para la codificación y de la herramienta utilizada para generarla.

En el proyecto actual, la elección del lenguaje de programación así como el de la herramienta utilizada para su desarrollo vienen condicionados por el tipo de aplicación que necesitamos por la definición que se hizo del proyecto.

4.4.1 Tipo de la arquitectura de la aplicación.

En nuestro caso, vamos a desarrollar un sistema con una arquitectura cliente/servidor y una interfaz Web de comunicación con los usuarios. El funcionamiento de las arquitecturas de este tipo es sencilla: la aplicación se encuentra en un servidor central al que los usuarios acceden a través de un software cliente, en nuestro caso, un navegador Web. Una vez que ha accedido a la aplicación, el usuario realiza peticiones que el servidor tiene que atender para generar una respuesta comprensible para el cliente (Figura 4.34).

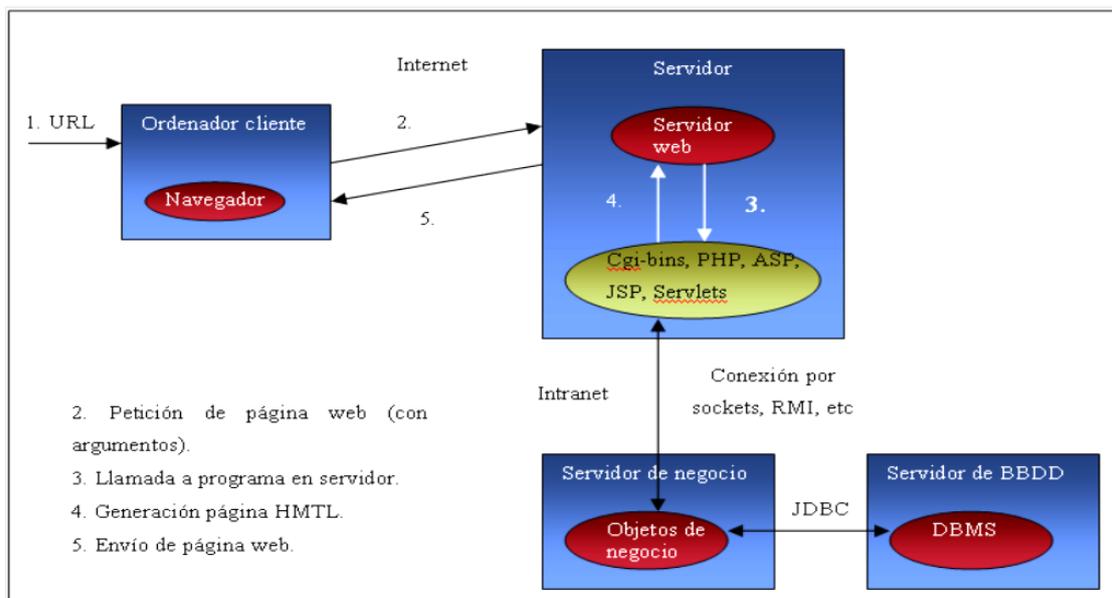


Figura 4.35 Arquitectura MVC cliente-servidor

Una arquitectura cliente/servidor Web libera, por lo tanto, al usuario final de la aplicación de tener que instalarla en su máquina. Además tiene la ventaja de que consigue que cada usuario sólo pueda acceder a la información que le corresponde. Asimismo, este tipo de arquitectura, gracias a su diseño modular, es fácilmente escalable y ampliable tanto en nuevos clientes como en servidores añadidos.

4.4.2 Lenguajes de programación utilizados.

Resulta obvio ante la arquitectura y el funcionamiento previsto de nuestra aplicación que el uso de HTML simple y llano no es adecuado sino que, se necesita otro lenguaje capaz de generar contenido dinámico desde el servidor de manera transparente al usuario final. Para este fin hemos utilizado JSP.

Java Server Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Los JSP's permiten la utilización de código Java mediante scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

JSP no se puede considerar un script al 100%, ya que antes de ejecutarse el Servidor de Aplicaciones compila el contenido del documento JSP (script y etiquetas), y genera una clase Servlet. Por lo tanto, se puede decir que aunque este proceso sea transparente para el programador no deja de ser una tecnología compilada.

La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos (así como las que requieren más seguridad) y dejando la parte encargada de formatear el documento HTML en el archivo JSP. La idea fundamental detrás de este criterio es el de separar la lógica del negocio de la presentación de la información, clave en el esquema MVC que estamos siguiendo.

Para realizar una interfaz web adecuada se hace necesario el uso de otros dos lenguajes de programación: CSS y Javascript.

CSS, acrónimo de Cascade Style Sheets, es un lenguaje formal que ayuda a separar la estructura interna de un documento de su presentación externa. Las etiquetas de estilo CSS pueden presentarse tanto dentro de un documento HTML (encerradas dentro de las etiquetas `<style type="text/css"></style>` en la cabecera) como en un documento aparte (con extensión .css) al que el documento HTML se encarga de llamar cuando es necesario. De esta última manera no sólo, se consigue separar la estructura de la presentación sino que se consigue la centralización del estilo. Debido a que una sola hoja de estilos CSS puede ser invocada por distintas páginas de la aplicación web, lo que ayuda de manera muy importante al mantenimiento de la coherencia y consistencia del diseño de la aplicación.

Por su parte, Javascript, lenguaje interpretado de sintaxis similar a lenguajes como Java o C que se ejecuta del lado del cliente, ayuda a comprobar si los datos que introduce el cliente son erróneos informando de su error mediante mensajes de error o alerta.

Al igual que ocurre con CSS, el código Javascript puede ir incrustado dentro del documento HTML (entre las etiquetas `<script type="text/javascript"></script>` en el cuerpo o la cabecera) o estar almacenado en ficheros aparte (con extensión .js) y ser invocados por el documento. Para nuestra aplicación, para los estilos CSS nos hemos decantado por la segunda opción.

Como ya se vio al comienzo de esta memoria, para la lógica de negocio del sistema se va a utilizar tecnología Servlets de JAVA.

Para implementar la tecnología Servlets de JAVA, nos servimos de uno de los múltiples frameworks³ que hay en el mercado. Se ha optado por usar Spring Framework MVC [33].

Spring Framework (también conocido simplemente como Spring) es un framework Java de Software Libre, con pocos requisitos de hardware para funcionar. Su

³ framework: Conjunto de procesos y tecnologías usados para resolver un problema complejo. Es el esqueleto sobre el cual varios objetos son integrados para facilitar una solución dada.

principal característica es brindar una factoría de objetos basado en la Inyección De Dependencia [34].

A pesar de que Spring Framework no obliga a usar un modelo de programación en particular, se ha popularizado en la comunidad de programadores en Java al ser considerado como una alternativa y sustituto del modelo de Enterprise JavaBeans. Por su diseño, el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria.

Mientras que las características fundamentales de este framework pueden emplearse en cualquier aplicación hecha en Java, existen muchas extensiones y mejoras para construir aplicaciones basadas en web por encima de la plataforma Java EE.

Spring proporciona:

- Una potente gestión de configuración basada en JavaBeans. Esto hace que la configuración de aplicaciones sea rápida y sencilla. Ya no es necesario usar el patrón singleton⁴ ni ficheros de configuración
- Una aproximación consistente y elegante. Puede ser usado en cualquier entorno, desde applets⁵ hasta contenedores J2EE. Estas definiciones de beans se realizan en lo que se llama el contexto de aplicación.
- Una capa de abstracción JDBC que ofrece una significativa jerarquía de excepciones (evitando la necesidad de obtener de SQLException que son las excepciones de SQL, los códigos que cada gestor de base de datos asigna a los errores), simplifica el manejo de errores, y reduce considerablemente la cantidad de código necesario.
- Integración con Hibernate, JDO⁶ e iBatis SQL Maps⁷ en términos de soporte a implementaciones DAO y estrategias con transacciones. Especial soporte a Hibernate y solucionando muchos de los comunes

⁴ En ocasiones necesitamos instanciar una clase solo una vez, la solución es hacer que sea la propia clase la responsable de controlar la existencia de una única instancia

⁵ ver documentación en: <http://docs.oracle.com/javase/tutorial/deployment/applet/>

⁶ ver: <http://www.oracle.com/technetwork/java/index-jsp-135919.html>

⁷ ver: <http://ibatis.apache.org/>

problemas de integración. Todo ello cumpliendo con las transacciones genéricas de Spring y la jerarquía de excepciones DAO.

- Un framework MVC (Model-View-Controller), construido sobre el núcleo de Spring. Este framework es altamente configurable vía interfaces y permite el uso de múltiples tecnologías para la capa vista como pueden ser JSP, Velocity⁸ o POI⁹ entre otros. De cualquier manera una capa modelo realizada con Spring puede ser fácilmente utilizada con una capa web basada en cualquier otro framework MVC, como Struts.

Toda esta funcionalidad puede usarse en cualquier servidor J2EE, y la mayoría de ella ni siquiera requiere su uso. El objetivo central de Spring es permitir que objetos de negocio y de acceso a datos sean reusables, no atados a servicios J2EE específicos. Estos objetos pueden ser reutilizados tanto en entornos J2EE (web o EJB), aplicaciones standalone¹⁰ o entornos de pruebas, sin ningún problema.

La arquitectura en capas de Spring Figura 4.36, “Spring: arquitectura en capas” ofrece cantidad de flexibilidad. Toda la funcionalidad está construida sobre los niveles inferiores. Por ejemplo se puede utilizar la gestión de configuración basada en JavaBeans sin utilizar el Framework MVC.

⁸ ver :<http://velocity.apache.org/>

⁹ ver : <http://poi.apache.org/>

¹⁰ standalone: De funcionamiento autónomo, sin necesidad de servidor

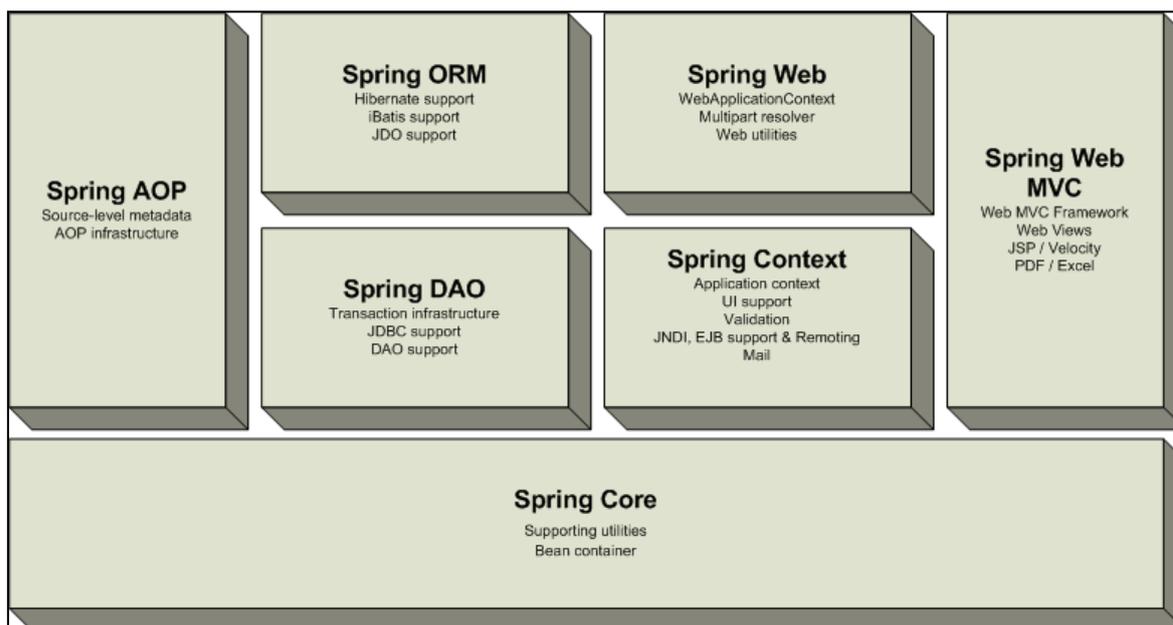


Figura 4.36 Spring: arquitectura en capas

4.4.3 Herramientas de desarrollo.

La herramienta escogida para desarrollar todo el código ha sido Spring Source Tool Suite en la versión 2.8.0 RELEASE ya que es una herramienta gratuita, basada en eclipse, y que incorpora todas las facilidades para implementar Spring, el servidor correspondiente, y las librerías que se utilizan.

Además se usa PhpMyAdmin, que es una herramienta escrita en PHP, con la intención de manejar la administración de MySQL a través de páginas Web, utilizando Internet. Actualmente permite la creación y eliminación de bases de datos, creación, eliminación y modificación de tablas, borrar, editar y añadir campos, ejecutar sentencias SQL, administrar claves en campos, administrar privilegios, y exportar e importar datos en varios formatos. Está disponible en 50 idiomas, y bajo licencia GPL.

4.5 Implantación y pruebas.

Esta fase de implantación junto con la del mantenimiento del software son partes muy importantes de la Ingeniería del software. Consisten en desplegar el software realizado y tratar de mejorar u optimizar cualquier problema encontrado.

El mantenimiento de software involucra cualquier tipo de pruebas del software realizadas, las cuales son un elemento crítico para la garantía de calidad del software y representan una revisión final de las especificaciones, del diseño y de la codificación.

4.5.1 Pruebas y validación.

La fase de pruebas es imprescindible en el desarrollo de cualquier software. Mediante este proceso conseguiremos garantizar un mínimo de calidad, minimizando la cantidad de fallos y analizando la usabilidad de la aplicación. Un buen sistema puede verse muy perjudicado por los errores no solucionados, por ello vamos a analizar las partes más críticas de nuestro sistema mediante las pruebas de validación. La calidad de la interfaz se evaluará mediante las pruebas de usabilidad.

El objetivo de esta fase es realizar un conjunto de pruebas sobre el sistema. Con esto intentaremos conseguir llegar a un sistema sin errores garantizando, como hemos dicho, la calidad del software. Para comprobar esto realizaremos unas pruebas de sistema.

Casos de Test. Pruebas caja negra.

Los test diseñados son los siguientes:

Autenticación

1. Autenticación de usuario

Requisitos Testeados	RF-1
Acción	Un administrador, introduce su nombre de usuario y contraseña
Checkpoint 1	El sistema muestra el menú principal.

2. Autenticación de usuario incorrecta.

Requisitos Testeados	RF-1
Acción	Un administrador, introduce erróneamente su nombre de usuario y contraseña. Deja alguno de los campos en blanco.
Checkpoint 1	El sistema muestra un mensaje de error informando de que el

	nombre de usuario o la contraseña son incorrectos.
--	--

Preprocesado expedientes***3. Añadir nuevo expediente de candidato.***

Requisitos Testeados	RF-2
Acción	Un administrador, quiere añadir un nuevo aspirante para que sea evaluado. El sistema reconoce solamente los ficheros que tengan el formato de Evaluacion-Nombre.xml
Checkpoint 1	El administrador sube un archivo que no cumple los requisitos. El sistema no lo procesará.

4. No se pueden recuperar los aspirantes. No hay ficheros en la Ruta especificada.

Requisitos Testeados	RF-2
Acción	En el preprocesado de los expedientes, se ha indicado una ruta inválida en la que no hay ficheros legibles por el sistema.
Checkpoint 1	El sistema muestra un mensaje de error informando de que en esa ruta, no hay ficheros.

5. Alguno de los criterios no corresponden con los que hay en el formulario.

Requisitos Testeados	RF-2
Acción	En el preprocesado de los expedientes, se ha seleccionado una Evaluación para poder leer los ficheros disponibles en la ruta, y los criterios que contienen, referentes al formulario de evaluación indicado para tal evaluación.
Checkpoint 1	Si durante la lectura de los ficheros, se producen incongruencias, es decir, hay criterios que están seleccionados para el formulario, pero el dato que contiene el fichero del

	<p>aspirante está mal relleno y no corresponde con un número, al finalizar el análisis del fichero, se mostrará una lista de errores, indicando el fichero, y el dato inválido para que se pueda tener constancia y corregirlo si es necesario.</p>
--	--

6. El fichero del aspirante no contiene la plaza a la que aspira.

Requisitos Testeados	RF-2
Acción	En el preprocesado de los expedientes, se ha seleccionado una Evaluación para poder leer los ficheros disponibles en la ruta, y los criterios que contienen, referentes al formulario de evaluación indicado para tal evaluación.
Checkpoint 1	Si no contiene plaza, el fichero no se procesa.

7. El fichero del aspirante no contiene datos válidos

Requisitos Testeados	RF-2
Acción	En el preprocesado de los expedientes, se ha seleccionado una Evaluación para poder leer los ficheros disponibles en la ruta, y los criterios que contienen, referentes al formulario de evaluación indicado para tal evaluación.
Checkpoint 1	El fichero contiene el tipo de plaza, pero los datos que aparecen no son correctos, es decir, no ha escrito números al rellenar los criterios que se indicaban. Al no poderlos procesar, se indica, que ha habido criterios en el fichero, pero aparecerán en la lista de errores.

Gestión del Formulario**8. Crear nuevo Formulario.**

Requisitos Testeados	RF-4
Acción	El administrador en la opción de Gestionar Formulario, crea un nuevo formulario.
Checkpoint 1	El formulario es creado y añadido a la lista de Formularios Disponibles.
Checkpoint 2	Si el administrador deja en blanco el nombre del formulario, el sistema le indicará que deberá darle un nombre. No es un error crítico, puesto que se podría indicar el nombre posteriormente editándolo.

9. Asignar nuevo Formulario

Requisitos Testeados	RF-4
Acción	El administrador en la opción de Gestionar Formulario, asigna un formulario ya creado, a una Evaluación.
Checkpoint 1	El Formulario aparece asignado a la Evaluación seleccionada.
Checkpoint 2	Si no hay ninguna Evaluación activa, no se podrá asignar el Formulario, y el sistema indicará un error, diciendo que no se ha podido asignar el Formulario.

10. Recuperar un Formulario.

Requisitos Testeados	RF-4
Acción	El administrador en la opción de Gestionar Formulario, recupera un formulario ya creado.
Checkpoint 1	Aparecen los Criterios y los pesos que previamente se habían añadido al formulario.
Checkpoint 2	Si no hay Criterios añadidos al Formulario, el sistema no muestra nada. Deja visibles los criterios disponibles en el

	sistema, para que el administrador los seleccione y los pueda añadir.
--	---

11. Modificar el nombre del Formulario.

Requisitos Testeados	RF-4
Acción	El administrador en la opción de Gestionar Formulario, modifica el nombre a un formulario ya creado.
Checkpoint 1	El nombre anterior queda cambiado por el nuevo.
Checkpoint 2	Si lo deja en blanco, el sistema le indicará un error indicando que deberá escribir algo.

12. Añadir Criterios al Formulario.

Requisitos Testeados	RF-4
Acción	El administrador en la opción de Gestionar Formulario, selecciona un formulario, y le da a Ver. Posteriormente, añade criterios al formulario, asignándole un peso.
Checkpoint 1	Si el peso no es correcto, o no es un número, el sistema se lo indicará.
Checkpoint 2	Si la opción elegida no es correcta de entre los criterios, el sistema lo indicará también.

13. Borrar Criterios del Formulario.

Requisitos Testeados	RF-4
Acción	El administrador en la opción de Gestionar Formulario, selecciona un Formulario, y le da a Ver. Posteriormente, elimina Criterios del Formulario.
Checkpoint 1	Se pide confirmación de borrado. Si se queda vacío, el sistema no muestra ningún criterio asignado al Formulario.

14. Borrar Formulario.

Requisitos Testeados	RF-4
Acción	El administrador en la opción de Gestionar Formulario, selecciona un Formulario, y le da a Borrar.
Checkpoint 1	Se pide confirmación de borrado. Si está siendo usado, no se borrará.
Checkpoint 2	Si no está siendo usado, desaparecerá de la lista.

Gestión de Plazas**15. Nueva plaza.**

Requisitos Testeados	RF-5
Acción	El administrador en la opción de Gestionar Formulario, selecciona crear una Nueva Plaza.
Checkpoint 1	La plaza queda creada con el nombre indicado.
Checkpoint 2	Si el nombre queda vacío, el sistema indica un error para que el administrador rellene el nombre.
Checkpoint 3	Si el nombre es demasiado largo, el sistema indica error.

16. Modificar Nombre de la Plaza.

Requisitos Testeados	RF-5
Acción	El administrador en la opción de Gestionar Formulario, selecciona modificar el nombre de una Plaza.
Checkpoint 1	El anterior nombre de la plaza, es sustituido por el nuevo.
Checkpoint 2	Si se queda vacío, el sistema muestra un error para que el administrador rellene el nombre.
Checkpoint 3	Si el nombre es demasiado largo, el sistema indica error.

17. Borrar Plaza.

Requisitos Testeados	RF-5
Acción	El administrador en la opción de Gestionar Formulario, selecciona Borrar una Plaza.
Checkpoint 1	Si aceptamos la confirmación de borrado, la plaza queda borrada.
Checkpoint 2	Se pide confirmación de borrado. Si se acepta, si la plaza está en uso, es decir, pertenece a algún cálculo en alguna Evaluación, el sistema para evitar incongruencias, muestra un error diciendo que esa plaza no se puede borrar.

18. Añadir Categoría. (peso Erróneo, Evaluación no seleccionada)

Requisitos Testeados	RF-5
Acción	El administrador en la opción de Gestionar Formulario, añade una Categoría a una Plaza.
Checkpoint 1	Si hay una Evaluación seleccionada, y el peso es correcto, la Categoría queda añadida a la plaza.
Checkpoint 2	Si hay un error en el peso, el sistema indica el error.
Checkpoint 3	Si la Evaluación no está seleccionada, el sistema no permite añadir la Categoría.
Checkpoint 4	Si el nombre es demasiado largo, el sistema indica error.

19. Eliminar Categoría de una Plaza

Requisitos Testeados	RF-5
Acción	El administrador en la opción de Gestionar Formulario, selecciona eliminar categorías de una Plaza.
Checkpoint 1	Se requiere una confirmación de borrado, y si se acepta, la Categoría desaparece de la plaza, en esa Evaluación.

Gestión de los Elementos de Evaluación***20. Nuevo en Categoría***

Requisitos Testeados	RF-6
Acción	El administrador en la opción de Gestionar Elementos de Evaluación, selecciona añadir una nueva Categoría.
Checkpoint 1	La categoría queda añadida con los datos indicados.
Checkpoint 2	Si deja datos incompletos, el sistema le indica un error.
Checkpoint 3	Si el identificador de la Categoría no es una letra seguida de un punto. "L." el sistema indica error.
Checkpoint 4	Si el nombre es demasiado largo, el sistema indica error.
Checkpoint 5	Si el identificador ya existe, se indica el error.

21. Modificar Categoría.

Requisitos Testeados	RF-6
Acción	El administrador en la opción de Gestionar Elementos de Evaluación, selecciona Modificar una Categoría.
Checkpoint 1	La categoría queda modificada con los datos indicados.
Checkpoint 2	Si deja datos incompletos, el sistema le indica un error.

22. Borrar Categoría.

Requisitos Testeados	RF-6
Acción	El administrador en la opción de Gestionar Elementos de Evaluación, selecciona Borrar una Categoría.
Checkpoint 1	Tras la confirmación del borrado, si la Categoría está siendo usada por alguna Evaluación, se indica que está en uso y que no puede ser borrada, para evitar incongruencias posteriores.
Checkpoint 2	Se borran en caso de que no esté siendo usada, todas sus subcategorías y criterios que dependan de esa categoría.

23. Nueva Subcategoría.

Requisitos Testeados	RF-6
Acción	El administrador en la opción de Gestionar Elementos de Evaluación, selecciona añadir Nueva Subcategoría.
Checkpoint 1	La categoría queda añadida con el identificador y el nombre indicados.
Checkpoint 2	Si el identificador de la Subcategoría no cumple el formato "L.N." siendo L una letra, y N un número, se indica error.
Checkpoint 3	Si La categoría, indicada por L. no existe, se indica error.
Checkpoint 4	Si el nombre es demasiado largo, el sistema indica error.
Checkpoint 5	Si se deja en blanco algún campo, se indica error.
Checkpoint 6	Si el identificador ya existe, se indica el error.

24. Modificar nombre de Subcategoría.

Requisitos Testeados	RF-6
Acción	El administrador en la opción de Gestionar Elementos de Evaluación, selecciona Modificar el nombre de una Subcategoría.
Checkpoint 1	El nombre queda modificado por el indicado.
Checkpoint 2	Si el nombre es demasiado largo, el sistema indica error.
Checkpoint 3	Si se deja en blanco algún campo, se indica error.

25. Borrar Subcategoría.

Requisitos Testeados	RF-6
Acción	El administrador en la opción de Gestionar Elementos de Evaluación, selecciona Borrar una Subcategoría.
Checkpoint 1	Tras la confirmación del borrado, si la Subcategoría está siendo usada por alguna Evaluación, se indica que está en uso y que no puede ser borrada, para evitar incongruencias posteriores.

Checkpoint 2	Se borran en caso de que no esté siendo usada, todos los criterios que dependan de esa subcategoría.
--------------	--

26. Nuevo Criterio.

Requisitos Testeados	RF-6
Acción	El administrador en la opción de Gestionar Elementos de Evaluación, selecciona añadir Nuevo Criterio.
Checkpoint 1	El criterio queda añadido con el identificador y el nombre indicados.
Checkpoint 2	Si el identificador del Criterio no cumple el formato “L.N.N.” siendo L una letra, y N un número, se indica error.
Checkpoint 3	Si la subcategoría, indicada por L.N. no existe, se indica error.
Checkpoint 4	Si el nombre es demasiado largo, el sistema indica error.
Checkpoint 5	Si se deja en blanco algún campo, se indica error.
Checkpoint 6	Si el identificador ya existe, se indica el error.

27. Modificar nombre de Criterio.

Requisitos Testeados	RF-6
Acción	El administrador en la opción de Gestionar Elementos de Evaluación, selecciona Modificar el nombre de un Criterio.
Checkpoint 1	El nombre del criterio es cambiado por el nuevo.
Checkpoint 2	Si el nombre es demasiado largo, el sistema indica error.
Checkpoint 3	Si se deja en blanco algún campo, se indica error.

28. Borrar Criterio.

Requisitos Testeados	RF-6
Acción	El administrador en la opción de Gestionar Elementos de Evaluación, selecciona Borrar un Criterio.
Checkpoint 1	Tras la confirmación del borrado, si el Criterio está siendo usado por alguna Evaluación, se indica que está en uso y que no puede

Checkpoint 2	<p>ser borrado, para evitar incongruencias posteriores.</p> <p>Se borran en caso de que no esté siendo usado.</p>
--------------	---

Cálculos y Resultados

29. Calcular Puntuaciones. Si solo hay un candidato.

Requisitos Testeados	RF-7
Acción	<p>El administrador en la opción de Evaluar Aspirantes selecciona calcular las puntuaciones.</p> <p>Para ello, ha seleccionado una Evaluación, y un Tipo de Plaza.</p> <p>Si no está seguro si los datos han sido modificados, antes de Calcular, seleccionará la opción de Refrescar.</p>
Checkpoint 1	Si todo va bien, se mostrarán los resultados.
Checkpoint 2	Si no hay criterios para la selección, se indicará el error.
Checkpoint 3	Si solo hay un candidato, no se puede calcular Topsis, por lo que se mostrará la puntuación obtenida multiplicando la ponderación de cada apartado, por su peso.
Checkpoint 4	Si solo hay candidatos empatados a puntos, no se puede calcular Topsis, y se procede de la misma manera que antes.
Checkpoint 5	Si no hay Formulario en esa Evaluación, se indica error.
Checkpoint 6	Si no hay categorías que correspondan a los criterios seleccionados para el Formulario, no se tendrán en cuenta para el cálculo, y no aparecerán en los gráficos ni en los resultados.

