FindYourWay

"Aplicación web: buscador de perfiles profesionales"

Trabajo Final de Grado

GRADO EN INGENIERÍA INFORMÁTICA

Autor

Cristina Cano Sandín

Consultor

Juan José Cuadrado Gallego

Junio 2014

Índice

Cristina Cano Sandín....1

Juan José Cuadrado Gallego....1

1. Resumen....4

2. Abstract....4

3. Introducción....4

Contexto....4

Objetivos....5

Alcance del proyecto....5

Entorno tecnológico....7

4. Organización del proyecto....9

Ciclo de vida clásico o en cascada....9

Descomposición en actividades (WBS)....10

Planificación del proyecto....11

5. Análisis de requisitos....13

Introducción....13

Especificación de requisitos....13

Definición de actores....14

Requisitos funcionales....14

1. Subsistema curricular....14

2. Subsistema Orientador....31

3. Subsistema Búsqueda de candidato....32

4. Subsistema Plan carrera....33

Subsistema Login....35

Requisitos no funcionales....37

6. Diseño técnico....38

Modelado clase-responsabilidad-colaborador (CRC)....39

Modelo del dominio....49

Modelo entidad-relación....49

Arquitectura software del sistema....50

7. Pruebas de software....65

8. Seguimiento del proyecto....78

10. Conclusión 79

11. Licencia 79

12. Anexo 79

13. Bibliografía 82

1. Resumen

La empresa MAC presentaba carencias en su gestión de la selección de candidatos para un determinado puesto, teniendo desaprovechadas las cualidades de muchos de sus empleados.

Con el desarrollo de nuestro producto software a medida, MAC tiene una gestión rápida y centralizada de los curriculum de sus empleados, puede tener acceso de forma distribuida a los mismos.

Cuenta con que sus empleados pueden tener el CV actualizado a través del aplicativo mejorando sus posibilidades de ser seleccionados para trabajar en otros proyectos y no quedar en staf. Las búsquedas son mucho más efectivas por la actualización continua de cualificación de los empleados, además de presentar la aparición de posibles aspirantes a los puestos.

Con todo lo realizado el negocio de la empresa se verá fortalecido y se incrementará sus ganancias.

La empresa ha quedado realmente satisfecha con el producto.

2. Abstract

The MAC company had shortcomings in its management of the selection of candidates for a given position, having missed the qualities of many of its employees.

With the development of our software MAC, will have a quick and centralized management of employees' curriculum and also can have access to them in a distributed way.

Expects that its employees must have the updated CV via the application to improve its chances of being selected to work on other projects and not be in staff. The searches are much more effective for the continuous updating of skills of employees, in addition to presenting the appearance of potential candidates for the posts.

Finally, the company's business will be strengthened and its profits will increase.

The company has been absolutly pleased with the product.

3. Introducción

Contexto

La empresa "MAC S.A" es una pequeña consultora de 250 empleados, en la que la línea principal de negocio es el outsourcing.

Cuando una empresa cliente le solicitaba un recurso, el departamento de RR.HH buscaba el perfil en la base de datos de empleados, el resultado de la búsqueda únicamente mostraba la categoría profesional de la última experiencia laboral de cada uno de los empleados que encontraba dicha

búsqueda. El porcentaje de satisfacción del cliente era muy alto, un 85%, pero la otra parte implicada, es decir, el empleado, no siempre quedaba satisfecho con la asignación al nuevo proyecto.

Algunos empleados se encontraban con que no contaban con algunas de las habilidades que se le requerían para el desempeño del trabajo en el nuevo proyecto. Este problema les obligaba a ser autodidactas, y además, las nuevas competencias adquiridas por el empleado con su autoformación no quedaban reflejadas.

Dentro del grupo de empleados que se encontraban insatisfechos con esta situación, el 50% dejaba la compañía yéndose a otras empresas molesto por la mala gestión que se realizaba respecto al capital humano.

En "MAC S.A" preocupados por la pérdida del 15% de sus clientes y del alto porcentajes de empleados que abandonaban la empresa se plantearon que debían realizar cambios en la gestión de los RR.HH.