Consultar datos de los Aspirantes

30. Consultar sobre los candidatos.

Requisitos Testeados	RF-8
Acción	<p>El administrador en la opción de Evaluar Aspirantes ve los datos de los Aspirantes.</p>

Checkpoint 1	Para ello, ha seleccionado una Evaluación, y un Tipo de Plaza. Si no está seguro si los datos han sido modificados, seleccionará la opción de Refrescar. Si todo va bien, se mostrarán los datos de los Aspirantes. Se indica el fichero al que corresponden para que sea más fácil identificarlos.
Checkpoint 2	Si no hay criterios en el Formulario seleccionado para esa Evaluación, se indicará el error.
Checkpoint 3	Si no hay Formulario en esa Evaluación, se indica error.

Resultados obtenidos.

A continuación mostramos una tabla con los resultados obtenidos, después de realizar los test diseñados sobre nuestra aplicación.

<i>Test 1</i>	
Checkpoint 1	OK
<i>Test 2</i>	
Checkpoint 1	OK
<i>Test 3</i>	
Checkpoint 1	OK
<i>Test 4</i>	
Checkpoint 1	OK
<i>Test 5</i>	
Checkpoint 1	OK
<i>Test 6</i>	
Checkpoint 1	OK
<i>Test 7</i>	
Checkpoint 1	OK
<i>Test 8</i>	
Checkpoint 1	OK

Checkpoint 2	OK
<i>Test 9</i>	
Checkpoint 1	OK
Checkpoint 2	OK
<i>Test 10</i>	
Checkpoint 1	OK
Checkpoint 2	OK
<i>Test 11</i>	
Checkpoint 1	OK
Checkpoint 2	OK
<i>Test 12</i>	
Checkpoint 1	OK
Checkpoint 2	OK
<i>Test 13</i>	
Checkpoint 1	OK
<i>Test 14</i>	
Checkpoint 1	OK
Checkpoint 2	OK
<i>Test 15</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 16</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 17</i>	
Checkpoint 1	OK
Checkpoint 2	OK
<i>Test 18</i>	
Checkpoint 1	OK

Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 19</i>	
Checkpoint 1	OK
<i>Test 20</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
Checkpoint 5	OK
<i>Test 21</i>	
Checkpoint 1	OK
Checkpoint 2	OK
<i>Test 22</i>	
Checkpoint 1	OK
Checkpoint 2	OK
<i>Test 23</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
Checkpoint 5	OK
Checkpoint 6	OK
<i>Test 24</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 25</i>	
Checkpoint 1	OK
Checkpoint 2	OK

<i>Test 26</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
Checkpoint 5	OK
Checkpoint 6	OK
<i>Test 27</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 28</i>	
Checkpoint 1	OK
Checkpoint 2	OK
<i>Test 29</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
Checkpoint 5	OK
Checkpoint 6	OK
<i>Test 30</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK

Figura 4.37. Resultados de las pruebas de Test.

Capítulo 5:

Conclusiones

5.1 Conclusiones

El propósito de nuestro proyecto, era la construcción de una herramienta web, para hacer que el proceso de sistema de selección de personal sustituto en la Universidad de Jaén, fuera lo más automatizado, y lo más independiente posible de la fuente desde donde se recogieran los datos de los aspirantes. Para ello, se ha construido un sistema Web, basado en un algoritmo de decisión multicriterio, que proporciona una valoración objetiva de los aspirantes, dependiendo de los puntos que se quieran evaluar en cada momento.

Hay que tener en cuenta dos aspectos fundamentales en los que se ha basado el desarrollo de este proyecto:

- i. Por un lado, la reducción del factor de la subjetividad a la hora de evaluar a un candidato. Para ello, se ha procurado, establecer una serie de puntuaciones, y un número elevado de criterios en los que el aspirante puede reflejar sus conocimientos, experiencia, y bagaje anteriores.
- ii. Por otro lado, tenemos la posibilidad, de que el sistema que se ha construido, sea independiente de la fuente desde la que se recojan los datos. Es decir, podríamos utilizar un sistema externo, que validara los datos, y con una interfaz amigable para el usuario, un formulario que el usuario rellenara, la introducción de los datos que proporcionara el usuario en formato de papel, por un administrador externo, etc. Para esta finalidad, se ha creado un esquema XML, que es el que ha de ser respetado para que el sistema pueda leer los datos. Así, conseguimos la independencia que se ha comentado anteriormente.

Creemos que los objetivos planteados en el capítulo 1 se han cumplido, y además, se ha conseguido un sistema basado en la arquitectura cliente/servidor, que hace que el procesamiento se haga de manera externa, en el servidor, y así poder hacer una separación clara del procesamiento y de la interfaz.

Asimismo, se ha introducido una arquitectura Modelo Vista Controlador, en la que se podrían basar futuras mejoras, como comentaremos posteriormente.

Como conclusiones personales, decir que la utilización del proceso incremental de la ingeniería de software en un sistema “real”, me ha servido para aplicar los conocimientos obtenidos a través de estos años de estudio, para poder superar los problemas que han ido surgiendo, así como para aprender que hay herramientas que nos facilitan mucho el proceso de programación, como son el uso de Frameworks, como en este caso, Spring MVC para Java, con el que podemos crear sistemas Web, de manera relativamente sencilla. El aprendizaje de esta tecnología ha sido laborioso, ya que en algunos aspectos no sigue la línea habitual que tenemos establecida para la llamada de métodos y el sistema de anotaciones con el que cuenta, también ha sido nuevo.

El profundizar en los esquemas para XML, así como definir uno que cumpliera todos los requisitos que queríamos plasmar, ha sido otro aprendizaje que llevó una parte importante del proceso de desarrollo. Se puede incluir en este punto, la validación de los XML y la lectura de los mismos de manera lo más eficiente posible a través de Java.

5.2 Futuras Mejoras

Las mejoras que se plantean en un futuro, vienen dadas por la integración del sistema actual, con otro externo, para la introducción de datos de los aspirantes, como se ha comentado anteriormente.

Nos consta, que se realizó un proyecto anterior, que consistía en la validación de los datos para la base de datos de aspirantes para la Universidad de Jaén. Planteamos como mejora, la unión de ambos, para conseguir un sistema completo, y una unificación del proceso de selección, desde que se tienen los datos, hasta que se hace la valoración completa.

Otra variación, sería que el propio sistema, a través del esquema que se utiliza, creara un formulario de entrada de datos que se le presentaría al aspirante, de manera anexa al sistema que se plantea.

Bibliografía

-
- [1] Chiavenato, Idalberto. *Gestión del Talento Humano*. Colombia : Editorial Mc Graw Hill, 2002.
- [2] Doyle, Christie E. *Selección de personal y Evaluación*. s.l. : Psychology Press, 2002.
- [3] PEREDA MARÍN, Santiago y BERROCAL BERROCAL, Francisca. Técnicas de Análisis y Descripción de Puestos en la Gestión por Competencias. [En línea] abril de 2003.
http://sorad.ual.es/mitra/documentos/tec_peredao3/ALMER%C3%8DA%20PUESTOS.pdf.
- [4] Vicerrectorado de Ordenación Académica, Innovación Docente y Profesorado. *Criterios Generales y Baremo para la Valoración del Mérito y la Capacidad en los Concursos Públicos de Selección de PDI que se convocan en la Universidad de Jaén*. JAÉN : UNIVERSIDAD DE JAÉN, 2005.
- [5] Lopez González, E. ,Mendaña Cuervo, C., Rodríguez Fernández, M.A. La selección de personal con un algoritmo genético borroso. *Investigaciones Europeas de Dirección y Economía de la Empresa*. León : Universidad de León, 1996, págs. 61-76.
- [6] García Cascales, M^a del Socorro. Métodos para la comparación de alternativas mediante un Sistema de Ayuda a la Decisión (S.A.D.) y “Soft Computing”. Cartagena : UNIVERSIDAD POLITÉCNICA DE CARTAGENA, 2009.
- [7] W3C. Guía Breve de Tecnologías XML. [En línea] Febrero de 2012.
<http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>.
- [8] Hwang, C. L. y Yoon, K. *Multiple Attribute Decision Making: Methods and Applications*. New York : Springer-Verlag, 1981.
- [9] Crecenegocios. Crecenegocios.com. [En línea]
<http://www.crecenegocios.com/concepto-y-funciones-del-area-de-recursos-humanos/>.
- [10] Cuesta, A. *Tecnología de Gestión de recursos Humanos*. La Habana : Editorial ISPJAE, 1997.

- [11] Chiavenato, Idalberto. *INTRODUCCIÓN A LA TEORÍA GENERAL DE LA ADMINISTRACIÓN*. s.l. : Mc Graw - Hill.
- [12] La web de los recursos humanos y el empleo. [En línea] 2006. <http://www.rrhh-web.com/reclutamientoyseleccion.html>.
- [13] Rodríguez, López J.L. *La entrevista en la empresa*. Madrid : Eudema, 1993.
- [14] Chernoff, H. *Elementary Decision Theory*. Nueva York : Dover Publications, 1987.
- [15] Chen, S. J. y Hwang, C. L. *Fuzzy Multiple Attribute Decision Making: Methods and Applications*. Berlín/Nueva York : springer-Verlag, 1992.
- [16] Fodor, J. y Roubens, M. *Fuzzy Preference Modeling and Multicriteria Decision Support*. Dordrecht : Kluwer Academic Publisher, 1994.
- [17] Duncan, R. y Raiffa, H. *Games and Decision. Introduction and Critical Survey*. Nueva York : Dover Publications, 1985.
- [18] Bouyssou, D., y otros. *Evaluation and Decision Models: A Critical Perspective*. s.l. : Kluwer Academic Publishers, 2000.
- [19] Doyle, J. *Prospects for preferences. Computational Intelligence*,. 2004.
- [20] Orlovski, S. A. *Decision-making with fuzzy preference relations. Fuzzy Sets and Systems*. 1978.
- [21] de Andrés, R., García-Lapresta, J. L. y Martínez, Luis. *A multi-granular linguistic model for management decision-making in performance appraisal*.
- [22] Caballero, R. y Fernández, G. M. *Toma de Decisiones con Criterios Múltiples*. s.l. : SEPUMA, 2002. 84-8442-630-0.
- [23] T. Ray, Erik. *Learning XML*. s.l. : O'REILLY, 2003.
- [24] Melero, JJ. Introducción al lenguaje XML. [En línea] Universidad de Granada, 2004. geneura.ugr.es/~jmelero/xml.

- [25] Harold, Elliotte Rusty. *XML Bible*. 2001.
- [26] Reino Romero, Alfredo. Introducción a XML en castellano. [En línea] 26 de Enero de 2000. www.ibium.com/alf/xml/index.asp.
- [27] W3c. W3c. [En línea] <http://www.w3schools.com/dtd/default.asp>.
- [28] Microsoft. Características y ventajas de InfoPath . [En línea] Febrero de 2012. <http://office.microsoft.com/es-es/infopath/caracteristicas-y-ventajas-de-infopath-2010-HA101806949.aspx>.
- [29] McLaughli, Brett D., Pollice, G. y West, D. *Head First Object-Oriented Analysis and Design*. s.l. : O'Reilly.
- [30] Conde, F. *Cómo desarrollar una buena interfaz de usuario... y no morir en el intento*. Jaén : s.n., 2011.
- [31] Feito Higuera, F., Ruiz de Miras, J., Molina Aguilar, A. *Análisis y Gestión de Datos*. s.l. : Universidad de Jaén, 1996.
- [32] Krug, S. *No me hagas pensar: una aproximación a la Usabilidad en la Web*. s.l. : Prentice Hall.
- [33] Spring FrameWork. [En línea] 15 de Diciembre de 2011. www.springframework.com.
- [34] Dos ideas sobre SpringFramework. [En línea] Diciembre de 2011. http://www.dosideas.com/wiki/Spring_Framework.

Anexo I.

Manual de Instalación

Este anexo está dedicado a realizar las configuraciones necesarias para poner en marcha nuestra aplicación.

En la instalación de las aplicaciones y del sistema en el servidor, se ha supuesto que se va a realizar en un entorno Windows, por lo que todas las imágenes de las instalaciones y la configuración de las aplicaciones se han hecho basándose en este Sistema Operativo.

Teniendo en cuenta las aplicaciones que van a ser instaladas en este manual, se debe de contar en el equipo en el que se realice la instalación del servidor del sistema con los siguientes requisitos mínimos.

Requerimientos de Hardware:

- Procesador: Pc (x86) 32 bits mínimo.
 - Recomendado Core 2 Duo
- Memoria RAM: 512MB (mínimo)
 - Recomendado 1 GB
- Espacio en disco: 512MB de espacio libre
 - Recomendado 1GB

Requerimientos de Software:

- Windows XP/Vista/7/8 (32 bits)
- Privilegios de administrador

Las aplicaciones necesarias para todo el proceso de instalación y configuración, así como la base de datos, están incluidas en el DVD que acompaña a esta memoria.

En el directorio \Aplicaciones podrá encontrar:

- Jdk de Java en su versión 7
- Apache-tomcat 7
- Wamp server 2
- Springapp.war que contendrá la implementación de la aplicación

- Fichero .sql que contendrá los datos necesarios para la base de datos.
- Script para la instalación de la base de datos y creación del usuario para el manejo de la misma.

Si cuenta en su sistema con alguna de estas aplicaciones en la versión que se menciona, no será necesario volverla a instalar.

Consideraciones previas

Es posible, y de hecho normalmente se realiza así, la instalación de cada uno de los citados elementos por separado. No obstante, hemos optado por la instalación de **WAMP server**, un paquete software que integra todas las aplicaciones, a excepción del entorno de desarrollo y la máquina virtual de Java.

Instalación de WAMP-Server

Se puede descargar desde <http://www.wampserver.com/en/> habiendo sido usada en el desarrollo de este proyecto, la versión de 32 bits, y PHP 5.3 (Figura I.1), no siendo necesaria ninguna versión especial de PHP, se ha escogido, por tener la versión de MySQL principalmente que se necesitaba.

Las versiones que tenemos en este paquete son las siguientes:

- Apache 2.2.22
- Mysql 5.5.24
- PHP 5.3.13
- PhpMyadmin 3.4.10.1
- Y otras utilidades como XDebug 2.1.2 ,XDC 1.5 ,SQLBuddy 1.3.3 ,webGrind 1.0

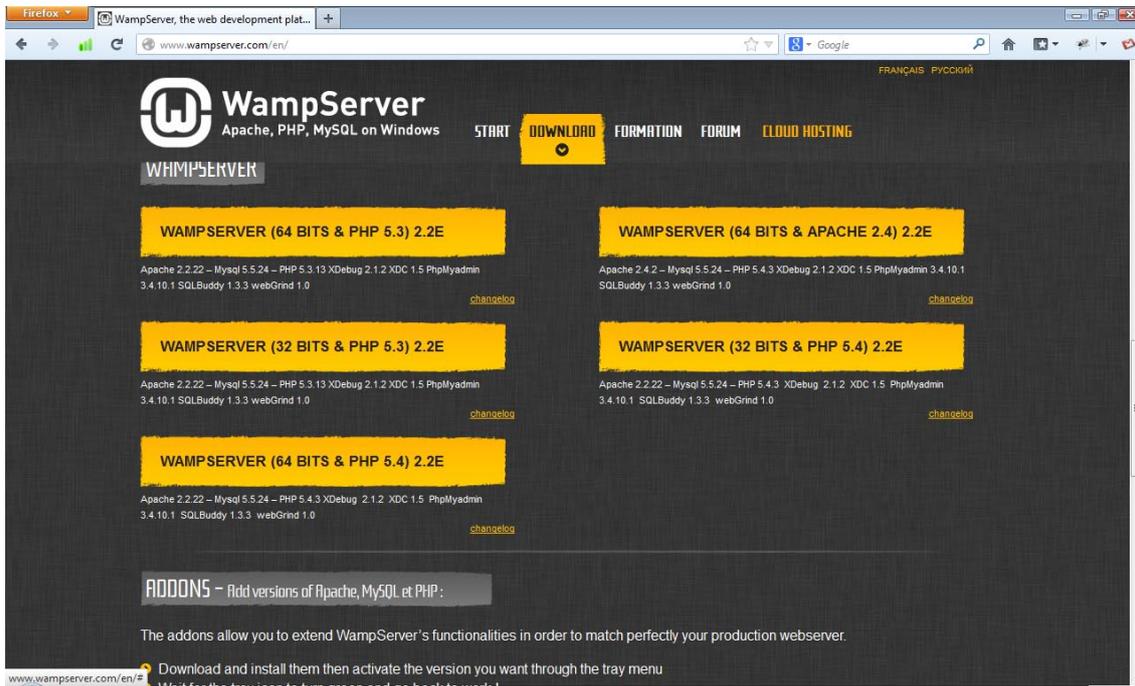


Figura I.1 Página de descarga WAMP Server

Una vez hacemos clic en el enlace, obtenemos lo siguiente (Figura I.2):

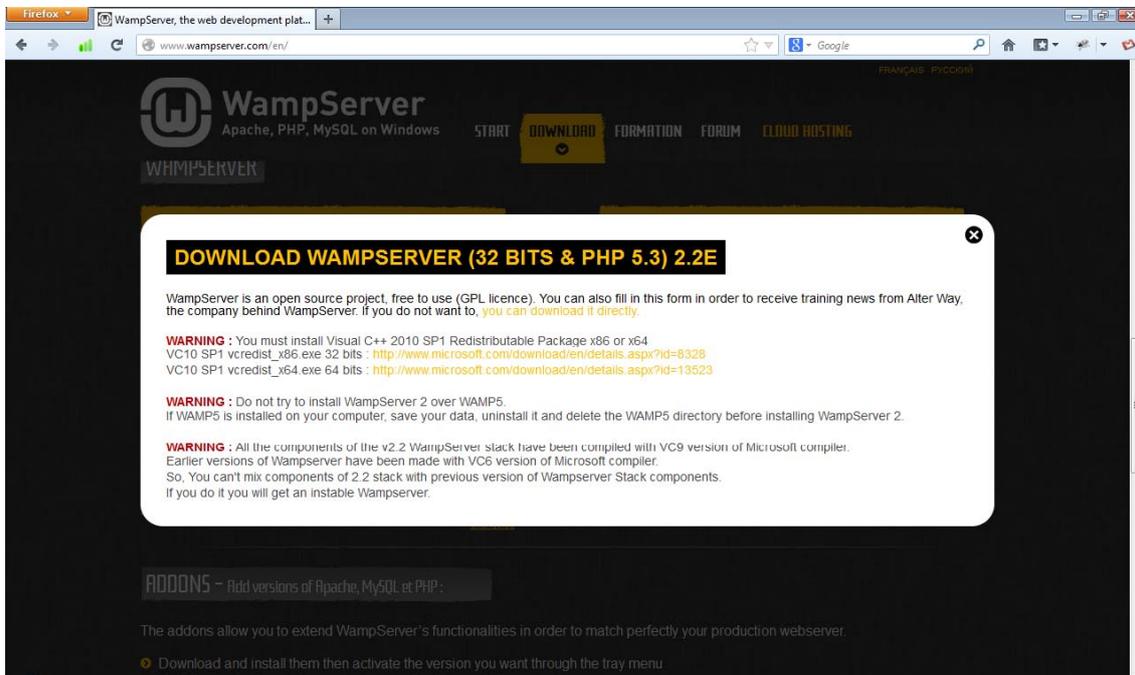


Figura I.2. Página descarga WAMP server. Opción 32 Bits

Pulsando en el enlace, y eligiendo una ruta de destino en el equipo para el instalador, comienza la descarga. Al finalizar, lo ejecutaremos, haciendo clic en `wampserver2.2e-php5.3.13-httpd2.2.22-mysql5.5.24-32b.exe` e iniciaremos la instalación como se indica en la siguiente Figura I.3 .

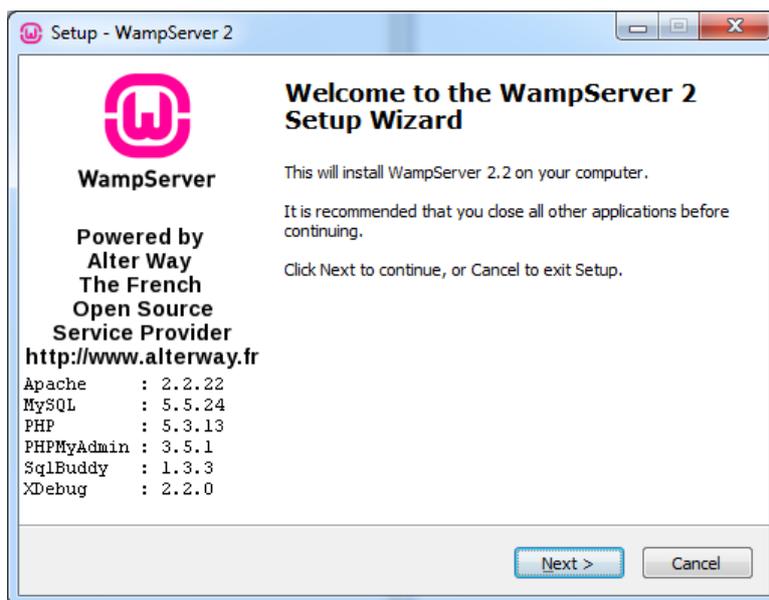


Figura I.3. Página Bienvenida Instalación WAMP Server

Pulsamos en siguiente, y se pasa al paso donde aparece la licencia(Figura I.4).

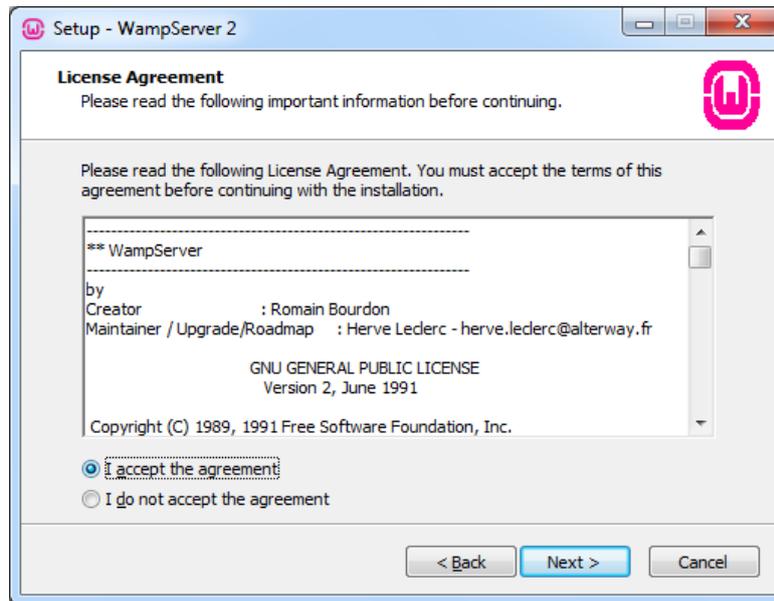


Figura I.4. Instalación Servidor WAMP. Licencia.

Tras leer la licencia, y pasar al siguiente paso (Next >) pasamos a indicar la ruta donde queremos instalar la aplicación (Figura I.5).

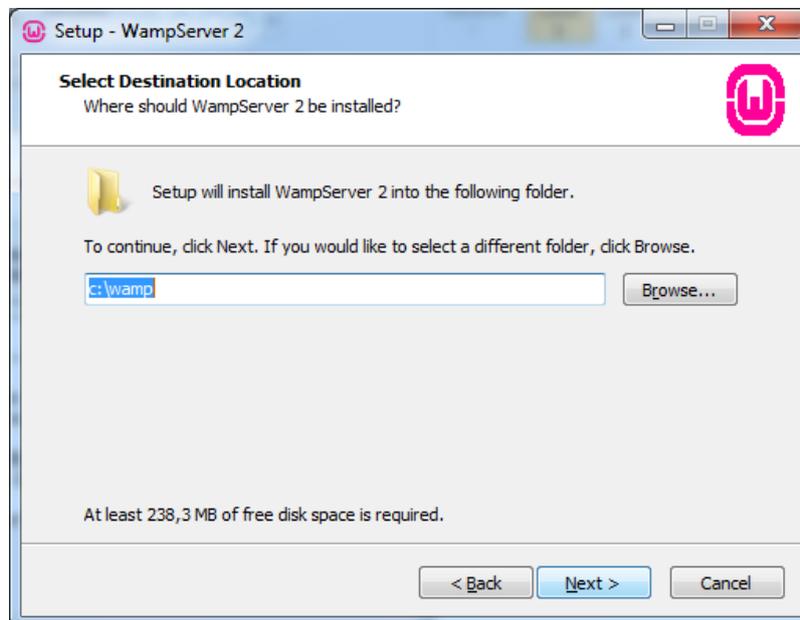


Figura I.5. Instalación Servidor WAMP. Ruta de Destino.

En nuestro caso, se ha obtenido por dejar la ruta por defecto (c:/wamp)

Si fuera necesario cambiarla, en el botón Browse.. Podríamos escoger la ruta que quisiéramos. Pulsamos en Siguiente (Next >) y obtenemos la Figura I.6:

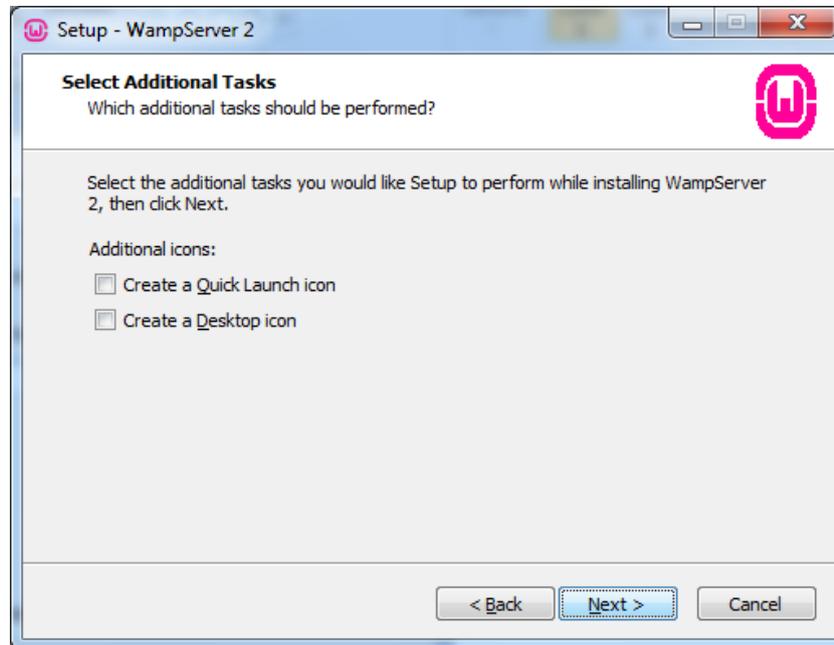


Figura I.6. Instalación Servidor WAMP. Tareas Adicionales.

Una vez hecho esto, pulsamos en siguiente (Next >) y obtenemos la siguiente Figura I.7.

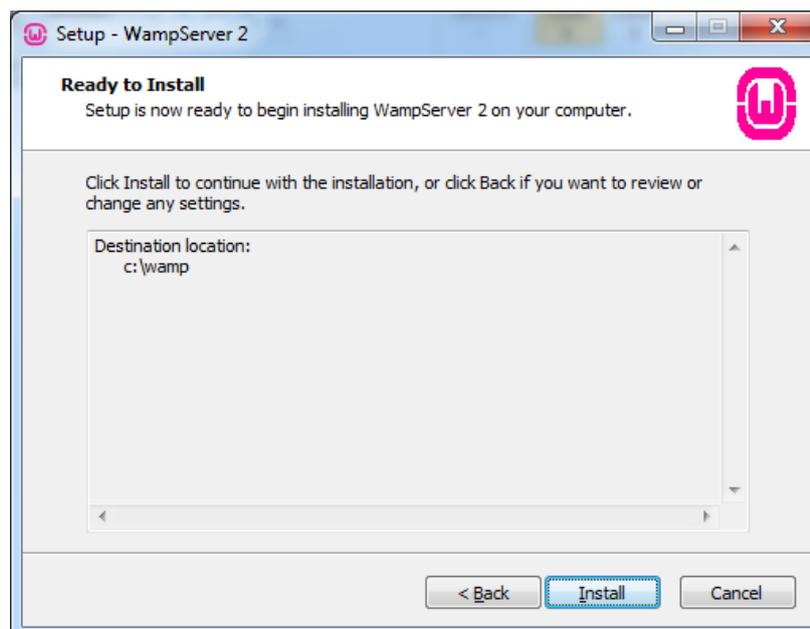


Figura I.7. Instalación Servidor WAMP. Instalar.

Después de comprobar que las opciones marcadas son correctas, pulsamos en Instalar (Install) y comienza la instalación (Figura I.8) :

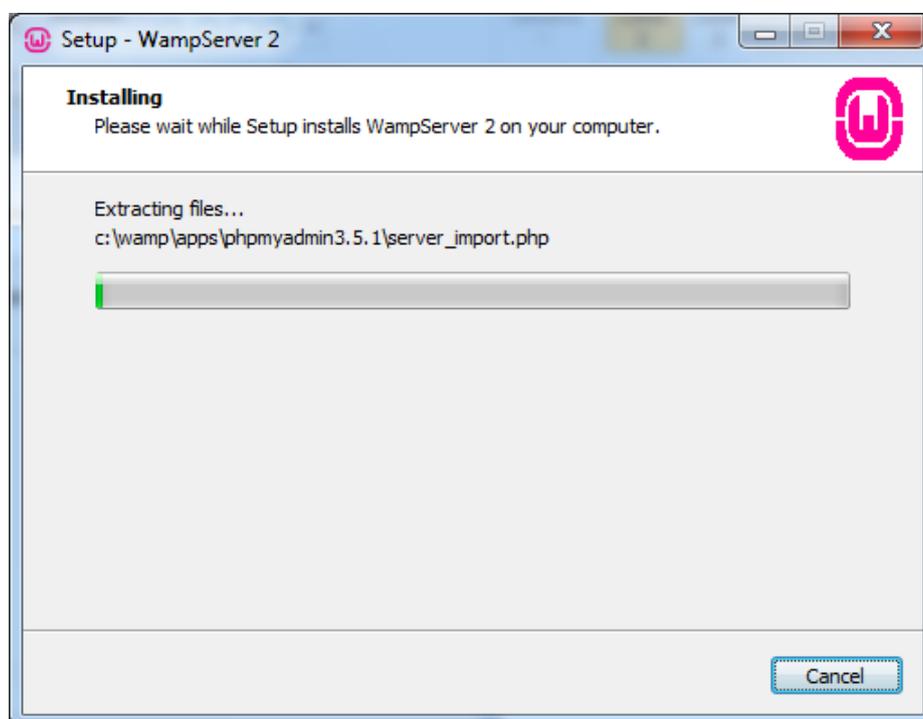


Figura I.8. Instalación servidor WAMP. Comienzo de la instalación.

Transcurrido un tiempo, pasamos a la configuración de PHP, básicamente en nuestro caso servirá para poder mostrar el administrador de la Base de Datos PHPMyAdmin, ya que PHP no es usado en nuestro sistema.

Indicaremos los parámetros SMTP y Email. En nuestro caso, dejaremos localhost, y el email por defecto, ya que no es necesaria ninguna configuración adicional (Figura I.9).

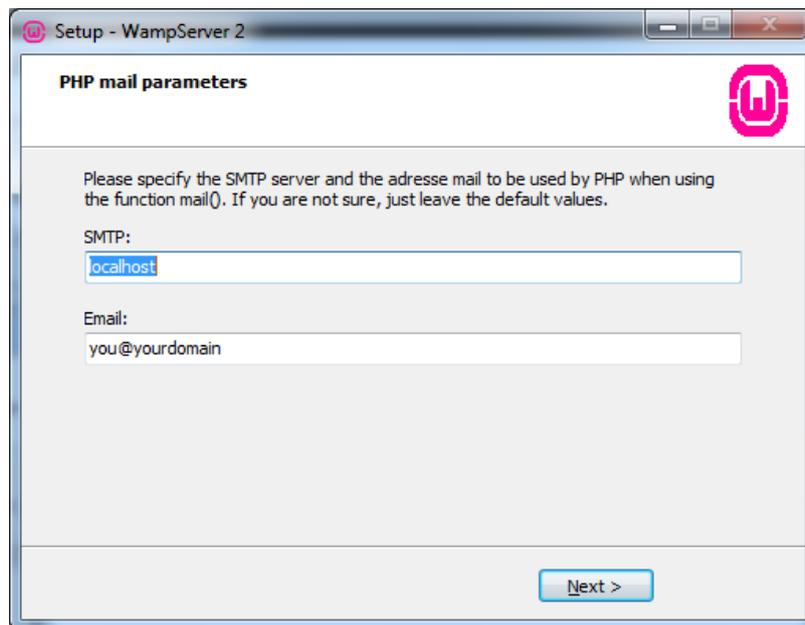


Figura I.9. Instalación Servidor WAMP. Parámetros PHP

Pulsando en Siguiente (Next >), tenemos la pantalla de fin de instalación. Hacemos clic en Finalizar (Finish), y obtenemos la Figura I.10:

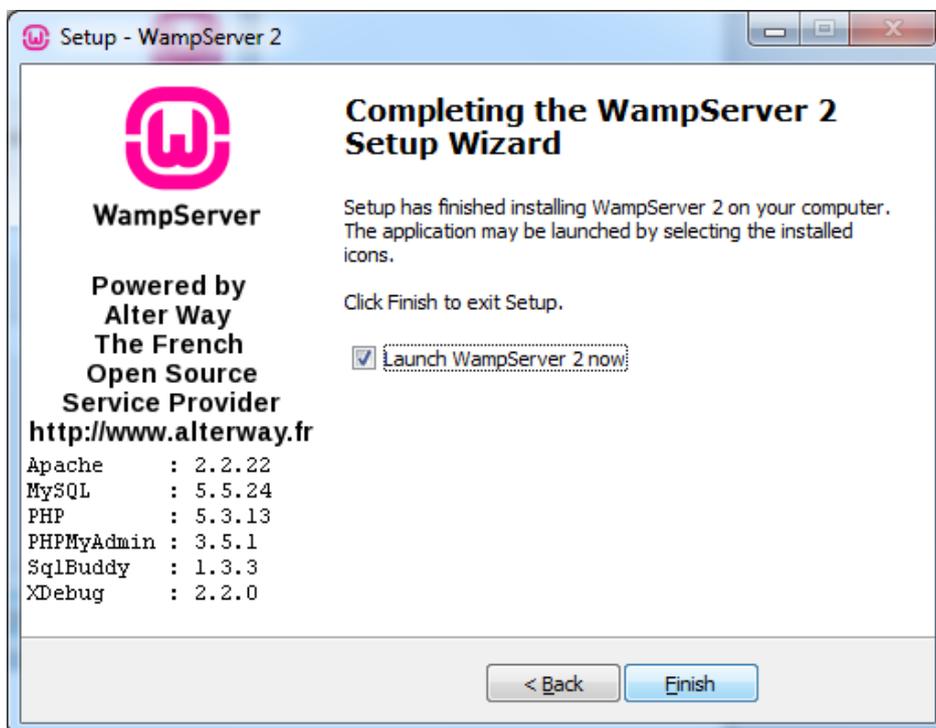


Figura I.10. Instalación Servidor WAMP. Pantalla Final.

Es posible, que finalizada la instalación, y al querer arrancar el servicio, obtengamos una advertencia de seguridad del Firewall de Windows. Solamente tendremos que permitirle el acceso. En la siguiente Figura I.11 se indican las opciones a marcar:

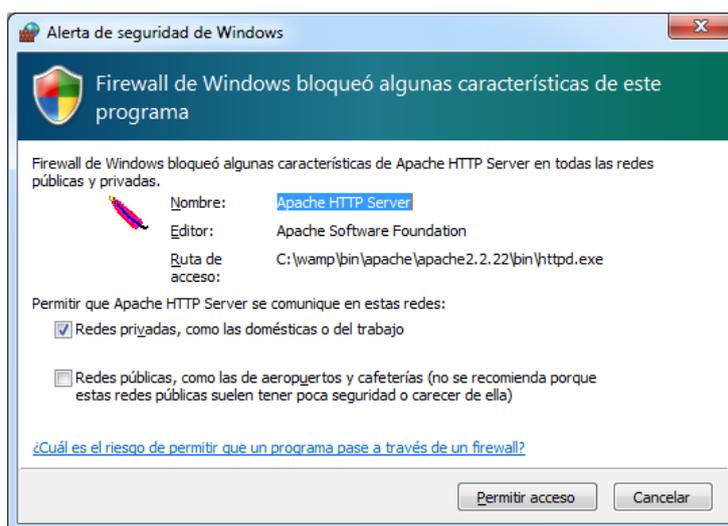


Figura I.11. Alerta de Seguridad Firewall. Apache.

Una vez terminado, tendremos un icono ver Figura I.12 , en la barra de estado, que nos indicará, si todo ha ido bien y es de color verde, que nuestro servidor PHP y MySQL está funcionando.

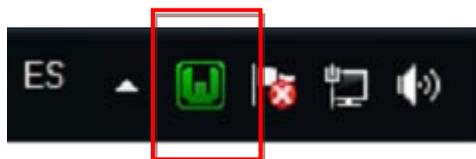
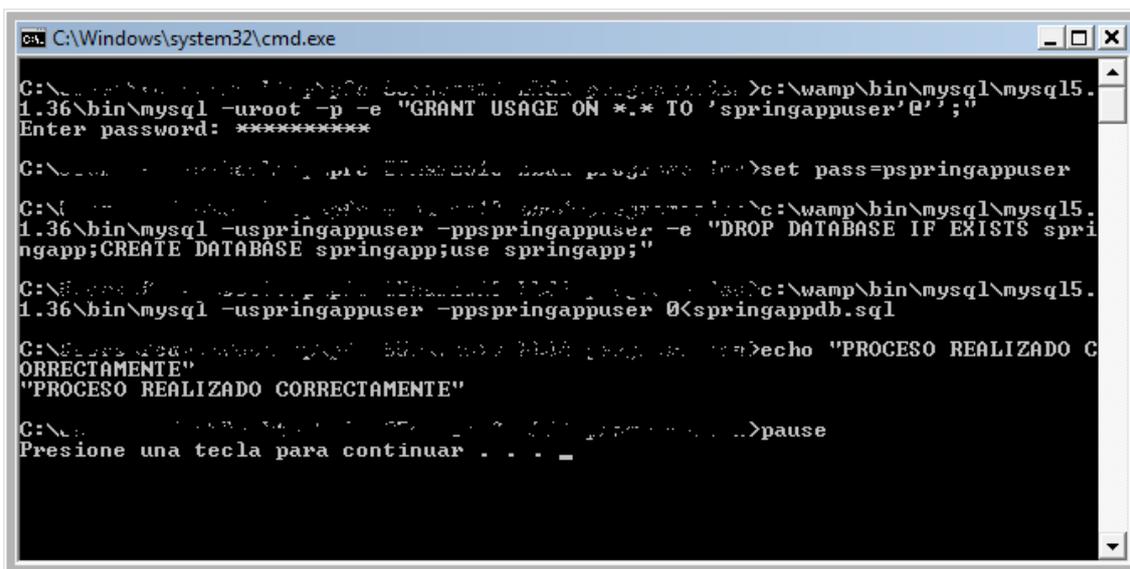


Figura I.12. Icono de Estado servidor WAMP.

Ultimada la instalación, pasamos a configurar la Base de Datos. Para ello, se cuenta con un script de instalación, usando los archivos por lotes de Windows para facilitar el proceso de instalación. En la carpeta de “aplicaciones” debemos de buscar el archivo:

Scriptbd.bat

Tras hacer doble clic en él, se abrirá una consola, en la que deberemos de escribir la contraseña de “root” que hayamos indicado en el proceso de instalación anterior. Una vez finalizada la creación del usuario, así como la importación de la base de datos, obtendremos el siguiente resultado (ver Figura I.13):



```
C:\Windows\system32\cmd.exe
C:\wamp\bin\mysql\mysql15.1.36\bin>mysql -u root -p -e "GRANT USAGE ON *.* TO 'springappuser'@'";
Enter password: *****
C:\wamp\bin\mysql\mysql15.1.36\bin>set pass=pspringappuser
C:\wamp\bin\mysql\mysql15.1.36\bin>mysql -u springappuser -ppsringappuser -e "DROP DATABASE IF EXISTS springapp;CREATE DATABASE springapp;use springapp;"
C:\wamp\bin\mysql\mysql15.1.36\bin>mysql -u springappuser -ppsringappuser -e "<springappdb.sql
C:\wamp\bin\mysql\mysql15.1.36\bin>echo "PROCESO REALIZADO CORRECTAMENTE"
"PROCESO REALIZADO CORRECTAMENTE"
C:\wamp\bin\mysql\mysql15.1.36\bin>pause
Presione una tecla para continuar . . .
```

Figura I.14. Script de creación e importación de la Base de Datos

Para comprobar que la instalación ha sido correcta, podremos abrir en el navegador la dirección localhost/phpMyAdmin, entrar en la base de datos “springapp” y ver que aparecen en la parte izquierda de la página (Figura I.15), las distintas tablas creadas.

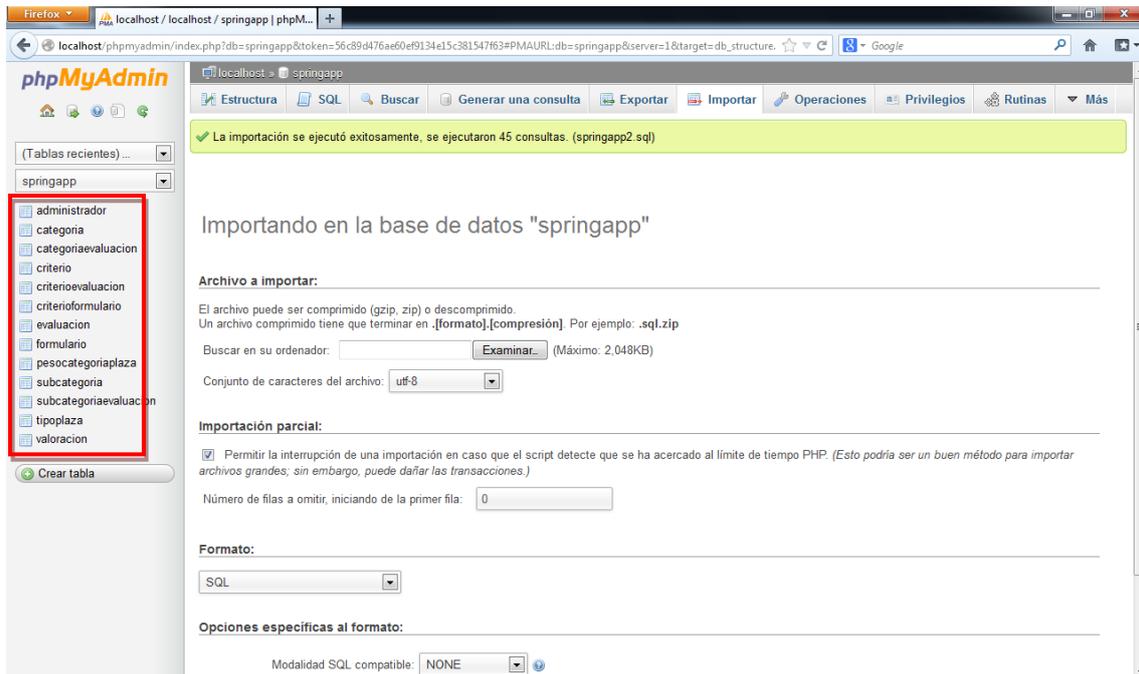


Figura I.16. Finalización de la creación de la Base de Datos.

Instalación Java

Antes de la instalación del Servidor TomCat, necesitamos la el JRE de Java ya que este servidor funciona bajo Java.

Para descargarlo, visitamos su sitio web, y en nuestro caso, descargaremos la versión para Windows (Figura I.17).

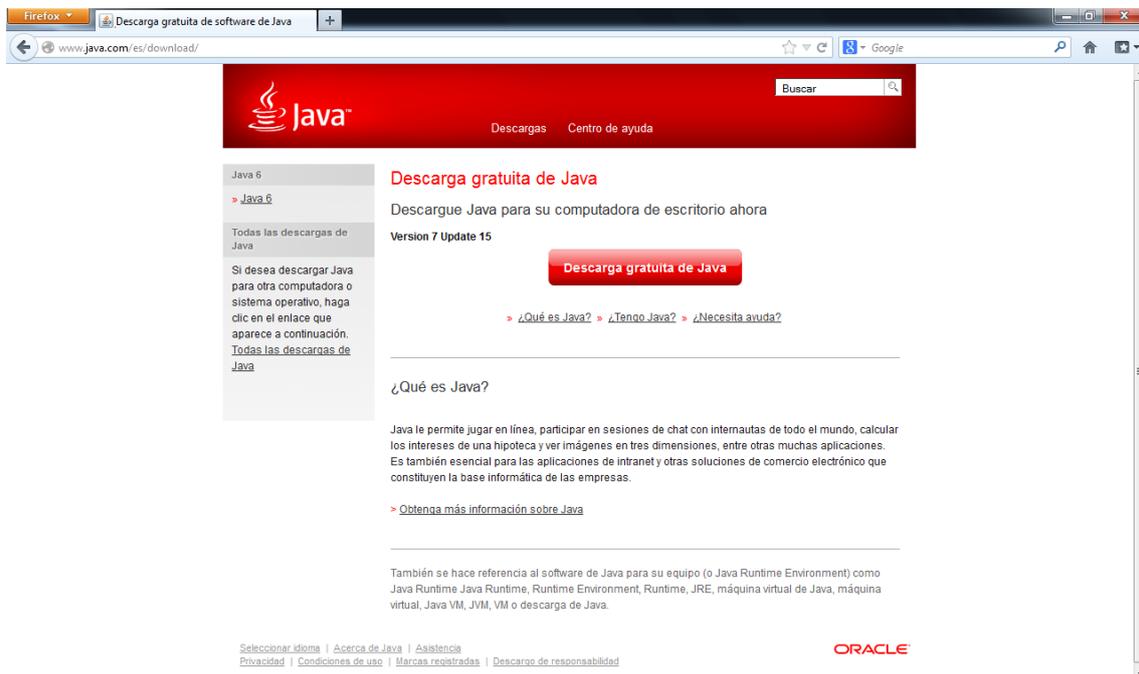


Figura I.17. Página de Descarga de JRE de Java.

Al inicio de la descarga gratuita deberemos de Aceptar las condiciones que nos marca la página web para continuar con la descarga. Para ello, pulsamos en la opción de Aceptar e Iniciar Descarga (ver Figura I.1816).



Figura I.18. Aceptar la licencia de JRE. Java

Elegiremos la opción de Guardar archivo (Figura I.1917), e indicaremos la ruta donde queremos almacenar el archivo de instalación.

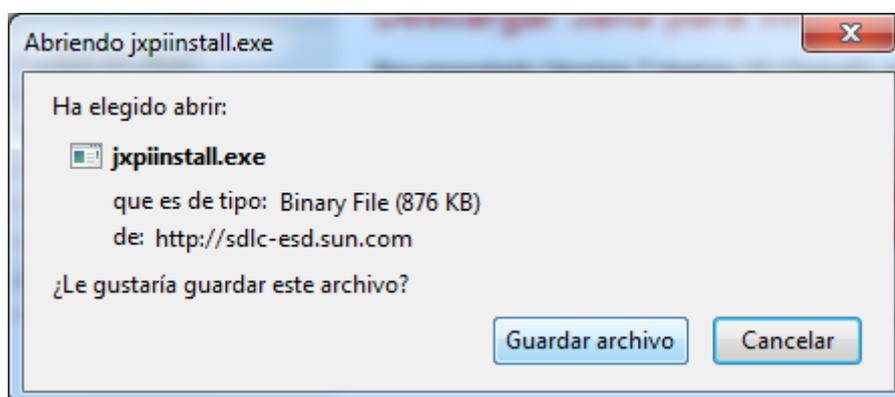


Figura I.19. Confirmación de guardar archivo JRE. Java

En nuestro caso, no es necesario visitar la página web ya que los archivos instalables se encuentran en el DVD. Finalizada la descarga, o una vez localizado el archivo en el DVD, pasamos a la instalación. Haciendo doble clic, se nos indica la pantalla de bienvenida (Figura I.20) para la instalación.



Figura I.20. Pantalla de Bienvenida Instalación Java.

Una vez pulsemos en instalar, comienza la descarga de algunos archivos necesarios. En esta ventana (Figura I.21) no es necesario realizar ninguna acción.



Figura I.21. Pantalla de descarga Instalación Java.

Pasado un breve tiempo de descarga, se nos indica que java ha sido instalado, y solamente deberemos hacer clic en Cerrar (Figura I.2220).

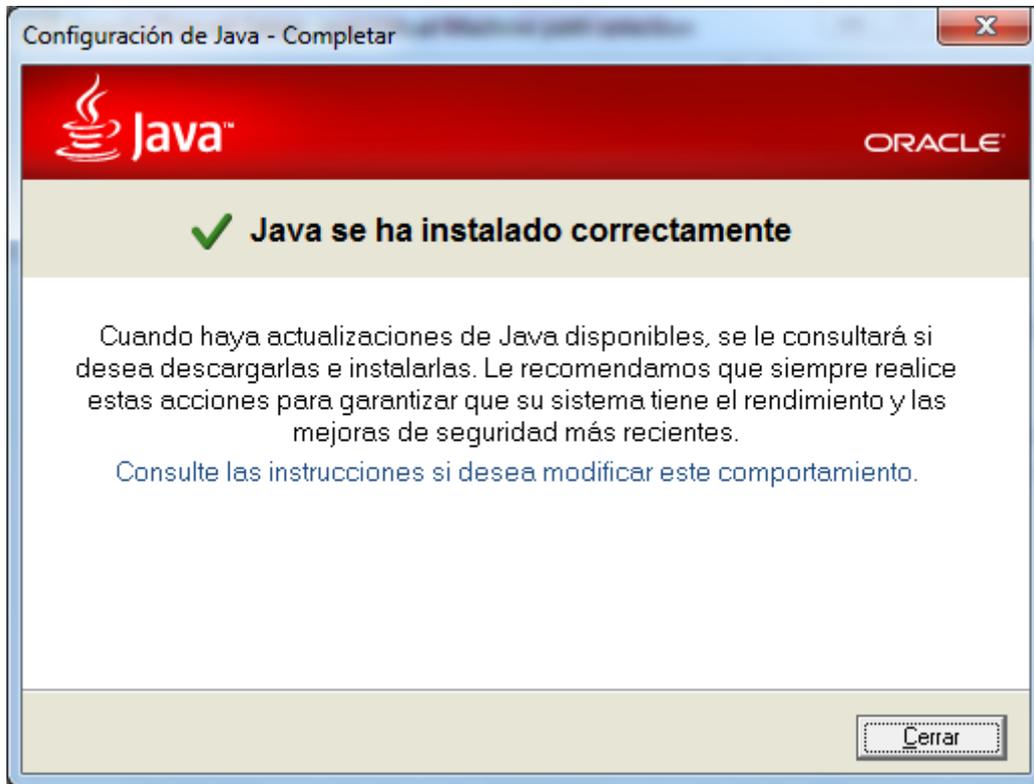


Figura I.22. Finalización Instalación JRE. Java.

Instalación Apache TOMCAT

Para la instalación de Apache TomCat, visitamos la página “<http://archive.apache.org/dist/tomcat/tomcat-7/v7.0.37/bin>” y descargamos el ejecutable “.exe” correspondiente a la versión 7.0.37, señalada en Figura I.23.

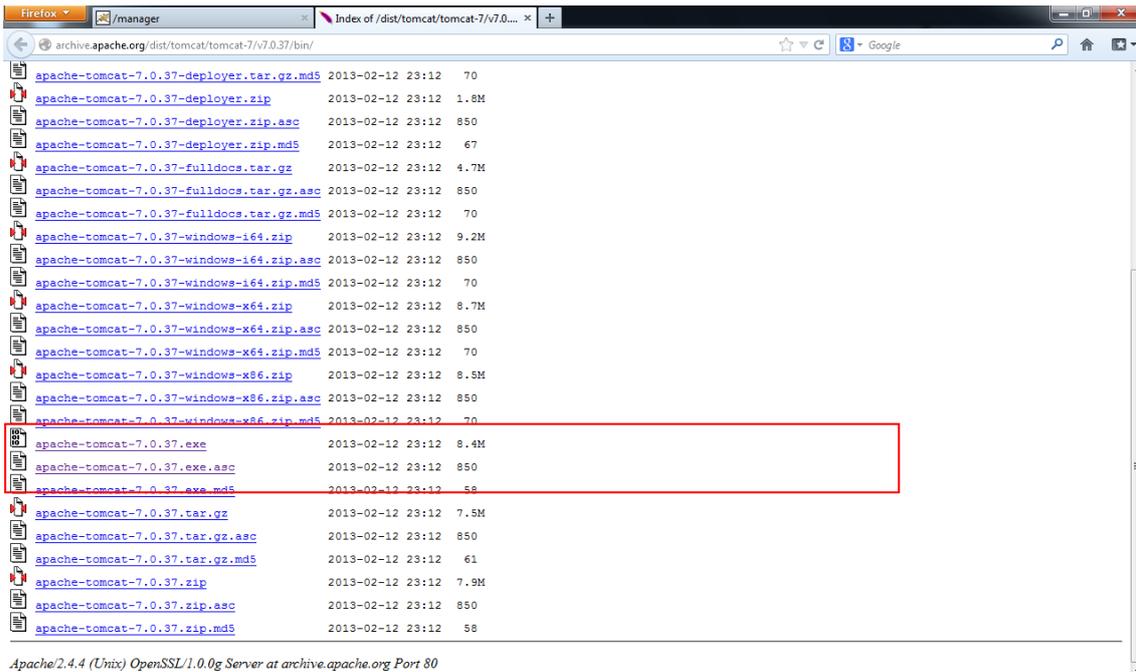


Figura I.23. Descarga de Apache TomCat desde su página web.