Objetivos

1.2.1 Objetivo general

El software "*FindYourWay*" nace de la necesidad de "MAC S.A" de gestionar adecuadamente la selección de los recursos para un determinado puesto en las empresas cliente, asegurándose que el recurso esté cualificado para el puesto, motivado y lo más importante, es que todas las competencias que adquiera la persona en el desempeño de su actividad laboral se reflejen en el sistema, de esta forma las búsquedas futuras serán eficientes.

1.2.2 Objetivos específicos

Uno de los objetivos es ofrecer una gestión efectiva del plan de carrera de los empleados, para ello, se debe ofrecer al empleado la posibilidad de decantarse hacia un área profesional específica.

Otro de los objetivos es realizar una búsqueda de candidatos adecuados para un determinado puesto.

Y finalmente, tener actualizado el currículo de los empleados con todas las aptitudes desarrolladas y adquiridas en el nuevo proyecto.

Alcance del proyecto

1.3.1 Descripción de necesidades

El desarrollo del software "*FindYourWay*" tiene que contemplar los siguientes requerimientos:

El proceso comienza con la necesidad de realizar una búsqueda de la persona adecuada, una vez se recibe la petición de un determinado perfil por parte de una empresa cliente.

El gestor de RR.HH entra en la plataforma, se loga y accede a la “Búsqueda de candidatos”. Podrá realizar la búsqueda por medio de tres parámetros por competencias/habilidades, por categoría profesional y, por conocimientos requeridos. El resultado de la búsqueda son los perfiles que cumplen todos los parámetros especificados. En el caso de no cumplir todos los requisitos, aparecerán como resultado los candidatos que cumplen parcialmente los parámetros (aspirante). De esta forma, el gestor podrá revisar sus planes de carrera y añadir los cursos que necesitan para que en un futuro próximo cumplan el perfil requerido por el cliente.

El gestor de RR.HH tiene otra funcionalidad el sistema, el “Alta de empleado”. Aquí aparece un formulario con los datos necesarios para que pueda meter el currículo con las siguientes secciones bien diferenciadas: datos personales, experiencia profesional, estudios, competencias y conocimientos.

El empleado es de los actores fundamentales en nuestro software, éste entra en la plataforma y puede ver tres secciones, su “Currículo”, “Plan de Carrera” y “Orientador”:

- En “Currículo” puede actualizar el currículo introduciendo una nueva experiencia laboral, nuevos conocimientos adquiridos y nuevas competencias.
- En “Plan de Carrera” aparecen los cursos a realizar y los que tiene ya completados.
- En “Orientador” puede marcar una rama profesional. La plataforma mostrará todos los posibles perfiles profesionales existentes y el usuario podrá marcar cual es el perfil que más le interesa para crecer en esa línea profesional. Solo tendrá la posibilidad de marcar un perfil. Por ejemplo, un programador de base datos puede elegir la rama “Administración” y seleccionar la categoría “Base de Datos”, en su plan de carrera se añadirán los cursos que deberá realizar para poder optar a esa categoría profesional. Cuando el empleado complete todos los cursos le otorgarán el perfil profesional en su currículo.

1.3.2 Requerimientos funcionales iniciales

El aplicativo tiene definido en su alcance los siguientes subsistemas:

- Subsistema Búsqueda de candidatos

El subsistema de “Búsqueda de candidatos” se encargará de encontrar uno o varios candidatos según los parámetros de búsqueda, quedando bien diferenciados tres parametros de búsqueda por perfil, por competencias y por conocimientos.

- Subsistema Plan carrera profesional

En este subsistema, el empleado encontrará la formación que ha realizado, superada o no superada, y la formación que tiene marcada como objetivo y que está pendiente de realizar.

El gestor de RRHH podrá añadir formación al plan de carrera de cualquier empleado.

- Subsistema Currículo

En el subsistema “Currículo” encontramos un formulario con las siguientes áreas diferenciadas: datos personales, experiencia profesional, estudios, competencias y conocimientos. En este subsistema se puede dar de alta por primera vez, rellenando los datos de cada área, o bien, modificar cada una de ellas.