Una vez finalizada la descarga, o encontrado el archivo correspondiente en el DVD, procederemos a la instalación. En la página de bienvenida (Figura I.2422), pulsamos en Siguiente (Next >).



Figura I.24. Pagina de Bienvenida. Instalación TomCat

Tras leer la licencia, antes de darle a siguiente, continuamos con la instalación (Figura I.2523).

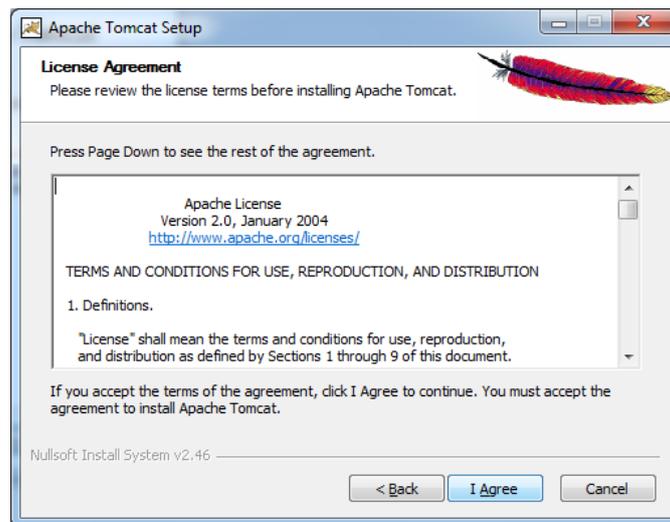


Figura I.25. Licencia Instalación Apache TomCat

Si quisiéramos personalizar algún aspecto de la instalación, este sería el momento. En nuestro caso, nos conformamos con la instalación por defecto, por lo que no habría que hacer nada especial. Continuamos haciendo clic en Siguiete (Next >). Ver Figura I.2624:

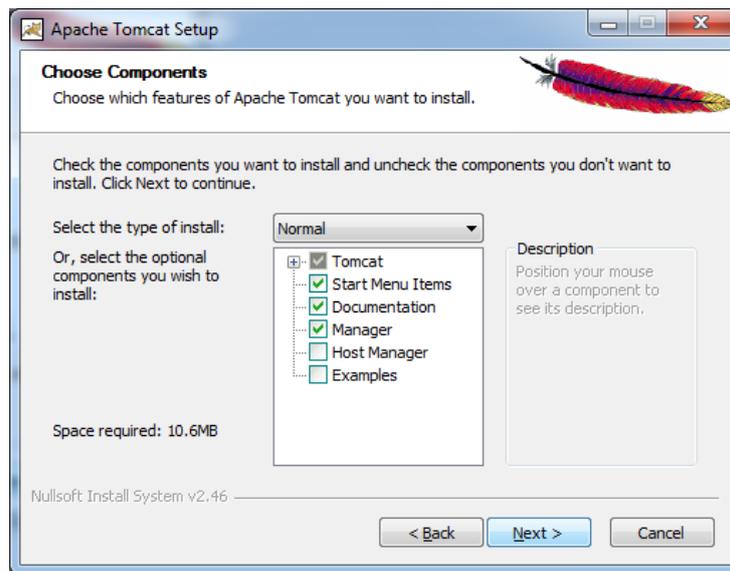


Figura I.26. Personalizar elementos a Instalar. Apache TomCat.

Aquí personalizaríamos los puertos, así como el usuario que va a tener acceso al sistema. En este caso, si es necesario poner un usuario y una contraseña, para posteriormente poder añadir nuestro sistema al servidor (Figura I.27).

- o **Usuario:** spring **Password:** spring

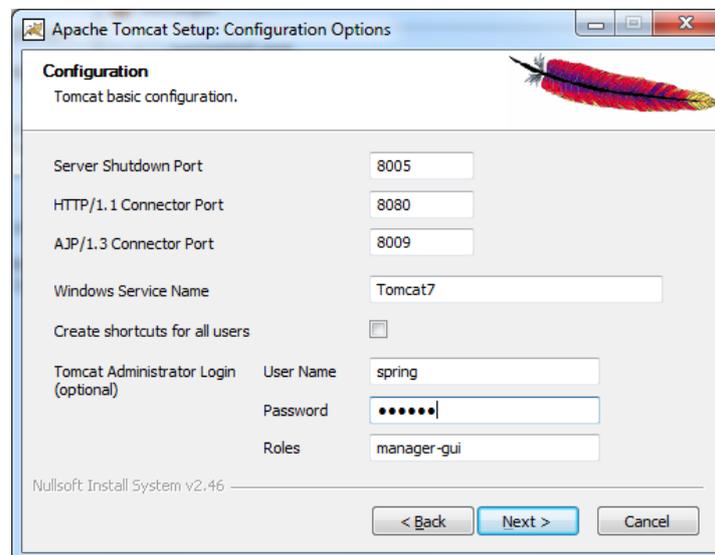


Figura I.27. Configuración puertos y usuario. Apache TomCat

A continuación, aparece por defecto la ruta de la instalación de JRE que hemos hecho anteriormente. Si no fuera así, solamente habría que buscar la ruta y pulsar en Siguiente (Next >). Ver Figura I.2826:

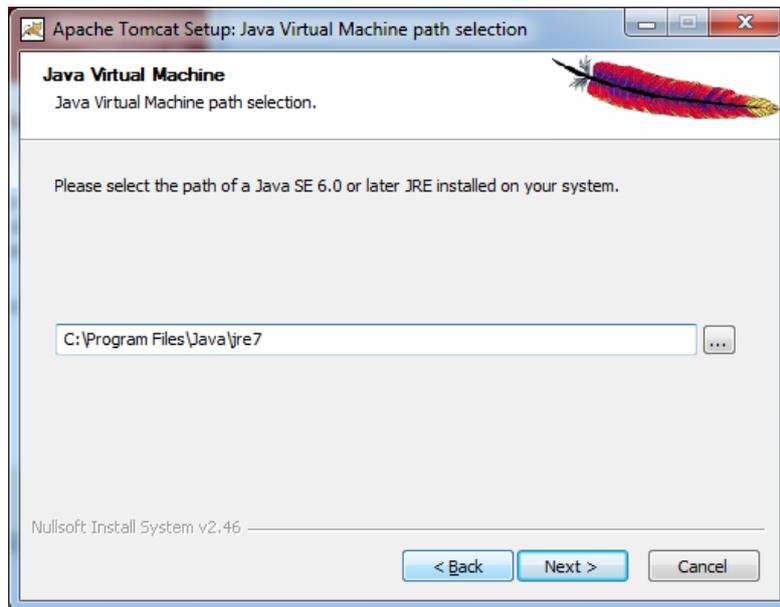


Figura I.28.Ruta de JRE java. Instalación Apache TomCat

Ahora deberíamos de indicar una ruta si quisiéramos cambiar la que viene por defecto. En nuestro caso (ver Figura I.29. Fichero de Destino. Instalación Apache TomCat) no es necesario, por lo que hacemos clic en Instalar (Install).

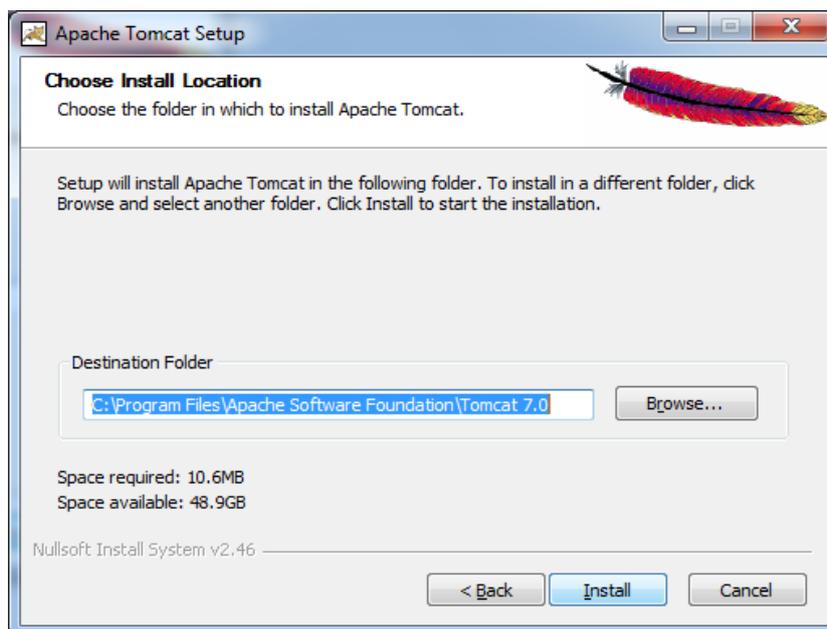


Figura I.29. Fichero de Destino. Instalación Apache TomCat

Y comienza la instalación (Figura I.3028).

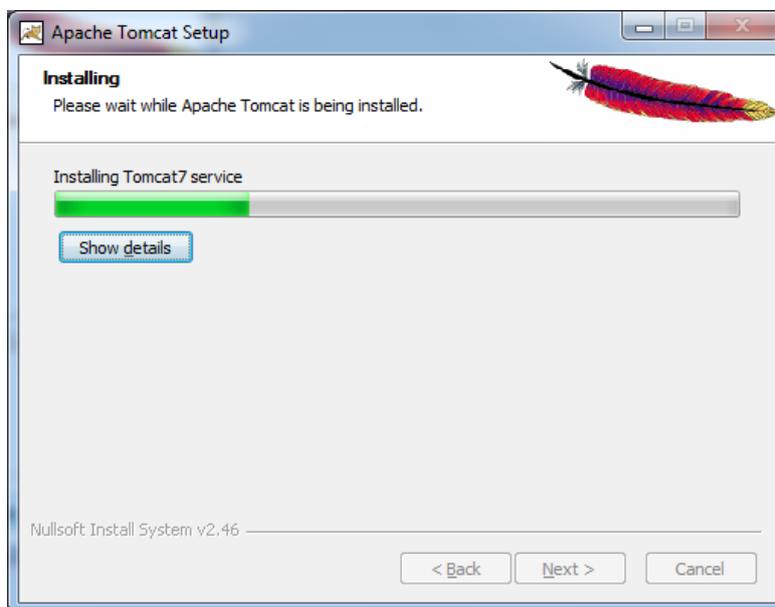


Figura I.30. Instalación del Servicio Apache TomCat.

Ahora procederemos a arrancar el servidor. Para ello, con la opción de Run Apache TomCat, hacemos clic en Finish (Figura I.31).

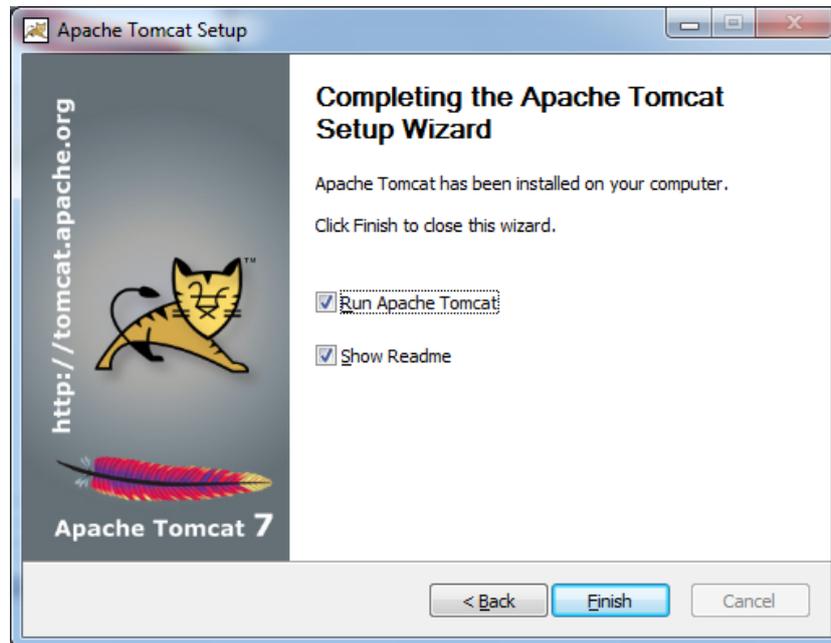


Figura I.31. Arrancar Apache TomCat Server.

Arrancando el servidor, nos encontramos con este cuadro de diálogo (ver Figura I.320), en el que no es necesario realizar ninguna acción.

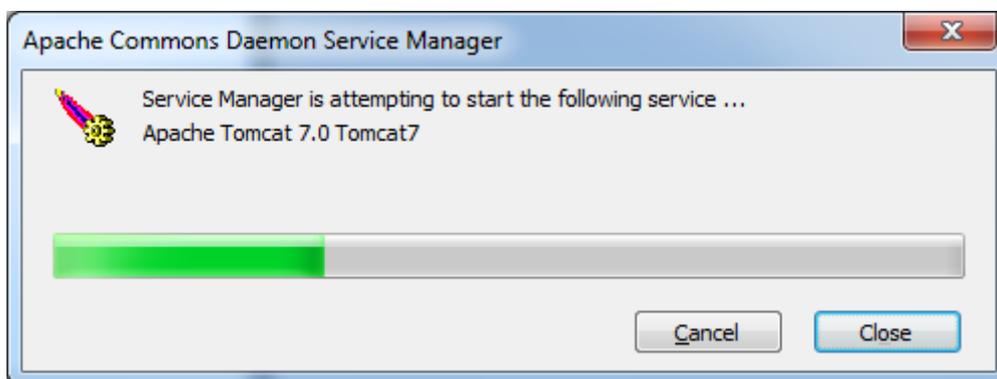


Figura I.32. Servicio Tomcat Arrancando.

Y una vez terminado de arrancar el servicio, encontramos una especie de pluma en la barra de Estado (ver Figura I.331), que nos indica que el servidor está funcionando.

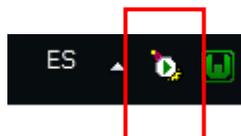


Figura I.33. Servicio Tomcat Arrancado

Desplegar la aplicación en el servidor.

Con el servicio arrancado, volvemos a realizar la configuración desde el navegador, esta vez, entrando en el puerto 8080. Para ello, indicamos en la barra de direcciones “http://localhost:8080”. Y Obtenemos la Figura I.34. Hacemos clic en Manager App.

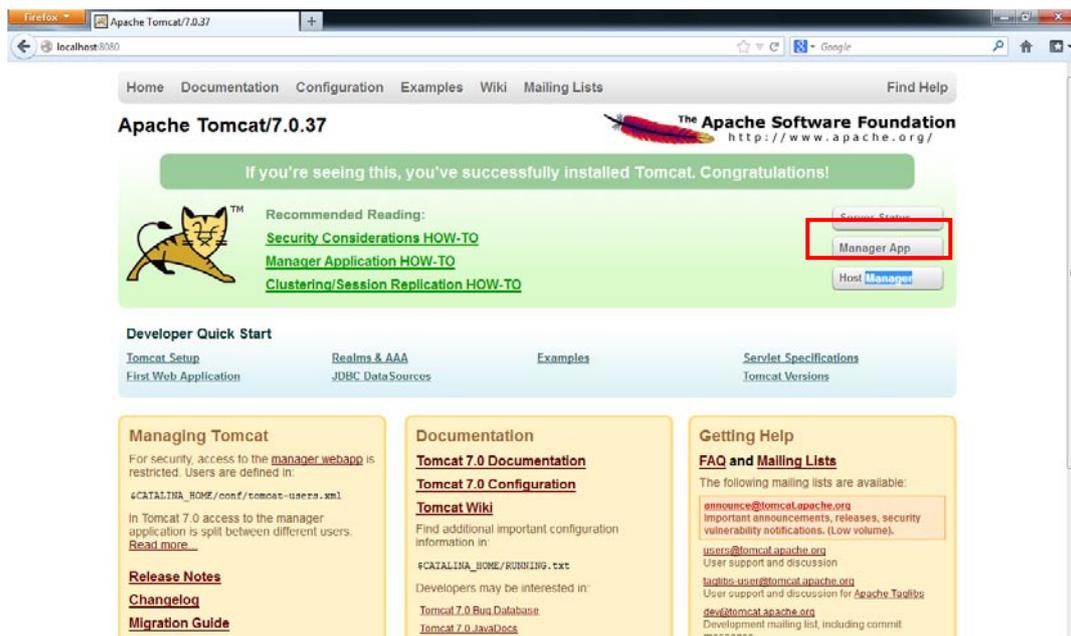


Figura I.34. Pagina de configuración Apache TomCat.

Se nos pide en el cuadro de diálogo (ver Figura I.35) que introduzcamos el usuario y la contraseña, en nuestro caso eran “spring” para ambos.

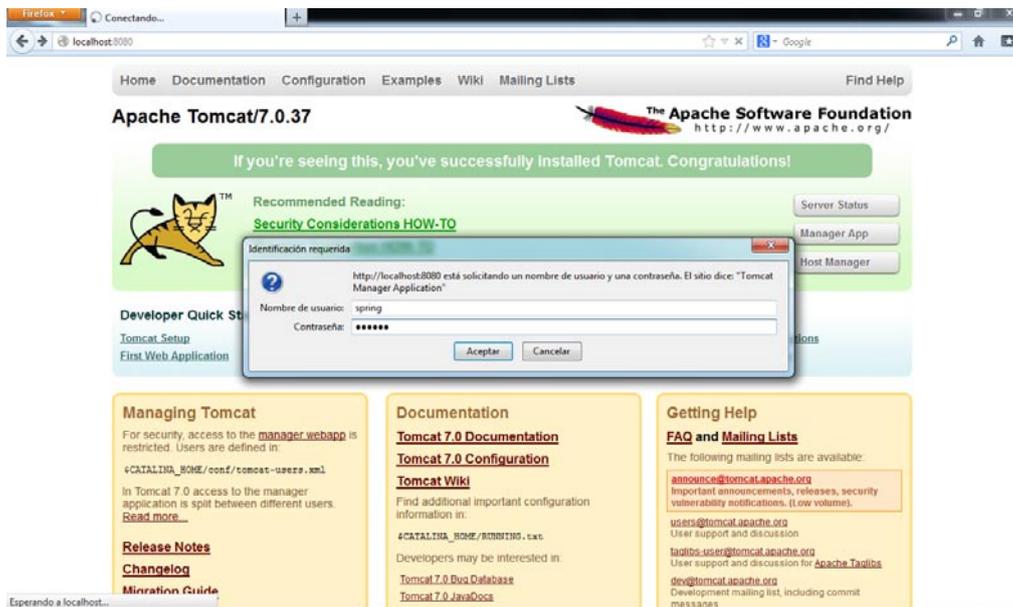


Figura I.35. Usuario y contraseña de TomCat.

Ahora pasamos a desplegar la aplicación (Figura I.36). Los pasos son bastante sencillos.

Tendremos que localizar el archivo “springapp.war” en el DVD, y pulsar en Desplegar.

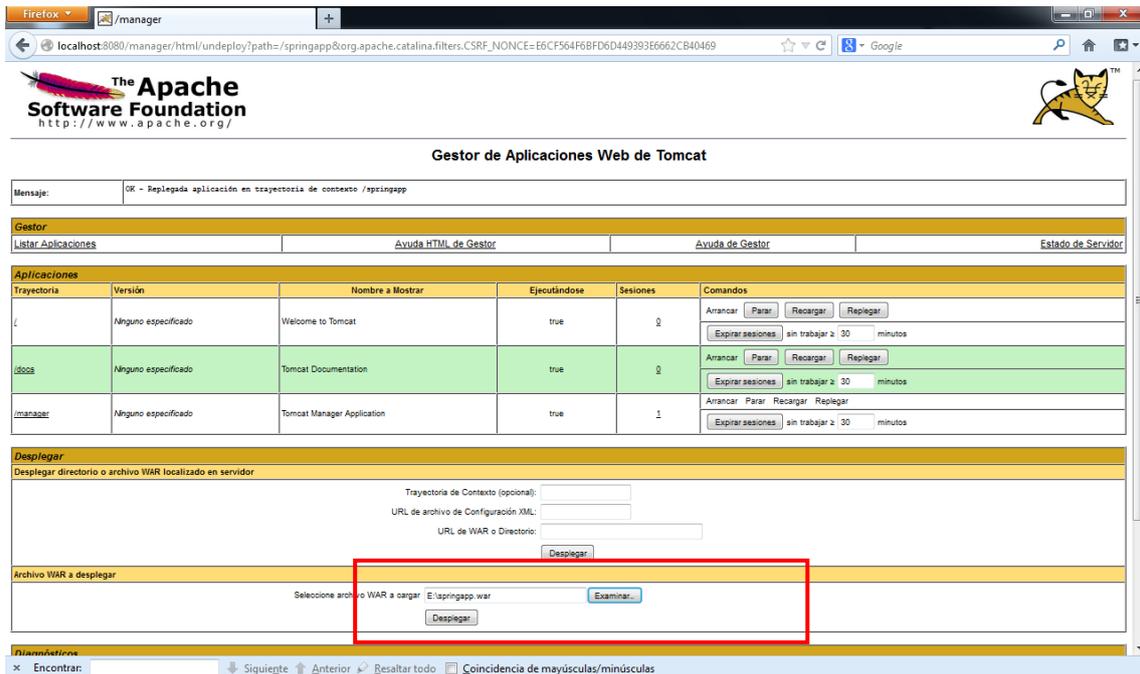


Figura I.36.Desplegando sistema. Apache TomCat.

Transcurrido un tiempo, tenemos nuestro sistema funcionando en el servidor. Podremos configurar como se ve en Figura I.37, el tiempo que tardan las sesiones en expirar, parar, recargar, etc. En nuestro caso, no es necesario hacer nada más.

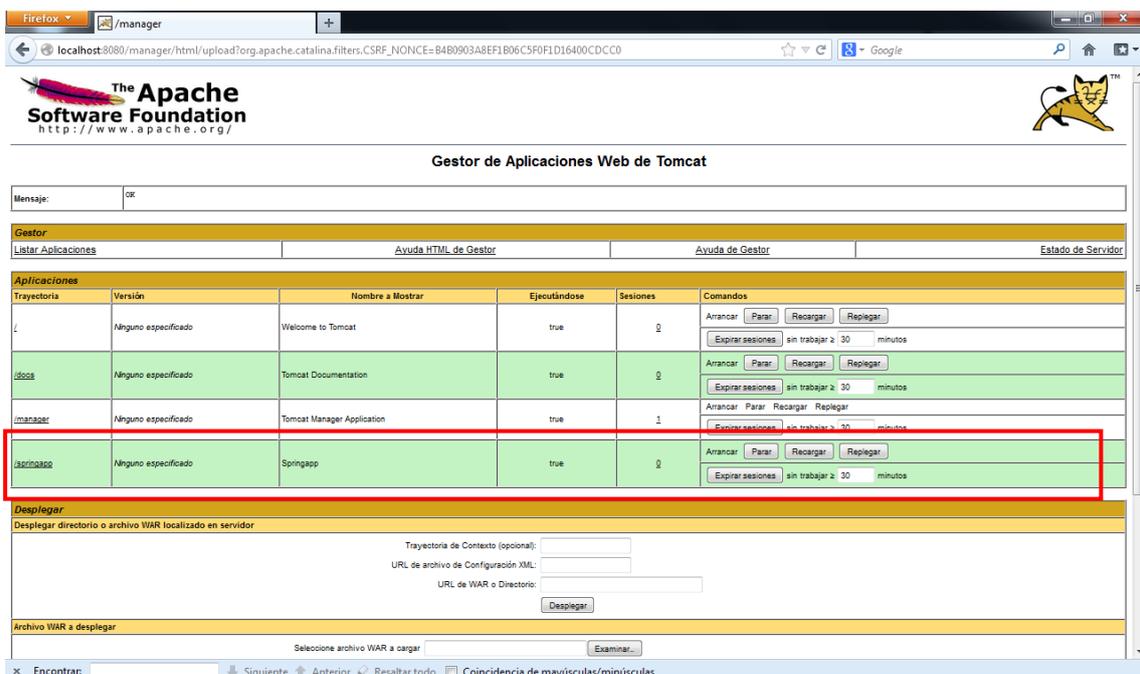


Figura I.37.Sistema springapp desplegado en el Servidor TomCat

Para confirmar que el sistema está funcionando, entraremos en la dirección localhost:8080/springapp y deberemos de visualizar la página de inicio del sistema, como se ve en Figura I.3836.



Figura I.38. Página de inicio del Sistema.

Anexo II.

Manual de usuario

Este manual de usuario está organizado como una visita guiada por la aplicación, cuando accedemos a la misma como Administrador.

El primer paso para utilizar la aplicación es abrir el navegador. En nuestro caso, el manual está realizado utilizando Firefox en su versión 19.0. Recomendamos el uso de este navegador para la visualización del sistema. No hay requisitos especiales de hardware y software, a excepción del uso de un navegador

Al abrir el navegador, debemos de teclear en la barra de direcciones `http://localhost:8080/springapp` y obtendremos la página de inicio, como se observa en la Figura II.1.



Figura II.1. Página de inicio del sistema.

Para entrar como Administrador debemos introducir el nombre de usuario y contraseña del administrador. Si alguno de estos datos (o los dos) no son correctos, el sistema nos muestra un mensaje de error como se observa en la Figura II.2.



Figura II.2. Pantalla de inicio. Errores en campos.

En el caso de una autenticación correcta nos aparece el menú principal del administrador, como en la Figura II.3.



Figura II.3. Página principal de administrador.

En la parte superior, encontramos el menú (ver Figura II.4), compuesto de Evaluar Aspirantes, Resultados, Gestión de los elementos de la Evaluación, y Gestión Formulario.



Figura II.4. Menú superior

Pasaremos a describir cada una de las opciones:

1. Evaluar Aspirantes

1.1. Añadir Nuevos Aspirantes

Teniendo la Evaluación, el formulario para la Evaluación, añadidos los criterios que van a ser tenidos en cuenta para evaluar a los candidatos, la importancia (peso) que va a tener cada una de las categorías para las plazas, podemos evaluar a los candidatos. Para ello, pasamos a la opción de “*Evaluar Aspirantes*”.

Si no se han añadido aspirantes al sistema, se indica que no hay ficheros válidos en el directorio, como muestra la Figura II.5.



Figura II.5. Evaluar aspirantes. Inicio.

Procederemos a añadir los aspirantes al sistema. Para ello, se utiliza la opción de “*Añadir nuevo Aspirante*”.

Seleccionaremos el fichero a través del botón “*Browse*”, y navegaremos hasta el directorio en el que se encuentre el fichero. Una vez localizado, hacemos clic en “*subir Fichero*” (Figura II.6).

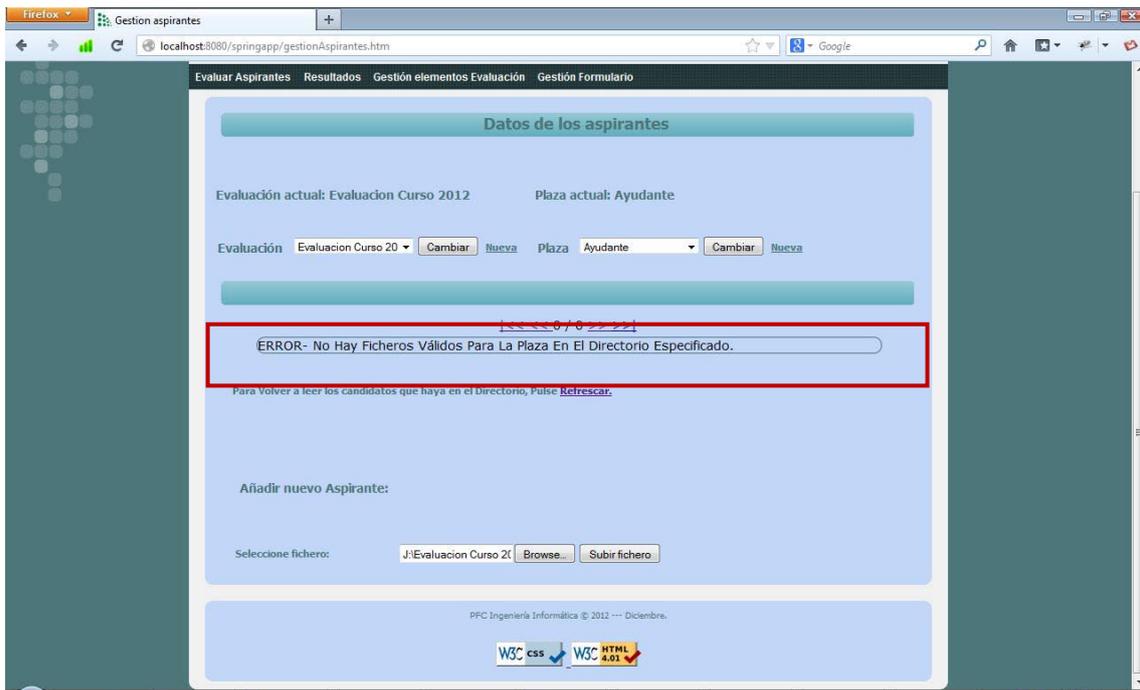


Figura II.6. Evaluar aspirantes. Subir fichero aspirante.

Cuando el fichero esté almacenado en el directorio que hemos indicado en la configuración del sistema, se nos indica que ha sido grabado correctamente (Figura II.7).

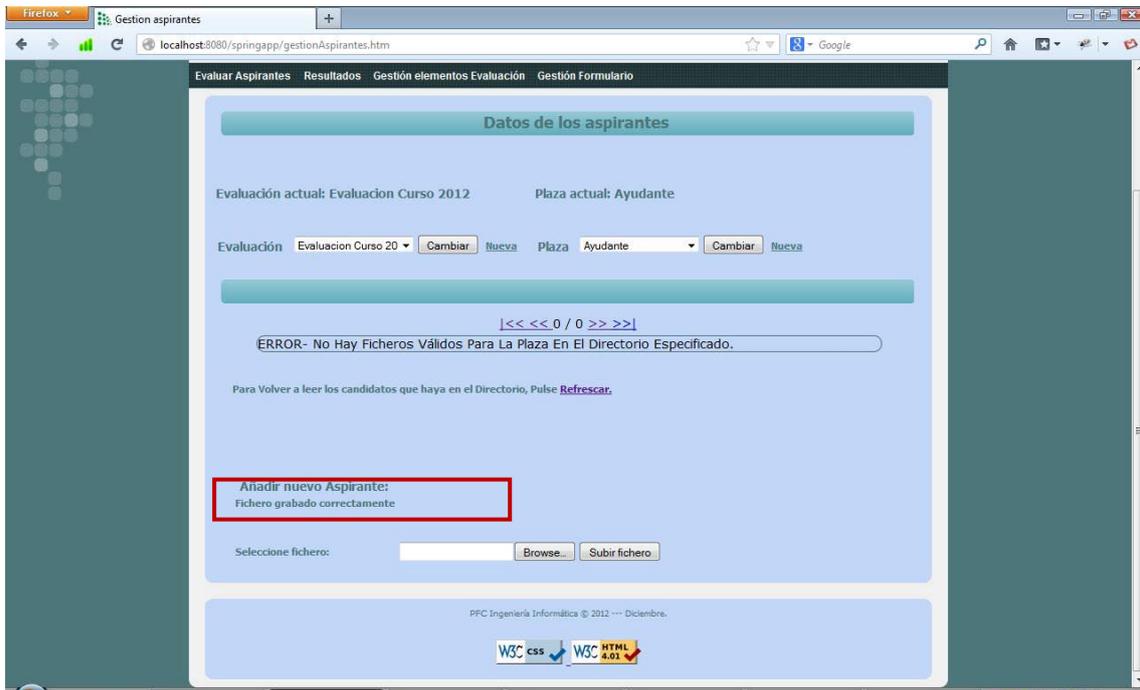


Figura II.7. Evaluar aspirantes. Fichero subido correctamente.

Si se encuentra el error de que no hay ficheros válidos en la plaza, es posible que sea porque la evaluación todavía no se ha refrescado (ver 1.2-Refrescar), o que para la plaza Actual, no hay aspirantes. En ese caso, el error será el mismo que se muestra en la Figura II.8.



Figura II.8. Evaluar aspirantes. Error no hay ficheros en plaza.

Si al entrar en una evaluación, aparece el siguiente error (Figura II.9), es debido a que para la evaluación seleccionada, en el directorio que se indica, no haya ningún aspirante válido, o que todavía no se haya refrescado los ficheros correspondientes.



Figura II.9. Evaluar aspirantes. Error no hay ficheros en directorio.

1.2. Refrescar Datos de Aspirantes

Tras subir los ficheros necesarios al sistema, haremos clic en “*Refrescar*”, y trascurridos unos instantes, obtendremos los candidatos, en caso de que haya datos válidos (Figura II.10). Es importante Refrescar, cuando se hayan producido cambios en el Formulario de Evaluación (modificación en Criterios añadidos o nuevas plazas creadas).

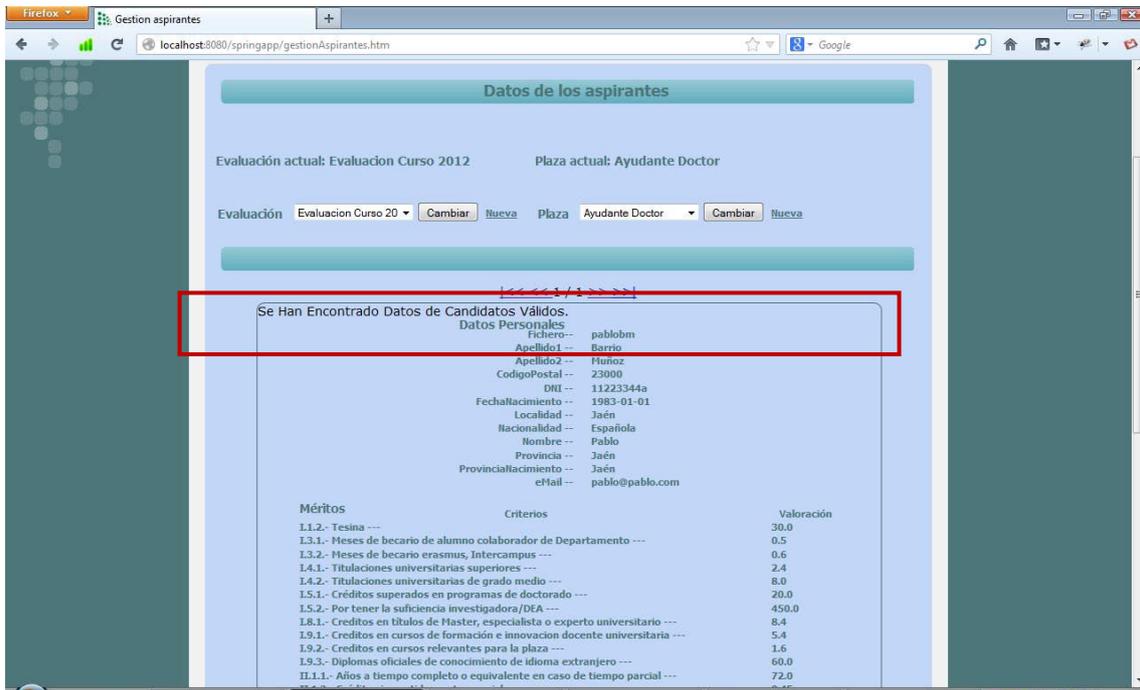


Figura II.10. Evaluar aspirantes. Candidatos válidos.

Si se han producido errores, se notifican, por si es necesario alguna rectificación en los ficheros. En la parte inferior, activando o desactivando la casilla de verificación, aparecen o desaparecen los errores (ver Figura II.11):

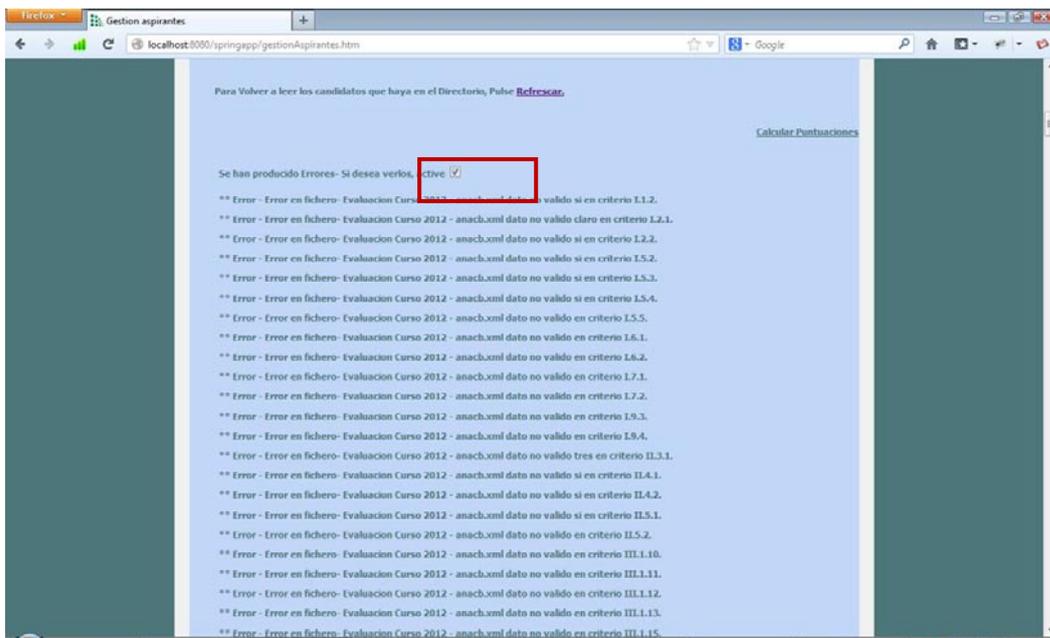


Figura II.11. Evaluar aspirantes. Errores al procesar los ficheros de aspirantes.

1.3. Calcular Puntuaciones

Una vez tenemos los aspirantes cargados en el sistema, procederemos a realizar los cálculos. Para ello, hacemos clic en la opción de “*Calcular Puntuaciones*” (Figura II.12)

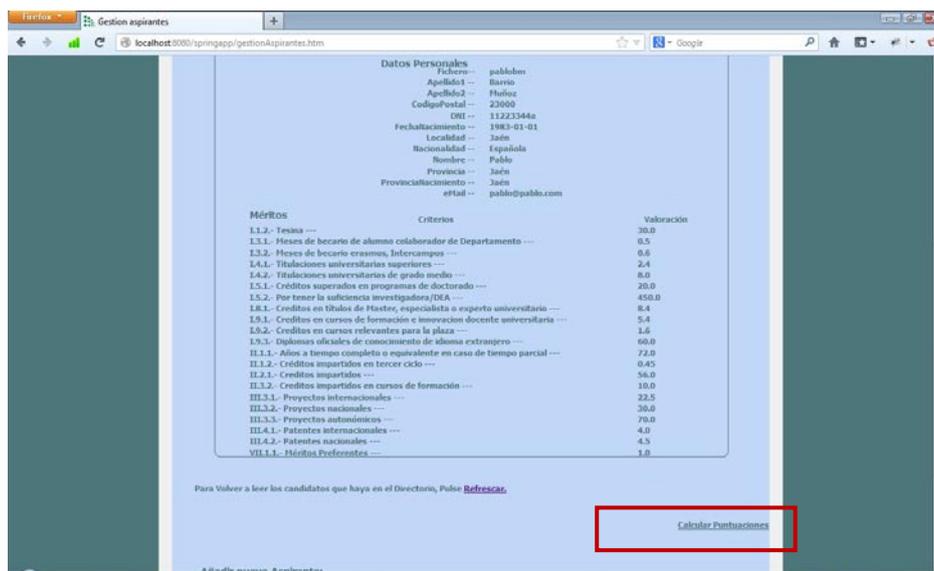


Figura II.12. Evaluar aspirantes. Calcular puntuaciones.

Trascurridos unos instantes, el sistema nos pasa al apartado correspondiente a los “*Resultados*”. Y obtenemos la siguiente Figura II.13. En ella, tenemos en la parte derecha, el ranking de los usuarios dependiendo de la puntuación que hayan obtenido.

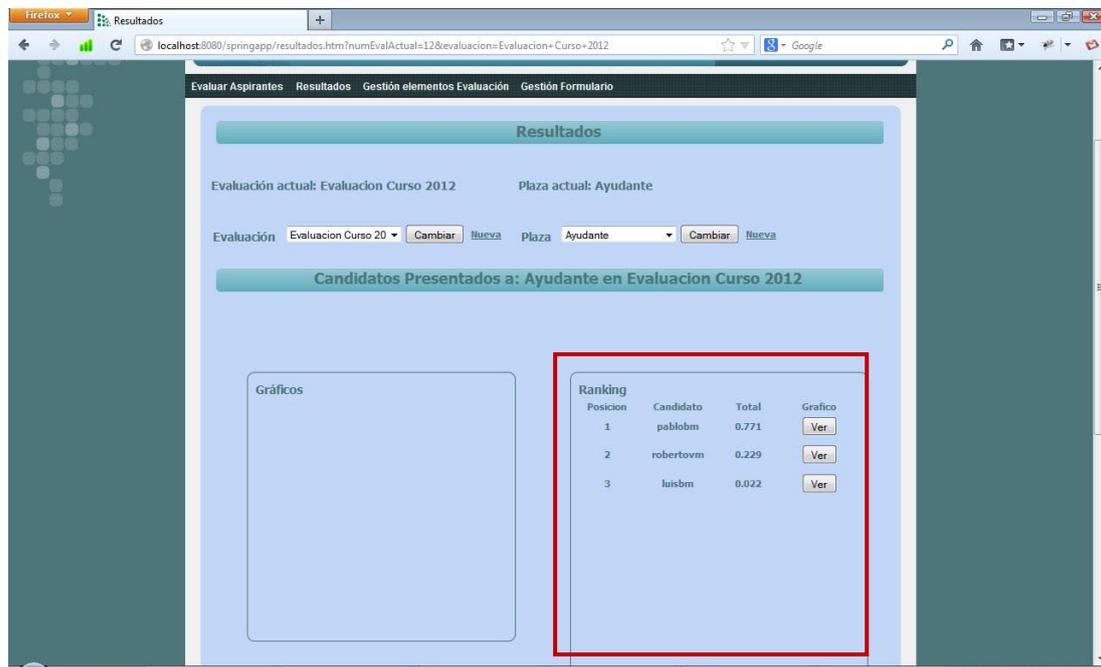
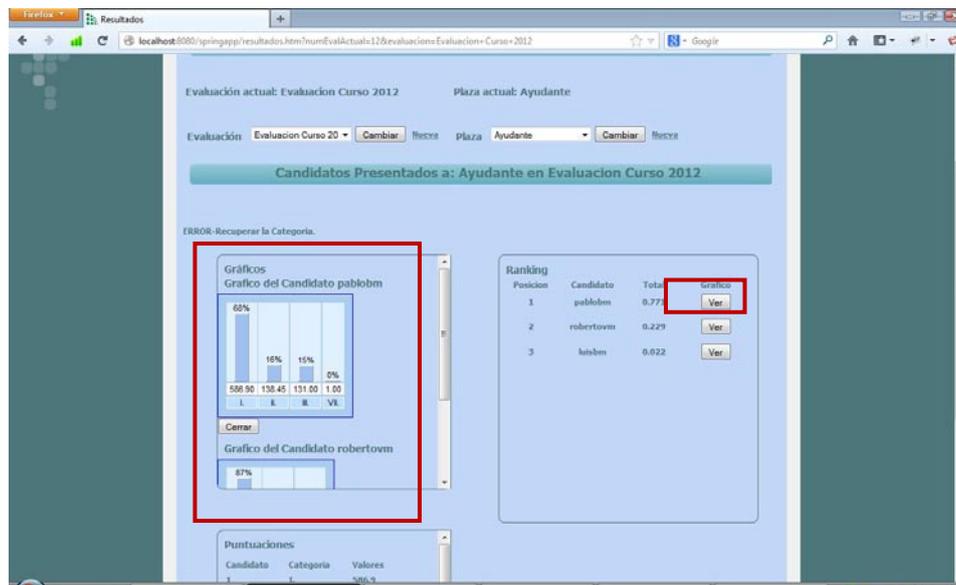


Figura II.13. Página de Resultados. Ranking

2. Resultados

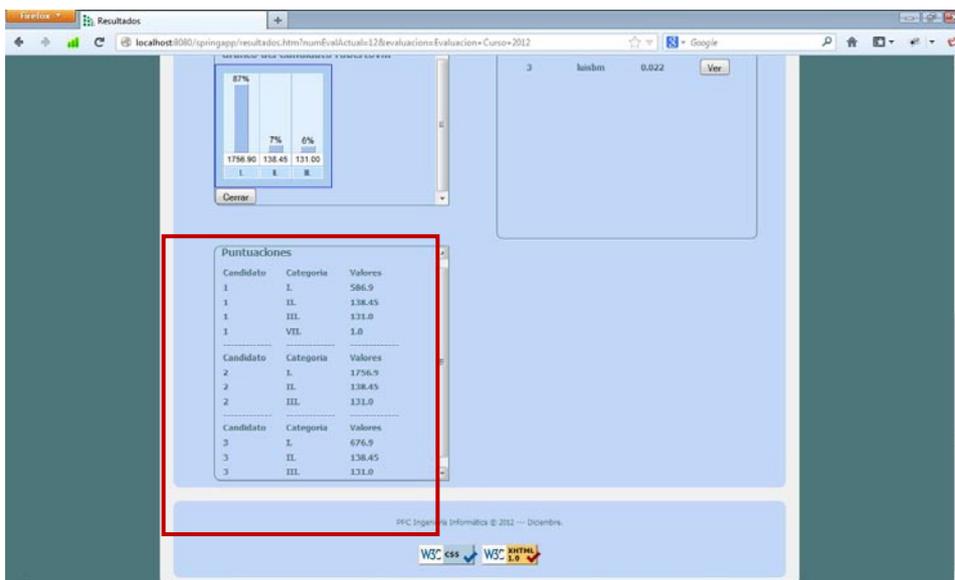
2.1. Ver Gráficos

Existe la posibilidad de ver los gráficos correspondientes a cada uno de los usuarios. Haciendo clic en la opción “Ver”, a la derecha, aparecen los gráficos, con cada una de las categorías evaluadas, y la puntuación obtenida en cada una de ellas. En la siguiente Figura II.14 , se pueden observar dos gráficos de dos aspirantes.



Si se quisiera ocultar algún gráfico, solamente hay que hacer clic en “Cerrar”.

En la parte inferior, además se puede consultar la puntuación obtenida por todos, de manera ordenada. En la Figura II.15 aparecen las puntuaciones de los candidatos.



3. Gestión de los Elementos de Evaluación

3.1. Ver Categorías, Subcategorías y Criterios

En la figura siguiente Figura II.16, podemos ver las categorías disponibles. Haciendo clic en la lista desplegable, tenemos todas las categorías con las que cuenta el sistema.

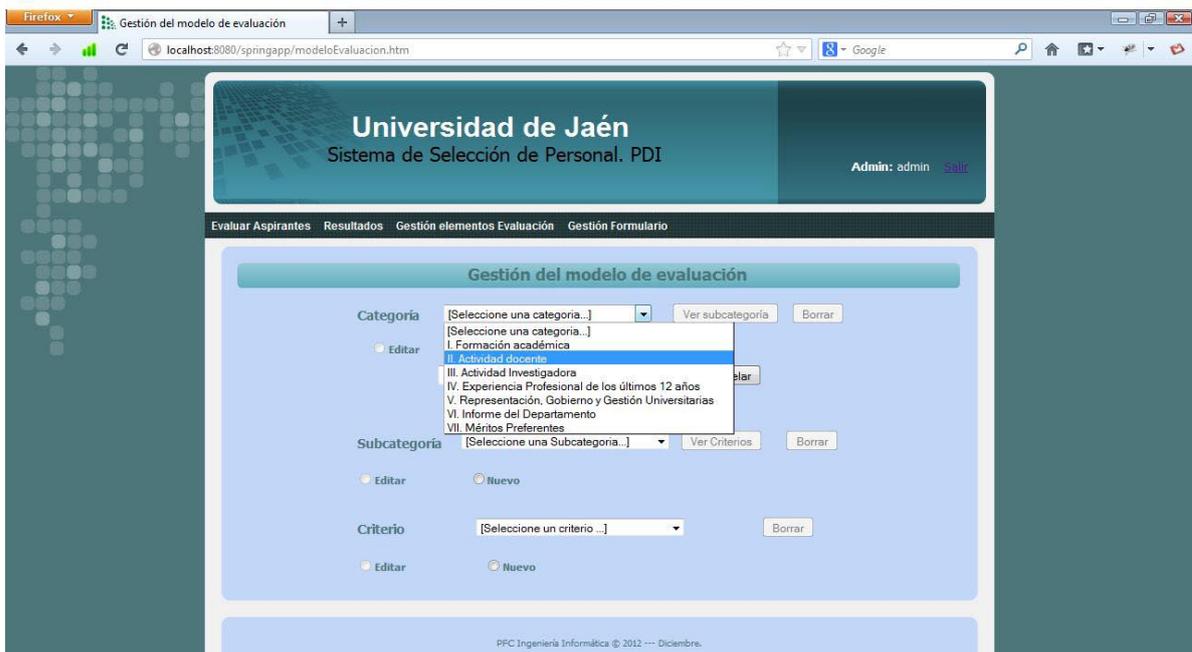


Figura II.16. Gestión del Modelo de Evaluación. Ver categorías.

Para ver las Subcategorías, es necesario que haya una categoría seleccionada. Para ello, seleccionamos una categoría, y hacemos clic en Ver Subcategorías. En la siguiente Figura II.17, podemos ver el paso que hay que seguir:

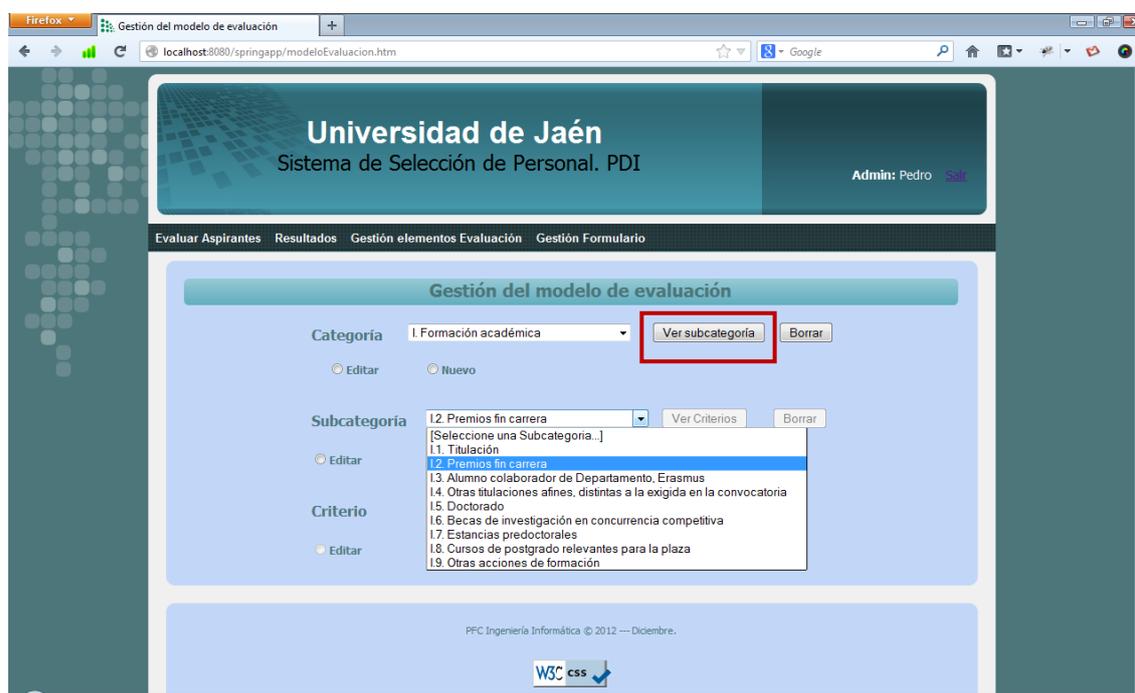


Figura II.17. Gestión del Modelo de Evaluación. Ver Subcategorías.

Para ver los Criterios con los que cuenta el sistema. Es necesario haber seguido dos pasos anteriores:

1. Haber seleccionado la categoría a la que pertenecen los criterios que queremos ver, y pulsado en Ver Subcategoría. Como en la figura anterior (ver Figura II.17).
2. Haber seleccionado la subcategoría a la que pertenecen los criterios que estamos buscando, y pulsado en Ver Criterio, como está reflejado en la Figura II.18.
3. Una vez realizados los pasos anteriores, podemos ver en el desplegable, los criterios que corresponden a esa Subcategoría. En la Figura II.18 podemos ver el resultado.

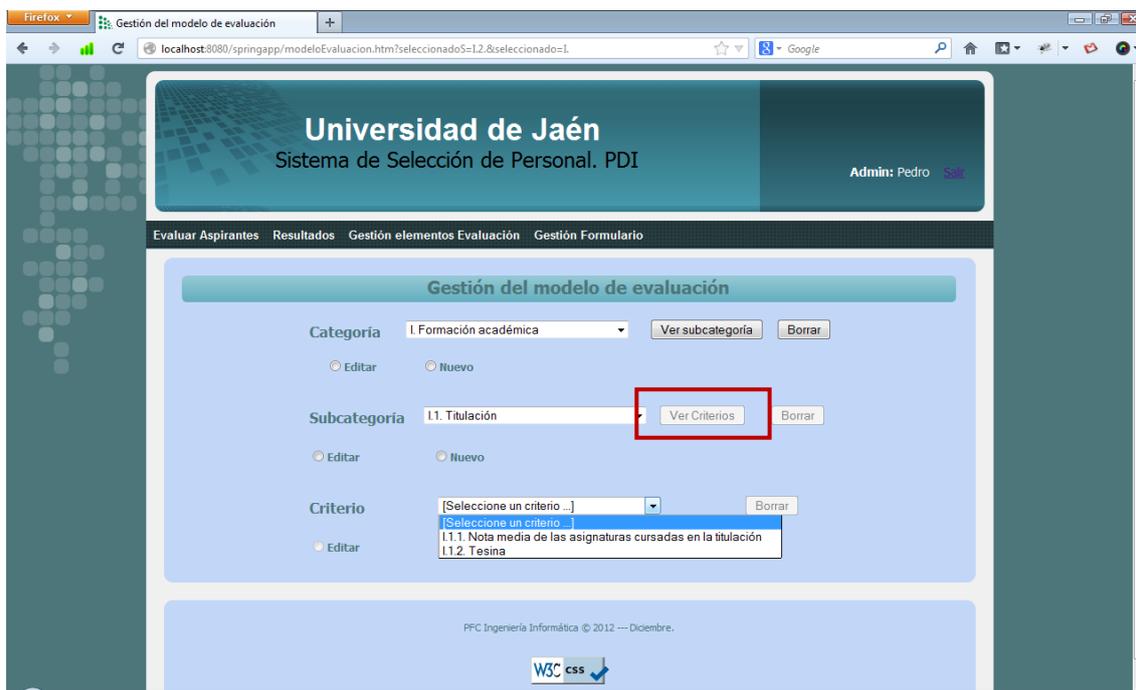


Figura II.18. Gestión del Modelo de Evaluación. Ver Criterios.