- Subsistema Orientador

En el subsistema “Orientador” muestra un listado de las clasificaciones profesionales. Cuando seleccionamos una de las clasificaciones aparecen todas las categorías profesionales que pertenecen a esta. Con el listado de categorías profesionales desplegado el empleado puede escoger únicamente una de ellas, esto activa la funcionalidad de agregar cursos en su plan de carrera, los cursos añadidos son los correspondientes a otorgar dicha categoría al empleado.

1.3.3 Funcionalidades propuestas para nuevas versiones

Para reducir costes, se propone que los cursos de formación presencial, que deben realizar los empleados de forma obligatoria para llevar a cabo las metas de su plan de carrera, sean documentados por expertos en el tema de la propia empresa. Se creará un subsistema donde los empleados puedan estudiar dichos cursos, para más tarde realizar los exámenes de forma presencial.

Entorno tecnológico

Java es un lenguaje de programación de propósito general orientado a objetos desarrollado por Sun Microsystems. También se puede decir que Java es una tecnología que no sólo se reduce al lenguaje, sino que además provee de una máquina virtual Java que permite ejecutar código compilado Java, sea cual sea la plataforma que exista por debajo; plataforma tanto hardware, como software (el sistema operativo que soporte ese hardware). El apoyo a esta tecnología viene dado por la gran cantidad de fabricantes que apoyan esta especificación de máquina virtual.

El lenguaje Java puede describirse de la siguiente manera:

- Simple
- Orientado a Objetos
- Tipado estáticamente
- Distribuido
- Interpretado
- Robusto
- Seguro
- Multihilo
- Con Recolector de basura (Garbage Collector)
- Portable

Por tener todas estas características, se decide utilizar en el marco tecnológico para el desarrollo del software el lenguaje Java. Otro motivo es que en primera instancia el acceso de los gestores de RRHH lo harán de forma de acceso de red local, mientras que los empleados deberán de acceder

de una forma remota ya que encuentran físicamente en el propio cliente, este acceso distribuido se realizará con la tecnología RMI de java.

RMI representa el modelo de objetos distribuidos que propone Java como solución para desarrollar aplicaciones distribuidas. Cuando se desarrolló RMI, se planteó como una solución para entornos exclusivos Java con la finalidad última de simplificar el desarrollo de aplicaciones distribuidas orientadas a objetos y escritas en Java. Se quería que RMI proporcionara los mecanismos necesarios para hacer transparente la complejidad de la comunicación en red.

Las características que definen RMI son las siguientes:

- Transparencia: permite la invocación de métodos de objetos remotos que se ejecutan en otras JVM como procesos separados, con independencia de que estos objetos estén en la misma máquina o en máquinas diferentes.
- Orientado a objetos: integra el modelo de objetos distribuidos en el lenguaje Java de manera natural y preservando, siempre que sea posible, la semántica de objetos de Java.
- Simple: aplica a la programación distribuida los mismos criterios de simplicidad de la programación Java estándar.
- Seguridad: usa un modelo de seguridad basado en los gestores de seguridad y cargadores de clases locales.
- Portabilidad: hace que cualquier sistema distribuido RMI sea portable a cualquier plataforma donde haya una JVM.
- Escalabilidad: mantiene la misma arquitectura con independencia del tamaño de la aplicación distribuida.
- Robustez: proporciona una jerarquía de gestión de excepciones remotas para detectar y gestionar los errores generados en el servidor.
- Eficiencia: permite la activación, desactivación y reactivación, así como la liberación automática de objetos remotos no referenciados, lo que mejoraría la gestión de recursos del servidor.

Además de RMI para poder realizar la aplicación deberemos usar el siguiente software:

- JDK 7 como plataforma base de desarrollo, con lo que se consigue que la aplicación sea multiplataforma, con el único requisito de tener la JVM instalada.
- Eclipse como entorno de programación.
- MagicDraw _17_0 como editor UML.
- PostgreSQL v.9.2 como gestor de bases de datos relacional.
- Herramienta de edición de textos Microsoft Office Word.
- Herramienta de edición de proyectos Microsoft Project.