3.2. Modificar Categorías, Subcategorías y Criterios

Para modificar las categorías, tendremos que seleccionar del desplegable de las categorías, la que queramos cambiar su denominación, es decir, cambiar su nombre, ya que no es posible cambiar el identificador que acompaña a la categoría. Una vez seleccionada, hacemos clic en “*Editar*”, y el sistema nos muestra dos espacios de los cuales, solamente podremos escribir en el de la derecha. Para finalizar el proceso, se hace clic en “*Guardar*”,

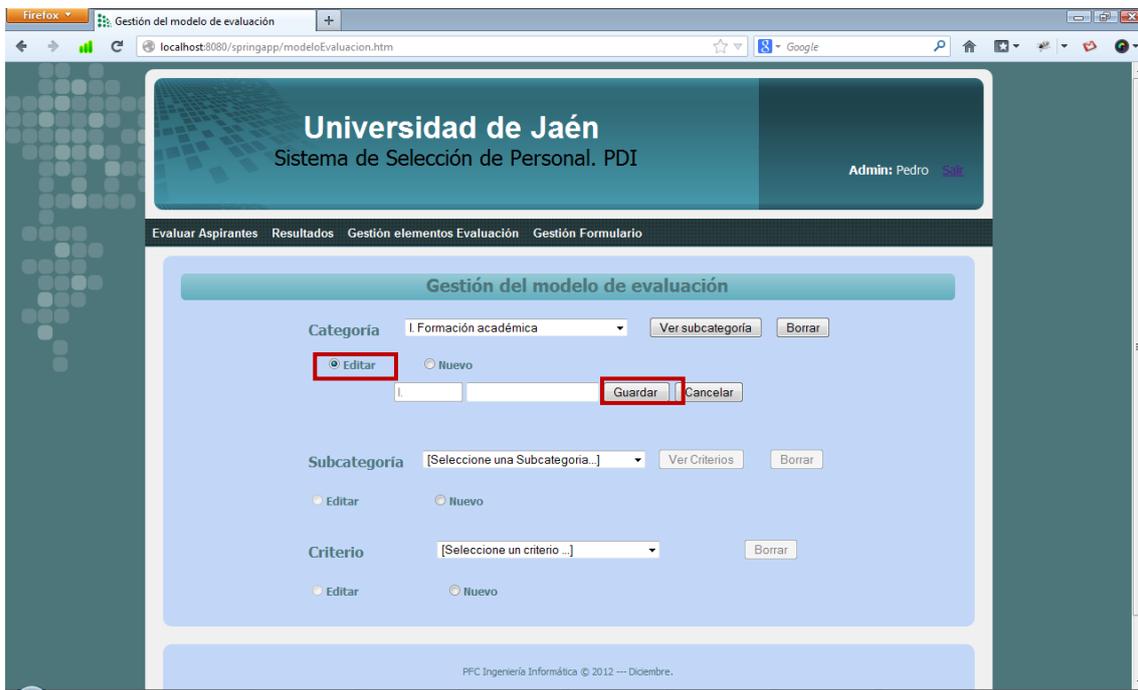


Figura II.19. Gestión del Modelo de Evaluación. Modificar Categorías.

Una vez hecho clic en “Guardar” el sistema nos indica que la categoría ha sido modificada, como se puede ver en la Figura II.20:

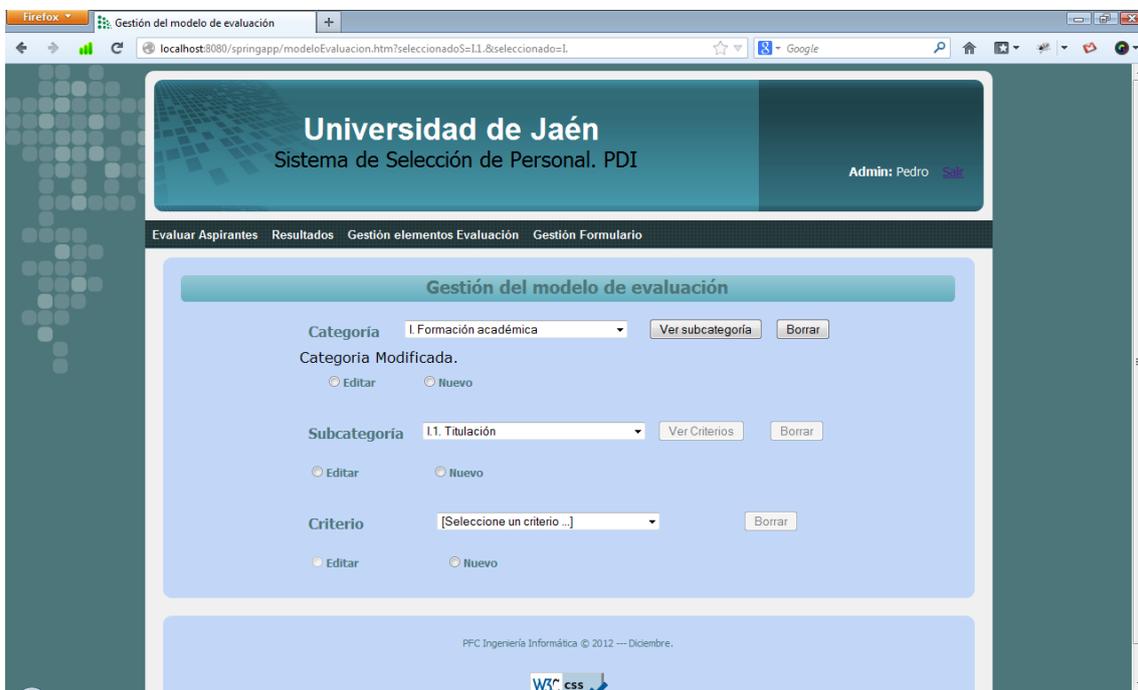


Figura II.20. Gestión del Modelo de Evaluación. Categoría Modificada.

Modificar Subcategorías

Para poder modificar una Subcategoría, debemos de saber la categoría a la que pertenece (ver apartado “*Ver Subcategorías*”). Habiendo seleccionado la subcategoría que queremos modificar, procedemos como para las categorías (Figura II.21):

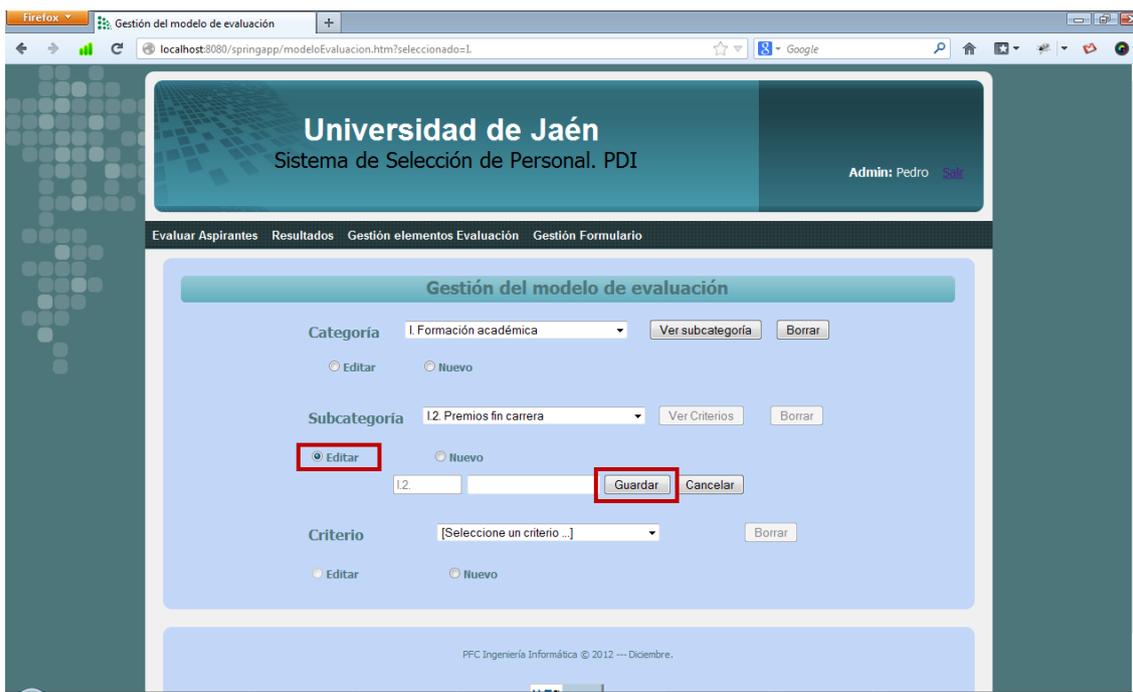


Figura II.21. Gestión del Modelo de Evaluación. Modificar Subcategorías.

Modificar Criterios

Para la modificación del nombre del criterio, hay que seguir los mismos pasos que para los anteriores. Lo primero es situarnos en el Criterio que queremos modificar para ello, ver el punto 3.1 -Ver Criterios. Una vez situados en el criterio, procedemos como en los apartados anteriores de Modificar Subcategorías y Categorías. Una vez pulsado “*Guardar*” (ver Figura II.22), obtendremos el mensaje de “*Categoría Modificada*”

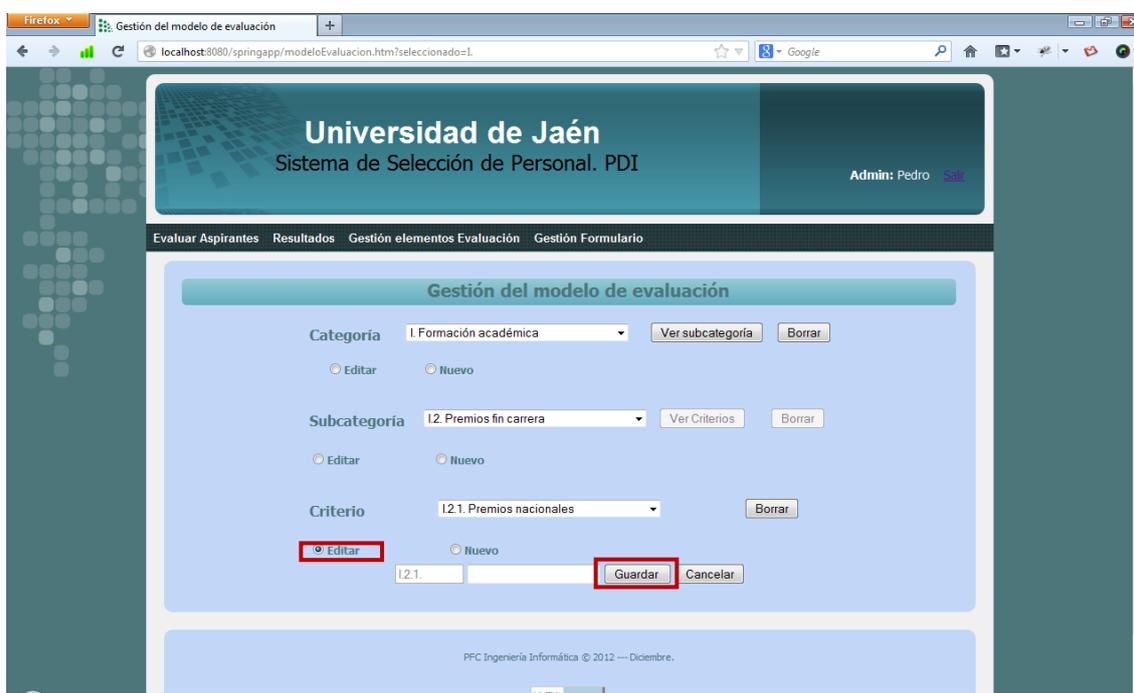


Figura II.22. Gestión del Modelo de Evaluación. Modificar Criterios.

3.3. Eliminar Categorías, Subcategorías y Criterios

En caso de que se necesite borrar una categoría, se seleccionará la categoría del desplegable, y se hará clic en “Borrar”, obteniéndose el siguiente cuadro de diálogo. En caso de que sí se quiera seguir con el borrado, se hará clic en “OK” (ver Figura II.23).

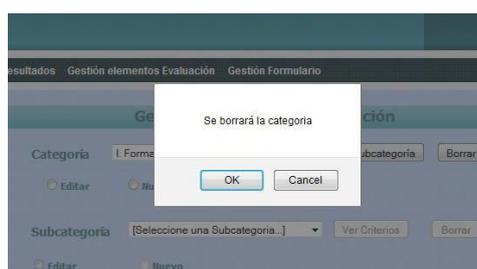


Figura II.23. Gestión del modelo de evaluación. Borrar categoría.

Si la Categoría está siendo usada en alguna evaluación, para evitar perder datos de los aspirantes, no se permite el borrado de dichas Categorías. Por ello, se indica el error “ERROR. Categoría en Uso”, como se muestra en la Figura II.24:



Figura II.24. Gestión del modelo de evaluación. Error categoría en uso.

3.4. Añadir Categorías, Subcategorías y Criterios

El procedimiento para añadir una categoría consiste en hacer clic en Nuevo, y rellenar la categoría, con el formato “Letra” seguida de un punto “.”. y el nombre correspondiente a la categoría (ver Figura II.25).



Figura II.25. Gestión del modelo de evaluación. Añadir categoría.

En este punto se pueden dar varios errores:

Dejamos nombre de la categoría en blanco (ver Figura II.26).



Figura II.26. Gestión del modelo de evaluación. Error campos en blanco.

Formato de la categoría inválido

Si escribimos lo siguiente en las categorías (Figura II.27):



Figura II.27. Gestión del modelo de evaluación. Categoría inválida.

Al hacer clic en “Guardar”, obtendremos el siguiente error: Formato de la categoría inválido. El formato a seguir para las categorías es de Letra “L” seguida de un punto “.”. Se puede ver mejor en la Figura II.28:



Figura II.28. Gestión del modelo de evaluación. Error de formato categoría.

Si el proceso ha ido bien, aparecerá el mensaje de “Categoría añadida”, y al desplegar las categorías, aparecerá la de nueva creación.(Figura II.29).

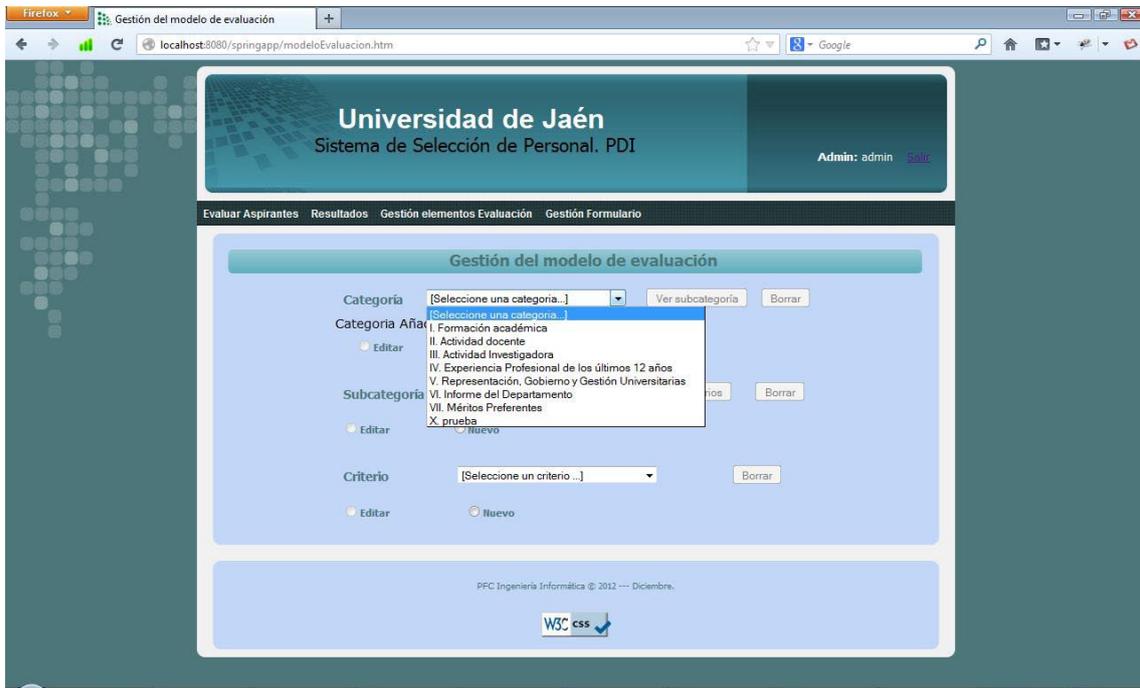


Figura II.29. Gestión del modelo de evaluación. Categoría añadida.

Para ver las subcategorías disponibles, haremos clic en “Ver Subcategorías”. En este caso, se ha seguido los pasos anteriores, pero en el apartado de Subcategoría, para añadir “X.1, X.2 y X.3”. Se puede ver el resultado en la Figura II.30:



Figura II.30. Gestión del modelo de evaluación. Subcategorías añadidas.

El formato a seguir para la subcategoría es Letra “L” seguido de punto “.” Y un número “D” y un punto “.”

Si no se sigue este formato, se indicará el error de formato inválido, como se indica en la siguiente Figura II.31:



Figura II.31. Gestión del modelo de evaluación. Error de formato subcategoría.

El procedimiento para los criterios es el mismo. El único aspecto a tener en cuenta sería el formato a seguir. Como se ve en el ejemplo, se han añadido tres criterios, que matienen en su identificador, el formato Letra “L”, punto “.”, numero, “D”, punto “.”, numero “D, punto “.”. Si no se sigue ese formato, obtendremos un error parecido al de la figura de las subcategorías(Figura II.32):

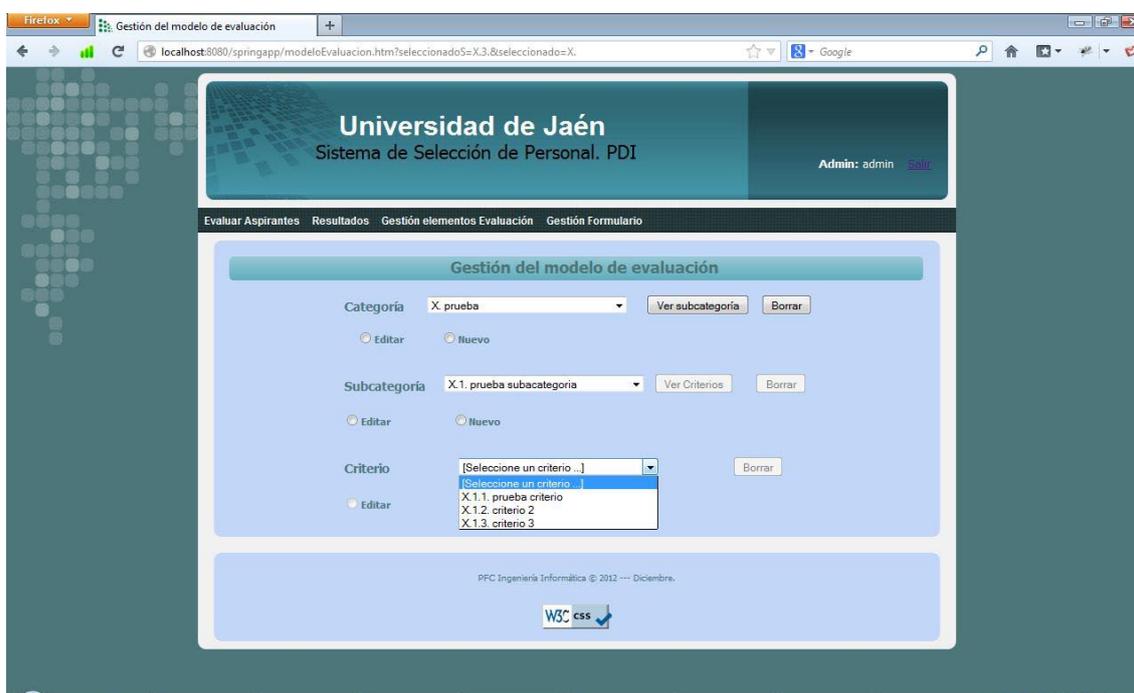


Figura II.32. Gestión del modelo de evaluación. Criterios añadidos.

4. Gestión del Formulario

4.1. Crear nueva Evaluación

En la opción “*Nuevo*”, establecemos el nombre de la Evaluación. En este caso, vamos a rellenar el campo del nombre, con Evaluación 2013 (Figura II.33).

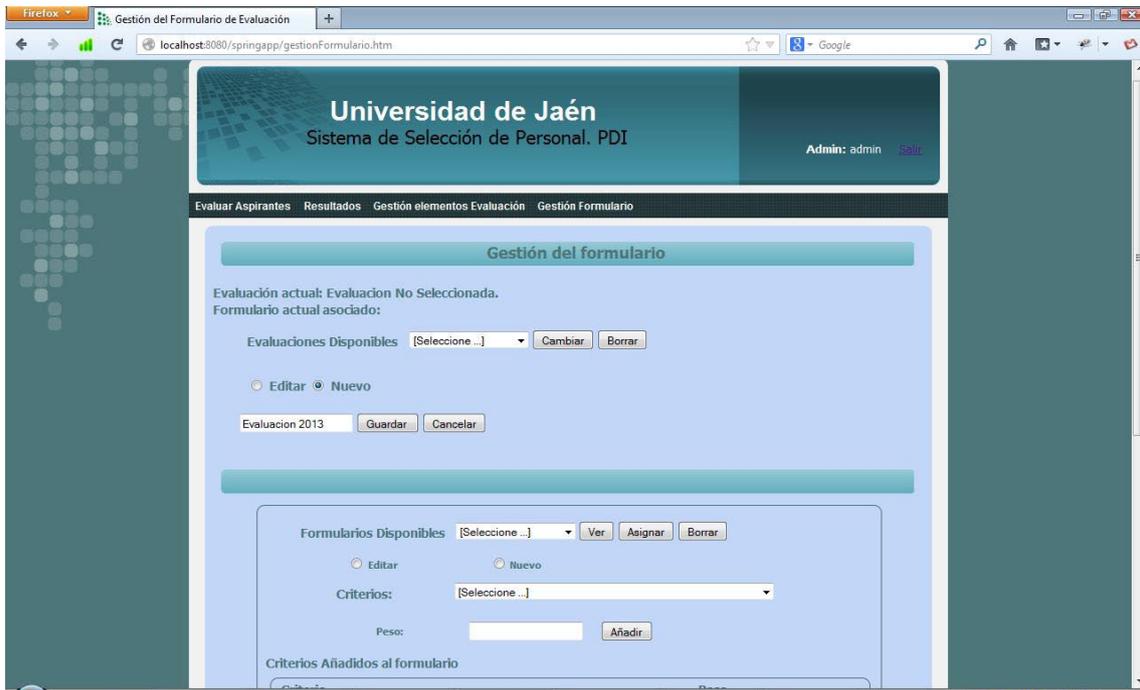


Figura II.33. Página gestión del formulario. Nueva Evaluación.

Si todo ha ido correctamente, obtendremos el mensaje de Evaluación Añadida, como podemos ver en la siguiente Figura II.34:

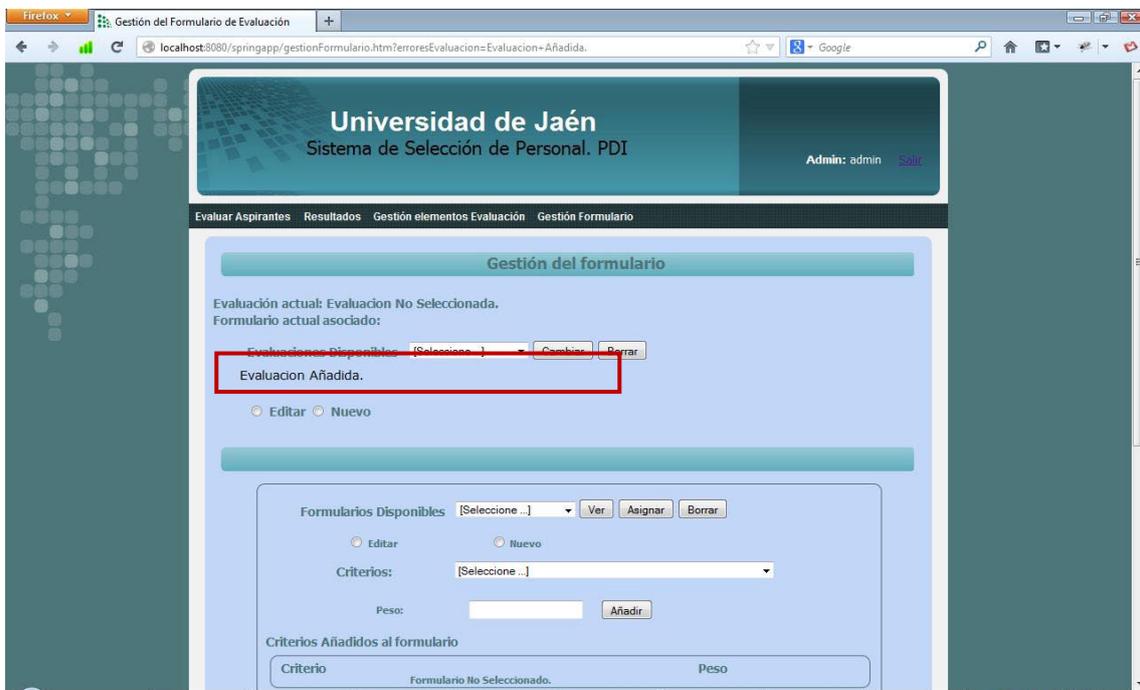


Figura II.34. Página gestión del formulario. Evaluación añadida.

Si por el contrario, nos olvidáramos de rellenar el nombre, obtendríamos el error de nombre Requerido (ver Figura II.35):



Figura II.35. Página de gestión del formulario. Error en nombre Evaluación.

Crear nuevo Formulario

El siguiente paso, sería crear un nuevo formulario. Los pasos son parecidos a los que hemos seguido para la creación de la Evaluación.

Introducimos el nombre, y pulsamos en Guardar. Si dejáramos en blanco el campo del nombre, obtendríamos un error de campo Requerido. (Ver Figura II.36)

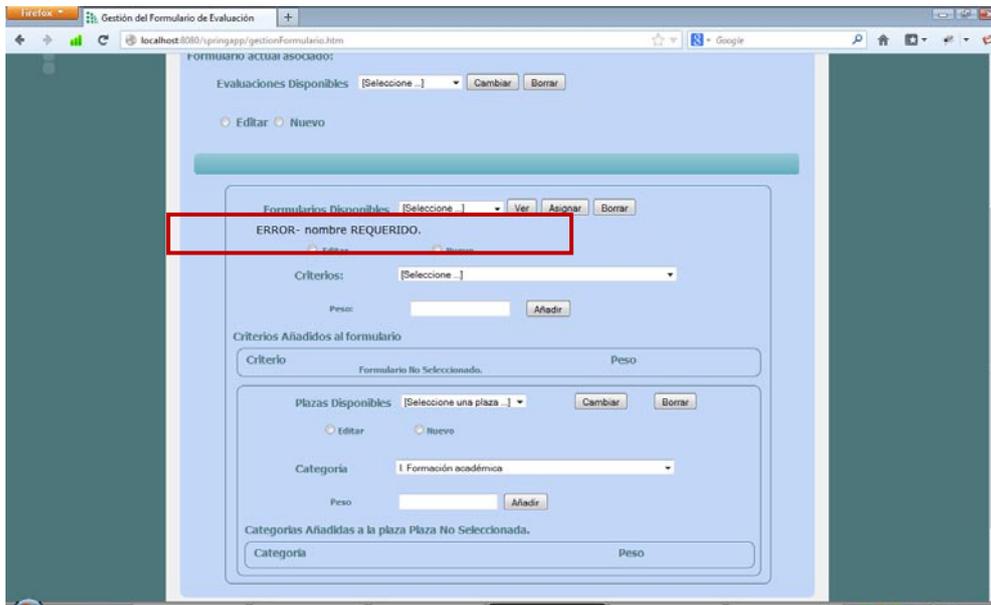


Figura II.36. Gestión del Formulario. Error en nombre de Formulario.

4.2. Crear nuevo Tipo de Plaza

Para la creación de una plaza nueva, hacemos clic en nuevo, en la misma pestaña de “Gestión del Formulario”. Una vez rellenado el campo (ver Figura II.37), hacemos clic en “Guardar”, y la plaza queda creada.

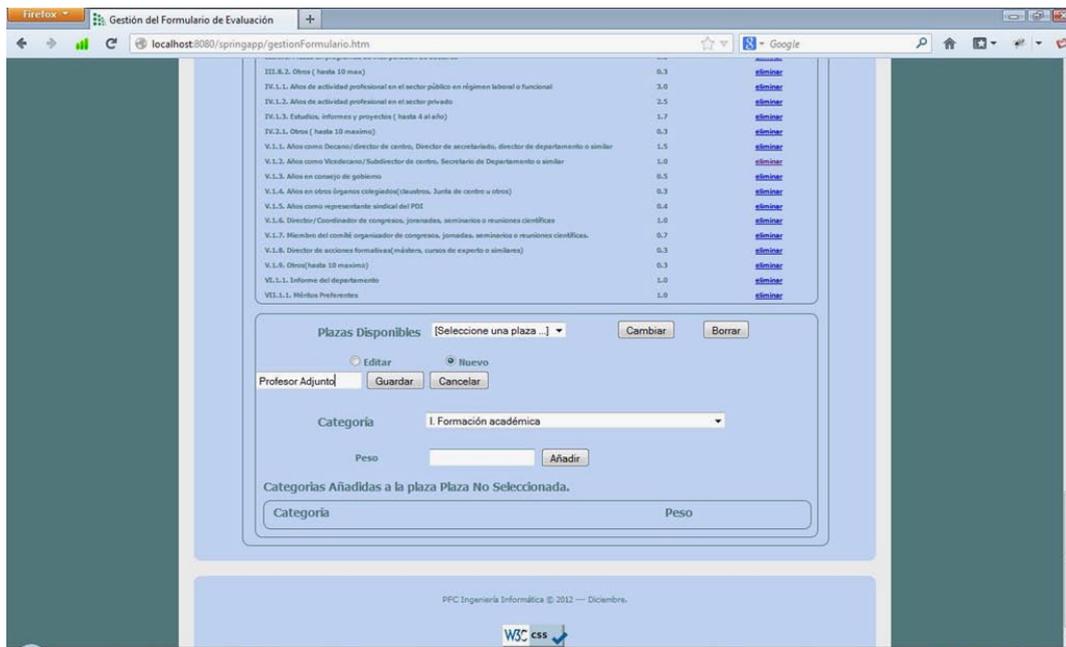


Figura II.37. Gestión del formulario de evaluación. Crear nueva plaza.

Si se deja el campo en blanco, se indicará un error de campo Requerido, como en la Figura II.38.

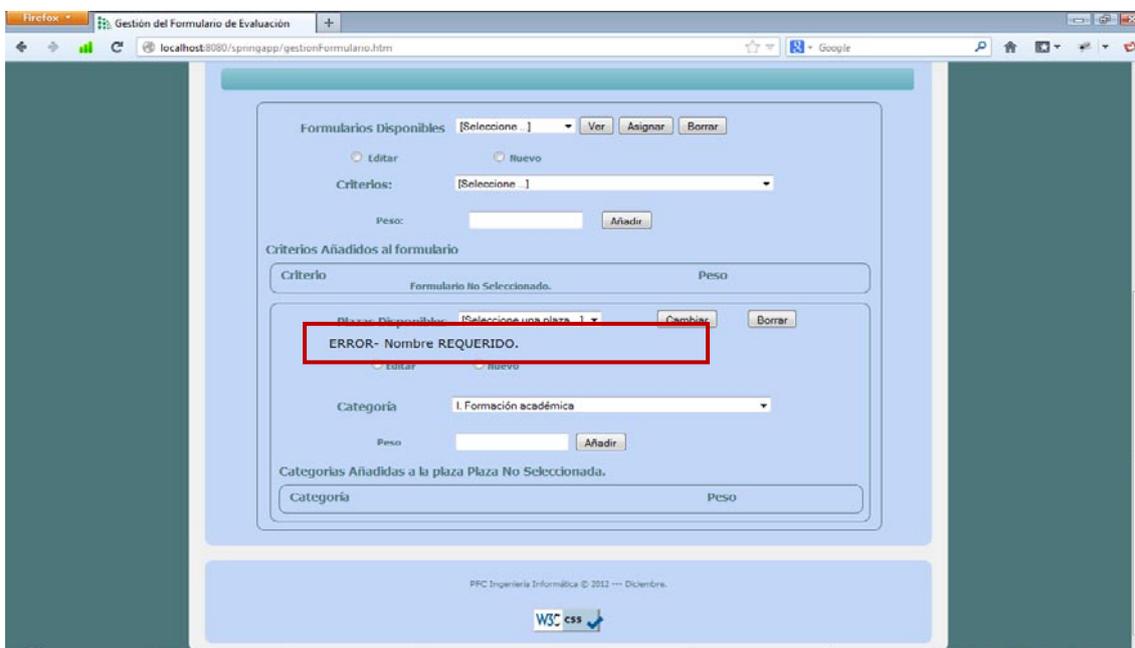


Figura II.38. Gestión del formulario de evaluación. Error nombre requerido Plaza.

4.3. Asignar un Formulario a una Evaluación

A continuación, seleccionamos una de las Evaluaciones para poder seguir. En nuestro caso, la Evaluación que acabamos de crear, Evaluacion 2013, y pulsamos en “Cambiar”, como se puede ver en Figura II.39:



Figura II.39. Página gestión del formulario. Cambiar Evaluación

Vemos en la siguiente Figura II.40 que la Evaluación Actual ha cambiado a Evaluación 2013.



Figura II.40. Gestión del formulario. Evaluación cambiada.

4.4. Añadir criterios a un Formulario

Una vez que hemos creado correctamente el formulario, pasamos a rellenarlo. Seleccionamos el Formulario de la lista, y pulsamos en ver (Figura II.41).



Figura II.41. Gestión del formulario de evaluación. Formulario añadido.

Al ser de nueva creación se indica el error “No hay criterios Asociados al Formulario Seleccionado”. Como podemos observar en la siguiente Figura II.42.

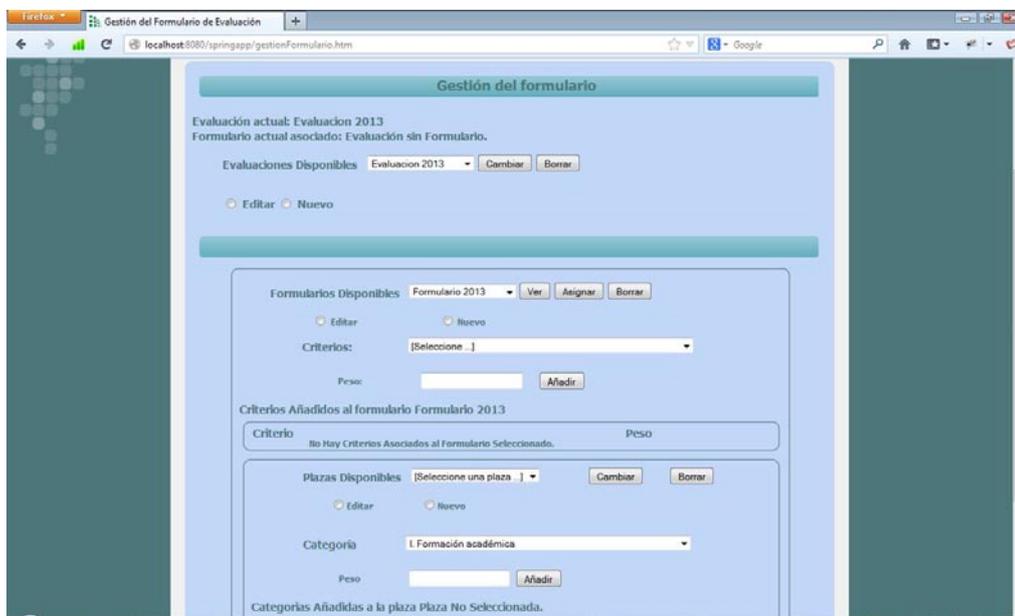


Figura II.42. Pagina de Gestión del formulario. Error no criterios asociados.

Añadir Criterios Al Formulario

Procederemos a añadir un nuevo criterio a la Evaluación seleccionada. Escogemos un nombre, y tras asignarle un peso, pulsamos en “Añadir”(ver Figura II.43).

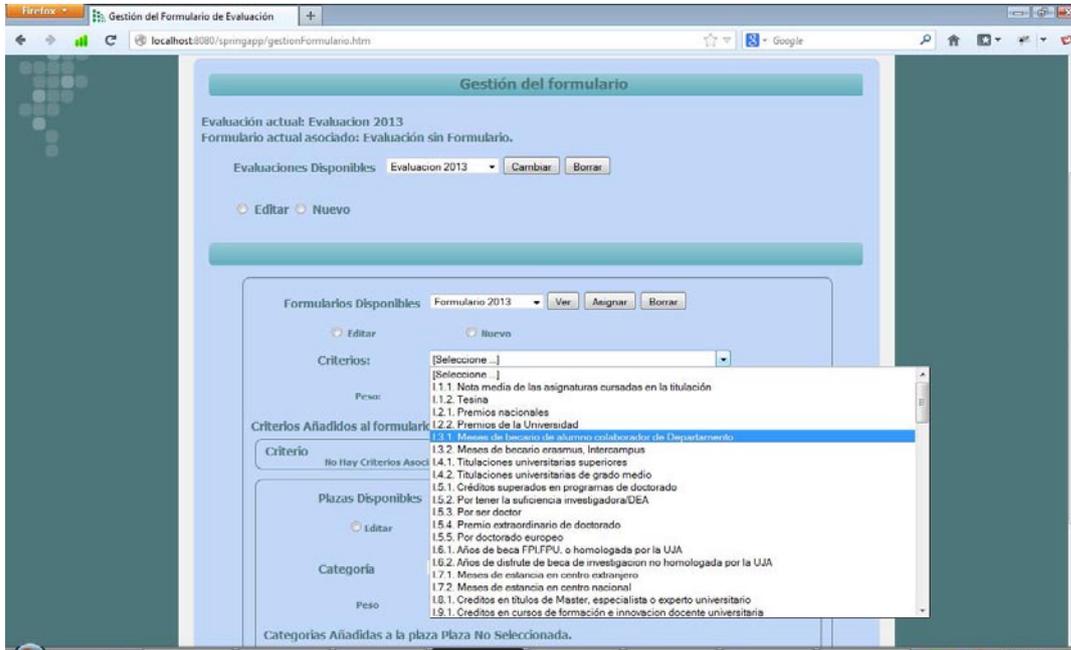


Figura II.43. Gestión del formulario de evaluación. Añadir Criterio.

Hecho esto, tendremos en la lista de criterios añadidos al formulario, el criterio que hemos escogido, con el peso asignado, como se puede ver en la Figura II.44:

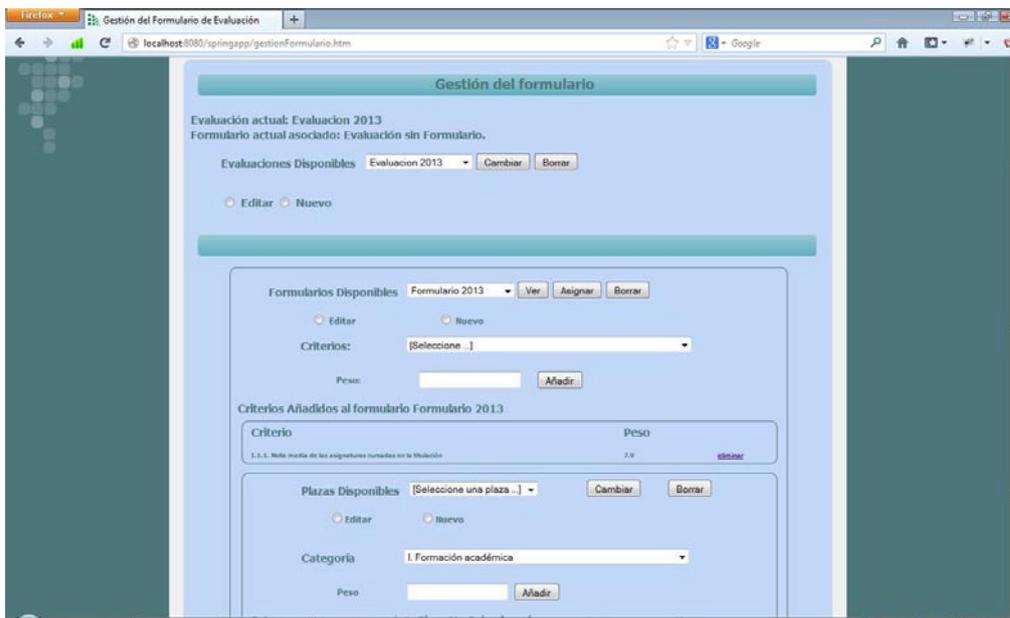


Figura II.44. Gestión del formulario de evaluación. Criterio añadido.

Error al introducir peso del criterio

En caso que el peso indicado sea inválido, indica con un error, que el formato es inválido(se ha introducido mal los decimales) o que no es un número (se ha escrito un carácter que no es un número). Como se puede ver en la Figura II.45:

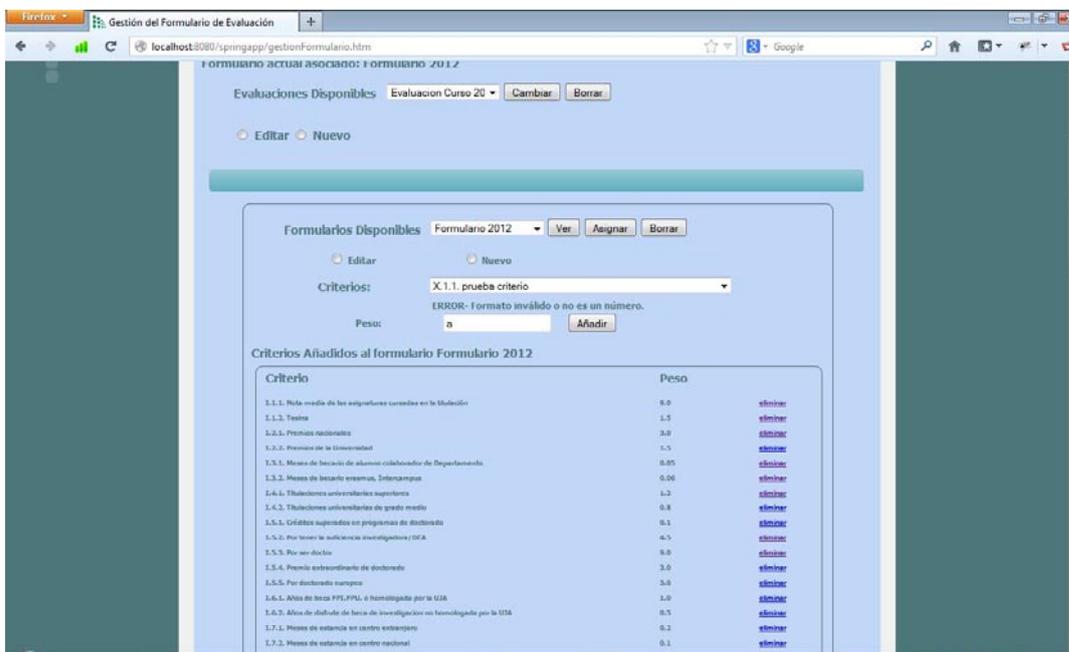


Figura II.45. Gestión del formulario de Evaluación. Error en peso.

Confirmación del borrado del criterio

Si quisiéramos eliminar un criterio de la lista del Formulario, seleccionaríamos la opción eliminar, de la parte derecha de la lista. Tras esto, se mostrará un mensaje para confirmar que queremos proceder con el borrado. En caso afirmativo, haremos clic en el cuadro de diálogo que nos muestra el sistema (Figura II.46).



Figura II.46. Gestión del formulario de evaluación. Borrar Criterio.

Una vez que ya sabemos cómo se añaden los criterios, veamos un ejemplo ya hecho. En esta Figura II.47, mostramos el Formulario 2012, que cuenta con los criterios y las valoraciones incluidos en el Baremo de la Universidad de Jaén.

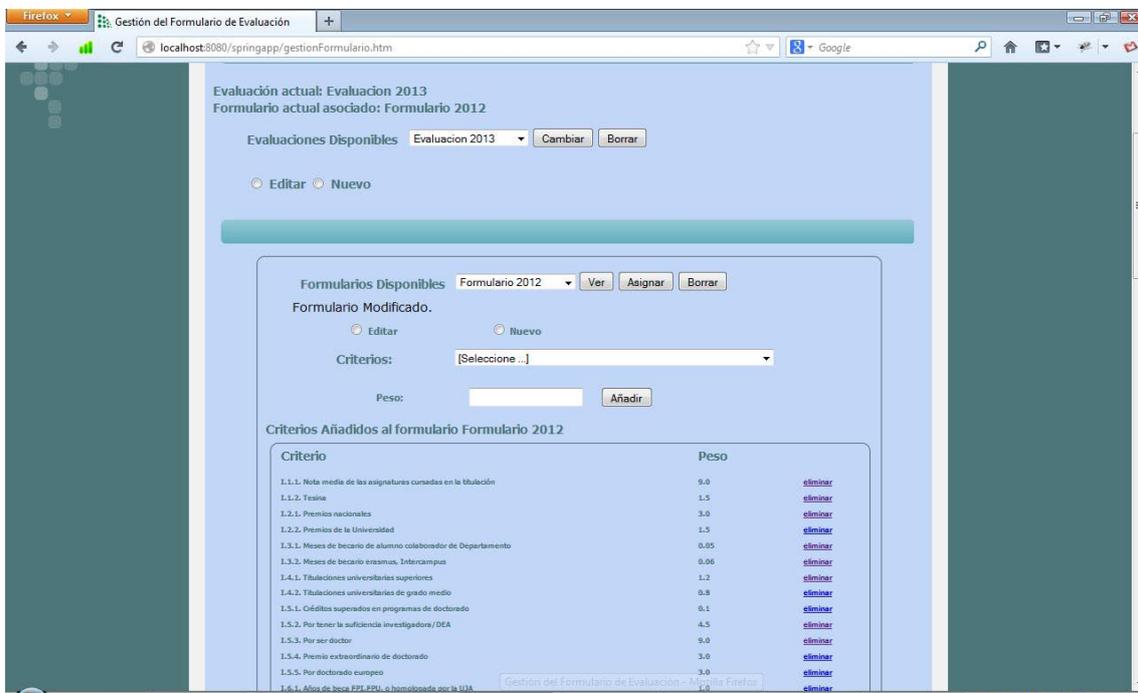


Figura II.47. Gestión de formulario de evaluación. Formulario 2012.

4.5. Añadir categorías a una Plaza

Una vez seleccionada una categoría (ver Figura II.48),

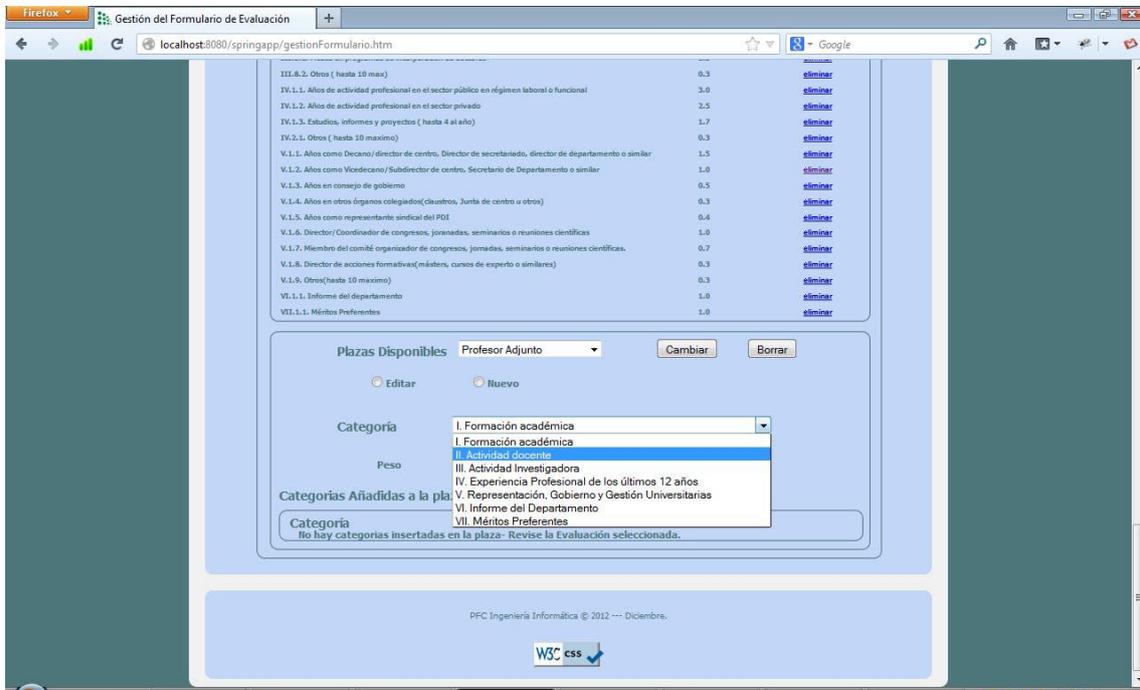


Figura II.48. Gestión del formulario de evaluación. Añadir Categoría.

y asignado un peso válido, pulsando en “Añadir”,el sistema nos muestra la lista de las Categorías añadidas a la plaza, como muestra la Figura II.49.

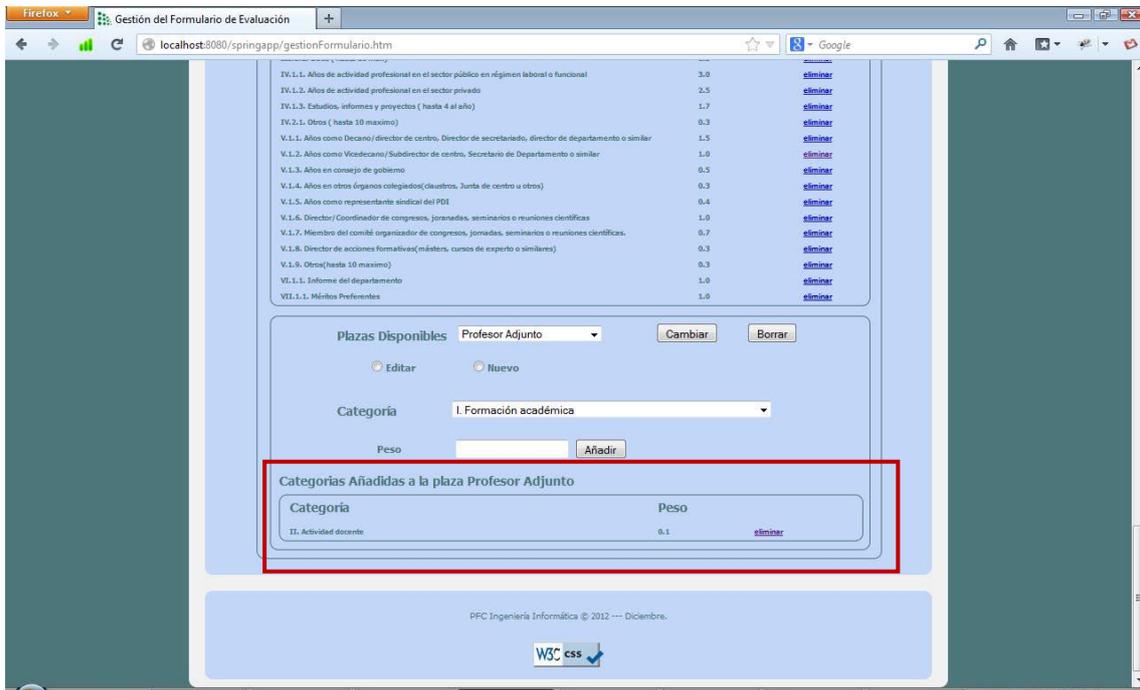


Figura II.49. Gestión del formulario de evaluación. Categorías añadidas a la plaza.

Introducir una categoría que ya está incluida.

Si al introducir una nueva categoría en la plaza, la categoría ya ha sido introducida anteriormente, obtendremos el error de que no es posible insertar la nueva categoría en la plaza (Figura II.50).

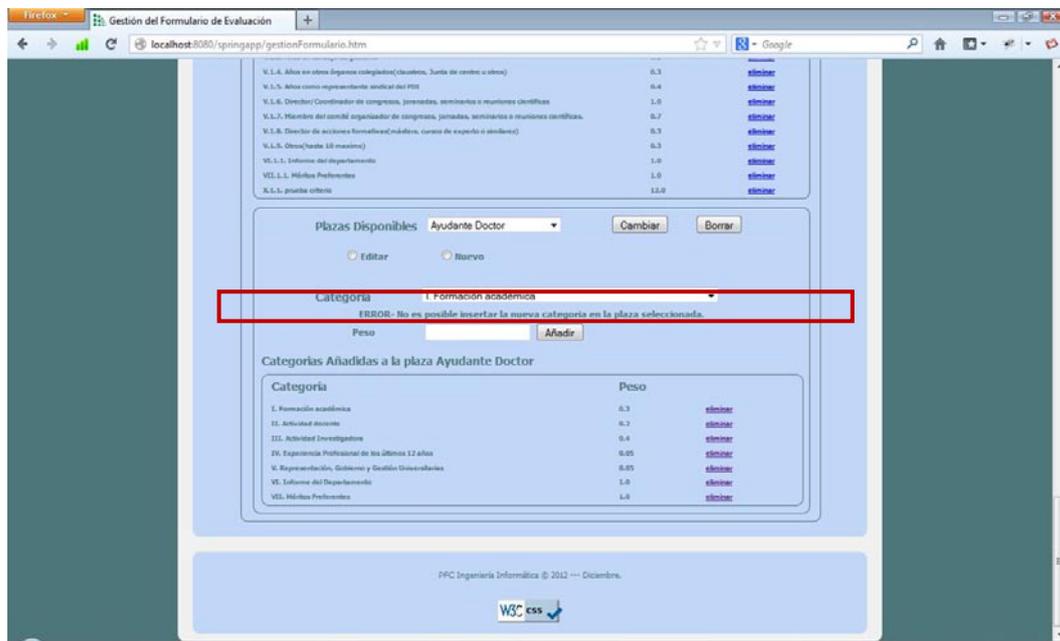


Figura II.50. Gestión del formulario de evaluación. Error categoría ya insertada.

Formato invalido al introducir la categoría en la plaza

Si al añadir una nueva categoría, seleccionamos la categoría de las categorías disponibles, y el peso que hemos indicado es incorrecto, no es posible añadir la categoría, y el sistema notifica un error que indica que o bien lo introducido es un numero que no cumple el formato requerido, o que no es un número. En ambos casos, el error es el mismo (Figura II.51).

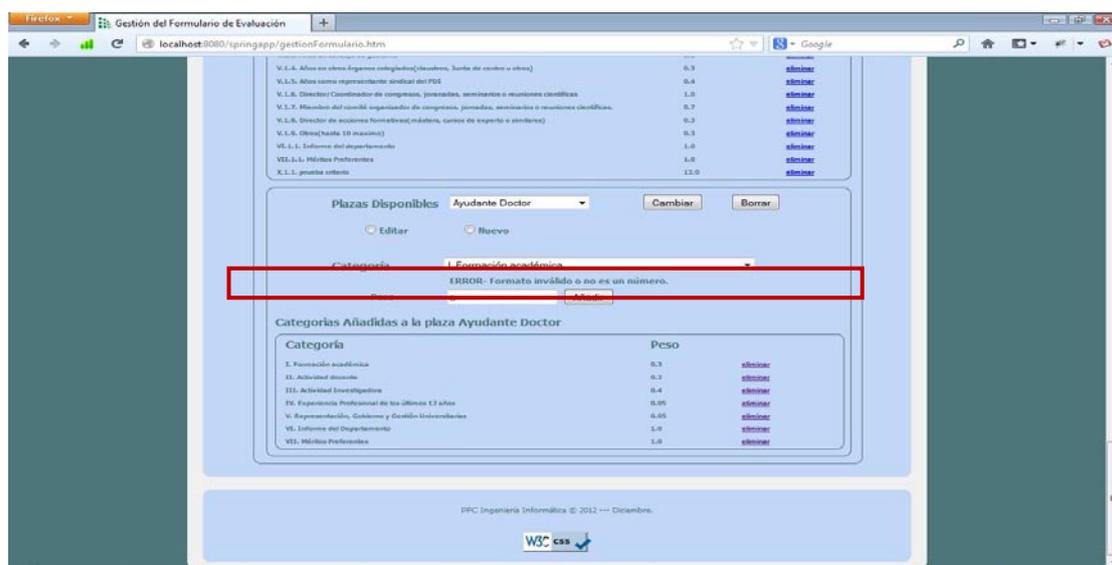


Figura II.51. Gestión del formulario de evaluación. Error en peso categoría.

En la siguiente Figura II.52, se expone un ejemplo completo, de cómo quedaría la lista de categorías, para una plaza real. En este caso, para el Ayudante Doctor.

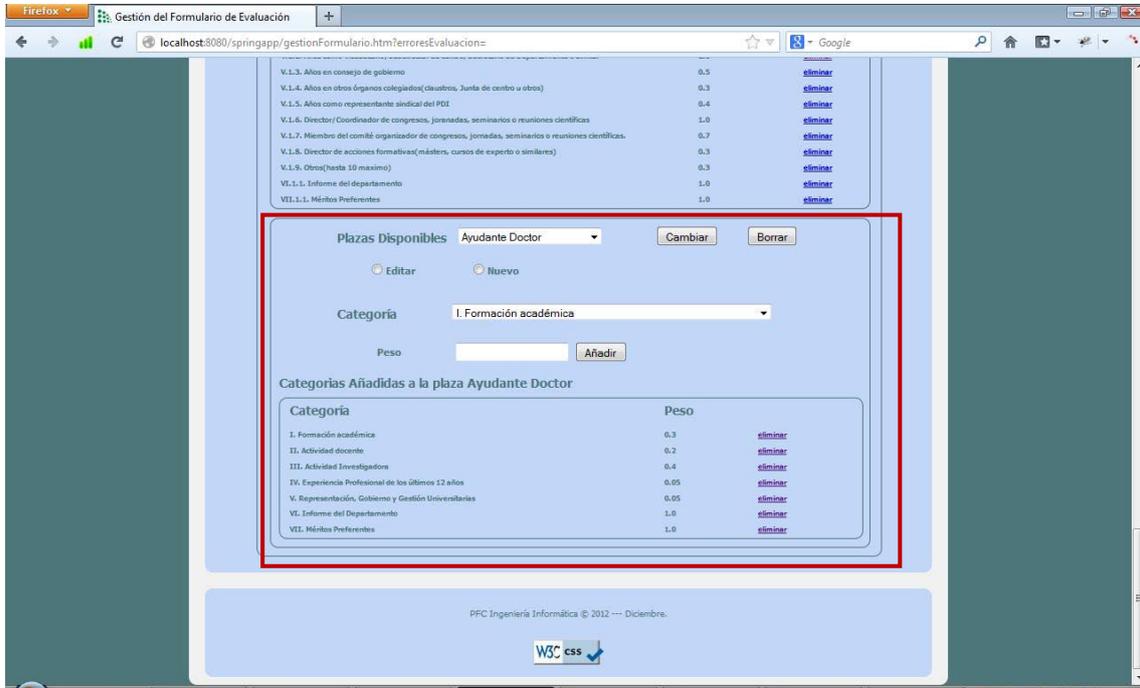


Figura II.52. Gestión del formulario de evaluación. Plaza Ayudante Doctor.

4.6. Borrar un Formulario.

Si se quiere borrar un formulario, los pasos son los mismos que para borrar una Evaluación, o una plaza. Seleccionamos entre los formularios disponibles, el que queremos eliminar, y hacemos clic en “Borrar”. Se nos pedirá una confirmación, ya que el borrado de un formulario no es reversible (ver Figura II.53).

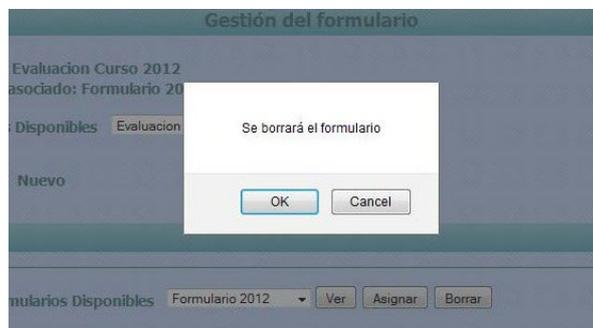


Figura II.53. Gestión del formulario de evaluación. Borrar formulario.

En caso de que el formulario esté siendo usado, como en este caso, en una evaluación, no es posible borrarlo, para no perder los datos . El sistema indica un error de formulario en uso, como indica la Figura II.54:

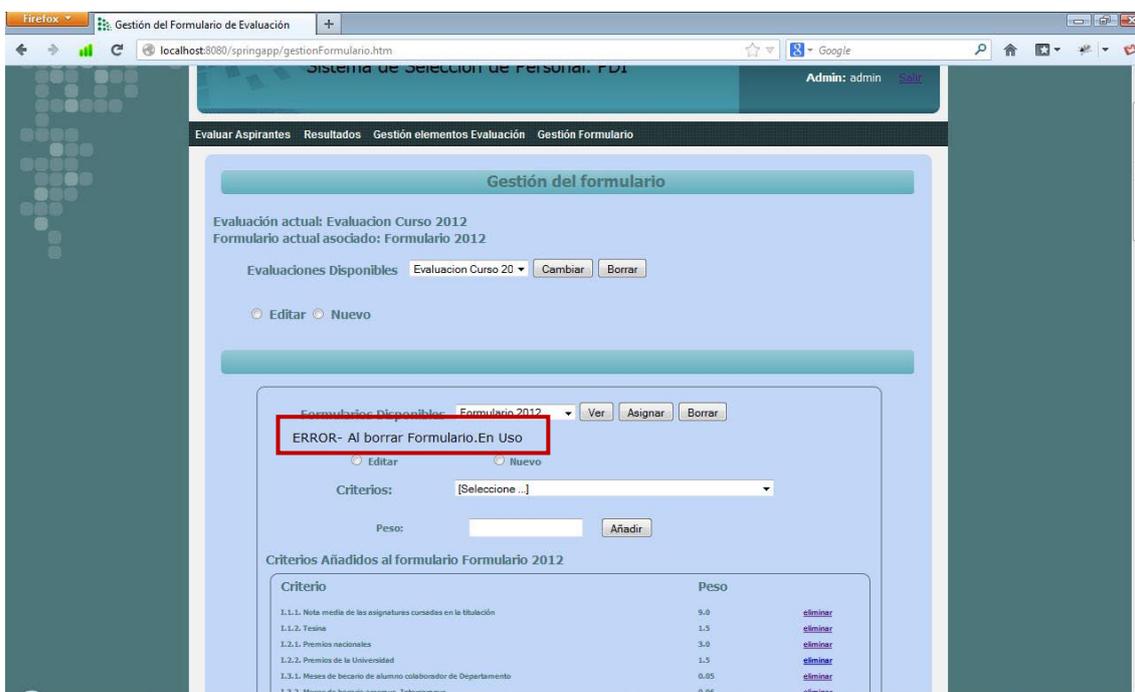


Figura II.54. Gestión del formulario de evaluación. Error formulario en uso.

4.7. Borrar una plaza.

En caso de que se quisiera borrar una plaza, hacemos clic en borrar, y se nos pedirá el siguiente mensaje de confirmación (Figura II.55):

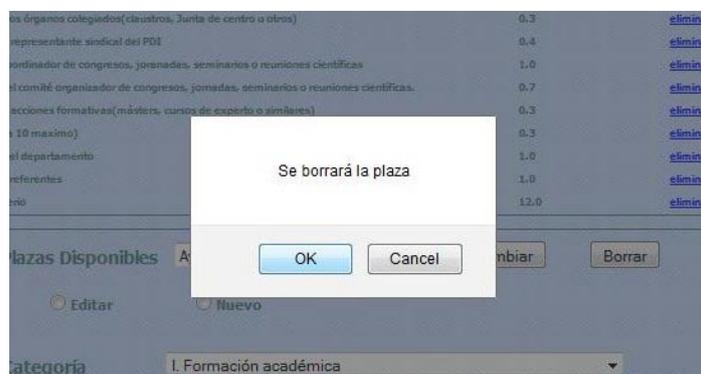


Figura II.55. Gestión del formulario de evaluación. Borrar una plaza completa.

Si la plaza está siendo usada en una evaluación, el sistema nos indicará que está en uso, y no se permite el borrado (ver Figura II.56). Esto se hace, para evitar perder datos de Evaluaciones pasadas, ya que queremos mantener un histórico.

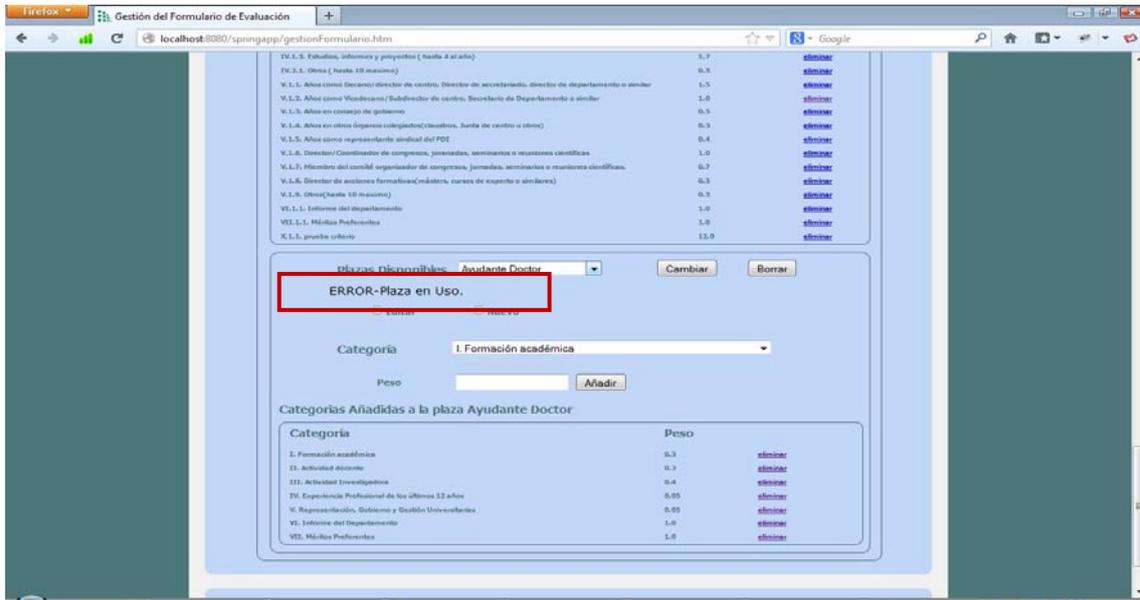


Figura II.56. Gestión del formulario de evaluación. Error plaza en uso.

4.8. Borrar una Evaluación

Para eliminar una Evaluación, seleccionaríamos una evaluación de las disponibles, y hacemos clic en “Borrar”. Hay que tener en cuenta, que al borrar la Evaluación, se perderán todos los datos de los aspirantes, plazas, y cálculos que se hayan realizado anteriormente. No es posible volver hacia atrás una vez se haya eliminado. Por ello, en este punto, se pide una confirmación para ver si estamos seguros del borrado (Figura II.57).



Figura II.57. Gestión del formulario de evaluación. Borrar Evaluación.

Una vez confirmamos el borrado, obtenemos el mensaje de “Evaluación Borrada”. La evaluación no está en la lista de Evaluaciones disponibles. No será posible como se indica anteriormente, recuperar una evaluación borrada.

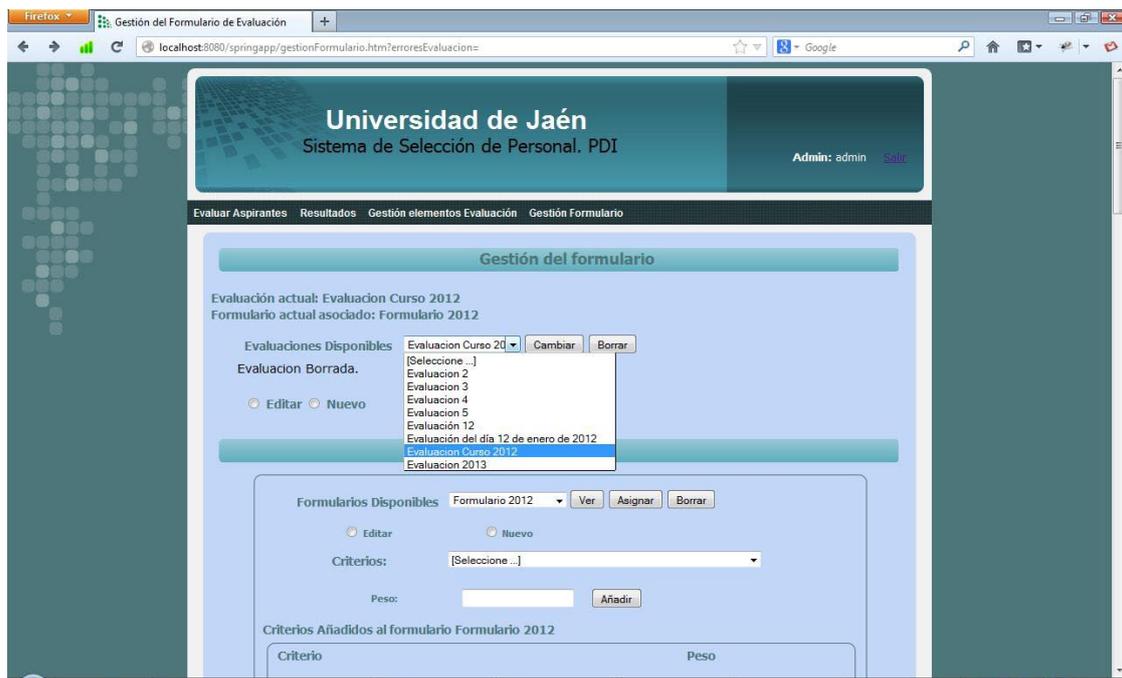


Figura II.58. Gestión del formulario de evaluación. Evaluación borrada

Anexo III.

*Esquema seguido para la definición
de los elementos XML.*

Este anexo, viene a aclarar la sintaxis utilizada en el “Schema XML” que se comentó en el Capítulo 3 del presente documento.

Esta definición, viene a detallar los elementos que van a ser válidos en el documento XML generado a partir del mismo.

Se comentarán los elementos que van a formar parte de la estructura, así como los elementos que van a ser válidos, y los que no van a ser permitidos en el mismo.

La finalidad de este esquema, es que el sistema obtenga los datos en el formato XML bien formado, y válido. Así, hacemos que los datos vengan con la misma estructura, y sean totalmente legibles por el sistema. Conseguimos extraer de manera uniforme los datos, y evitamos un procesamiento más complejo, que sería necesario en caso de que el XML no fuera validado con este esquema.

A continuación, se muestra el esquema total, generado con la herramienta XML Schema, que fue la escogida para realizarlo.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation>PDI. Universidad de Jaén</xs:documentation>
  </xs:annotation>
  <xs:simpleType name="TipoIdCateg">
    <xs:annotation>
      <xs:documentation>El formato será "D." siendo D letra </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:pattern value="\D{1,3}\."/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="TipoIdSubcateg">
    <xs:annotation>
      <xs:documentation>El formato será "D.N." siendo D letra y N
numero</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:pattern value="\D{1,3}\.\d{1,3}\."/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="TipoIdCriterio">
    <xs:annotation>
      <xs:documentation>El formato será "D.N.N." siendo D letra y N
numero</xs:documentation>
    </xs:annotation>
  </xs:simpleType>

```

```

        <xs:restriction base="xs:string">
            <xs:pattern value="\d{1,3}\.\d{1,3}\.\d{1,3}\."/>
        </xs:restriction>
    </xs:simpleType>
    <!-- Definición de los datos personales del Aspirante -->
    <xs:complexType name="DatosPersonales">
        <xs:sequence>
            <xs:element name="DNI" type="xs:string" nillable="false" minOccurs="1"
maxOccurs="1"/>
            <xs:element name="Apellido1" type="xs:string" minOccurs="1"/>
            <xs:element name="Apellido2" type="xs:string" />
            <xs:element name="Nombre" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="FechaNacimiento" type="xs:date" maxOccurs="1"/>
            <xs:element name="Localidad" type="xs:string" maxOccurs="1"/>
            <xs:element name="ProvinciaNacimiento" type="xs:string" maxOccurs="1"/>
            <xs:element name="Nacionalidad" type="xs:string" maxOccurs="1"/>
            <xs:element name="Domicilio" type="xs:string" maxOccurs="1"/>
            <xs:element name="eMail" type="xs:string" maxOccurs="1"/>
            <xs:element name="Municipio" type="xs:string" maxOccurs="1"/>
            <xs:element name="Provincia" type="xs:string" maxOccurs="1"/>
            <xs:element name="CodigoPostal" type="xs:string" maxOccurs="1"/>
            <xs:element name="Telefono" type="xs:string" maxOccurs="2"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="Categoria">
        <xs:sequence>
            <xs:element name="Subcategoria" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element
name="Criterio"
maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element
name="valorCriterio" maxOccurs="1"/>
                                </xs:sequence>
                                <xs:attribute name="idCriterio"
type="TipoIdCriterio" use="required"/>
                                <xs:attribute
name="nombreCriterio" type="xs:string" use="required"/>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                    <xs:attribute name="idSubcategoria" type="TipoIdSubcateg"
use="required"/>
                    <xs:attribute name="nombreSubcateg" type="xs:string"
use="required"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="idCategoria" type="TipoIdCateg" use="required"/>
        <xs:attribute name="nombreCategoria" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:annotation>
        <xs:documentation>
            Definición de los elementos de evaluacion
        </xs:documentation>
    </xs:annotation>

```

```

        </xs:documentation>
    </xs:annotation>
    <xs:element name="Aspirante">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="DatosPersonales" type="DatosPersonales"/>
                <xs:element name="TipoPlaza" type="xs:string" />
                <xs:element name="Categ" type="Categoria" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>

```

En ella podemos destacar los siguientes elementos:

Primeramente, la siguiente parte, es la que da mayormente, el esquema a seguir:

```

    <xs:element name="Aspirante">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="DatosPersonales" type="DatosPersonales"/>
                <xs:element name="TipoPlaza" type="xs:string" />
                <xs:element name="Categ" type="Categoria"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

```

Como se puede observar, el elemento principal, es el “Aspirante”, compuesto a su vez, por 3 elementos, que vienen a ser Datos personales, en el que se van a incluir otros elementos que detallaremos posteriormente. A continuación, forzamos a que haya un nombre de Tipo de Plaza, necesario para saber a qué plaza va a optar el aspirante en cuestión.

A continuación, nos encontramos con las categorías, que tendrán una estructura interna que también se detalla más adelante. Aquí, nos encontramos con que Categoría va a poder aparecer varias veces, para dar lugar a que aparezcan tantas categorías sean necesarias. Esto se consigue, poniendo la propiedad maxOccurs a unbounded.

En la parte de Datos personales, con tamos con la siguiente estructura:

```

    <!-- Definición de los datos personales del Aspirante -->
    <xs:complexType name="DatosPersonales">
        <xs:sequence>
            <xs:element name="DNI" type="xs:string" nillable="false" minOccurs="1"
maxOccurs="1"/>

```

```

<xs:element name="Apellido1" type="xs:string" minOccurs="1"/>
<xs:element name="Apellido2" type="xs:string" />
<xs:element name="Nombre" type="xs:string" minOccurs="1"
maxOccurs="1"/>
<xs:element name="FechaNacimiento" type="xs:date" maxOccurs="1"/>
<xs:element name="Localidad" type="xs:string" maxOccurs="1"/>
<xs:element name="ProvinciaNacimiento" type="xs:string"
maxOccurs="1"/>
<xs:element name="Nacionalidad" type="xs:string" maxOccurs="1"/>
<xs:element name="Domicilio" type="xs:string" maxOccurs="1"/>
<xs:element name="eMail" type="xs:string" maxOccurs="1"/>
<xs:element name="Municipio" type="xs:string" maxOccurs="1"/>
<xs:element name="Provincia" type="xs:string" maxOccurs="1"/>
<xs:element name="CodigoPostal" type="xs:string" maxOccurs="1"/>
<xs:element name="Telefono" type="xs:string" maxOccurs="2"/>
</xs:sequence>

```

Los elementos, van a tener un número mínimo de apariciones (“minOccurs”) que van a forzar, a que por lo menos, aparezcan una vez. Los elementos marcados con “maxOccurs”, van a exigir, que como máximo, haya el número de elementos que se indica en esa restricción.

Nos encontramos además, con la etiqueta “**nillable**” en la que indicamos, que en el caso de los datos personales, es indispensable que aparezca un DNI.

Los tipos de datos permitidos, son cadena de caracteres (“string”) y fecha (“date”) para la Fecha de nacimiento del Aspirante.

Para las categorías, hay una estructura un más compleja:

```

<xs:complexType name="Categoria">
  <xs:sequence>
    <xs:element name="Subcategoria" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Criterio"
maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element
name="valorCriterio" maxOccurs="1"/>
              </xs:sequence>
              <xs:attribute
name="idCriterio" type="TipoIdCriterio" use="required"/>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

<xs:attribute
name="nombreCriterio" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="idSubcategoria"
type="TipoIdSubcateg" use="required"/>
<xs:attribute name="nombreSubcateg" type="xs:string"
use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="idCategoria" type="TipoIdCateg" use="required"/>
<xs:attribute name="nombreCategoria" type="xs:string" use="required"/>
</xs:complexType>

```

En este caso, tenemos un elemento principal, que va a ser la categoría, formado por el Identificador de la Categoría (“idSCategoria”), y por el nombre de la categoría. Ambos van a ser obligatorios. Esta a su vez, va a estar formada, por otro elemento Subcategoría, que también se permite que se repita varias veces a lo largo del documento XML, ya que aparece la propiedad maxOccurs a “unbounded”. Va a estar compuesto, del identificador (“idSubcategoria”) y de un nombre para la misma (“nombreSubcateg”).

Anidado dentro de Subcategoría, nos encontramos con el elemento Criterio, que es el que en verdad es importante.

En él, se distinguen varios elementos. El nombre del criterio, que va a ser obligatorio (“required”), el identificador del criterio, (“tipoIDCriterio”), que también va a ser obligatoria, así como el valor que va a tener el criterio. En este caso, será un único valor, para que no haya problemas a la hora de la lectura, y el procesamiento posterior.

Solamente nos queda explicar las restricciones que se establecen para los identificadores tanto de las Categorías, como para las Subcategorías y Criterios que aparezcan en el documento XML. Deberán de cumplir estos formatos, para, como hemos comentado, que coincidan con los elementos que vamos a tener almacenados en la base de datos, y sean totalmente legibles por el sistema.

```

<xs:simpleType name="TipoIdCateg">
  <xs:annotation>
    <xs:documentation>El formato será "D." siendo D letra
</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\D{1,3}\."/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="TipoIdSubcateg">
  <xs:annotation>
    <xs:documentation>El formato será "D.N." siendo D letra y N
numero</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\D{1,3}\.\d{1,3}\."/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="TipoIdCriterio">
  <xs:annotation>
    <xs:documentation>El formato será "D.N.N." siendo D letra y N
numero</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="\D{1,3}\.\d{1,3}\.\d{1,3}\."/>
  </xs:restriction>
</xs:simpleType>

```

En primer lugar, nos encontramos con la restricción para el tipo de la Categoría. (“tipoIdCateg”). Este elemento, va a estar formado, por una letra, seguido de un punto “.”.

Para la subcategoría, el formato, como se indica en el documento es "D.N." siendo D letra y N numero.

Y para finalizar, el formato que se tendrá que cumplir para que el identificador del criterio cumpla con el esquema que se ha detallado será de "D.N.N." siendo D letra y N numero.

Con todas estas restricciones explicadas anteriormente, podemos obtener un XML además de bien formado, válido, y así poder un documento XML como el que se refleja en el Capítulo 3.