4. Organización del proyecto

La gestión de proyectos es una disciplina que se lleva estudiando desde los años cincuenta y que ha dado lugar a toda una serie de métodos que nos dicen cómo llevar a cabo un proyecto. Para poder declinarnos por un determinado método debemos observar el contexto del proyecto que queremos llevar a cabo entre los factores que podemos a tener en cuenta para elegir el tipo de método estarán (Wysocki, 2009): el riesgo, valor de negocio, duración, complejidad, tecnología utilizada, número de departamentos afectados, coste.

La clasificación de los mismos sería según el objetivo que estemos que cubrir está claro o no si conocemos los detalles de cómo será la solución o no.

Tenemos 4 grupos diferenciados (Wysocki, 2009):

	Solución conocida	Solución poco conocida
Objetivo claro	1	2
Objetivo poco claro	3	4

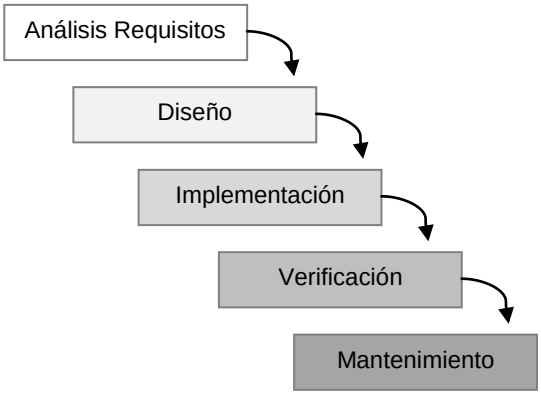
Nosotros usaremos una metodología de tipo 1 en la cual tenemos claro el objetivo y conocemos todos los detalles de la solución, esta metodología se denomina ciclo de vida en cascada.

Ciclo de vida clásico o en cascada

El ciclo de vida en cascada es fácil de aplicar pero es poco tolerante a cambios. Esta metodología tiene una gran similitud con una cadena de producción donde cada trabajador está especializado en realizar una mini tarea de toda la cadena.

El producto software se produce con esta metodología pasa por las etapas siguientes:

1. Requisitos: Los requisitos definen las funciones del sistema software o sus componentes.
2. Análisis y Diseño: En esta etapa se determina cómo debe ser el producto tanto desde el punto de vista externo como interno. El análisis da un punto de vista externo documentando mediante modelos sobre qué hace el sistema, mientras que el diseño da el punto de vista interno documentando cómo obra (qué componentes forman parte de él, cómo se relacionan entre ellos, etc.).
3. Implementación: Se escribe el código, los manuales y se genera el producto ejecutable. Este código se debe escribir según las indicaciones efectuadas en la fase de análisis y diseño.
4. Pruebas: Se verifica que el producto desarrollado se corresponda con los requisitos. En este punto se muestra el producto a los usuarios finales para que validen el resultado.
5. Mantenimiento: Se pone el sistema a disposición de todos los usuarios y se corrigen los defectos que se vayan encontrando.



Ciclo de vida en cascada.

Para mejorar la tolerancia a los cambios y la adaptabilidad de estos métodos, se pueden establecer ciclos de retroalimentación en el análisis final de cada etapa.

Para el desarrollo de este software se no tendremos en cuenta la etapa de mantenimiento.

Descomposición en actividades (WBS)

La estructura de descomposición del trabajo (Work Breakdown Structure) o WBS es una presentación simple del trabajo requerido para realizar un proyecto.

Por tanto no es más que división jerárquica del trabajo y los entregables del proyecto en componentes de menor tamaño, creando lo que se denomina paquetes de trabajo (work packages), pudiendo colocar y controlar dichos paquetes dentro de un calendario, facilitando la planificación y organización de los trabajos que se van realizando.

A partir de las fases del ciclo de vida en cascada que vamos a aplicar nuestro proyecto y teniendo en cuenta las entregas de los PEC's se propone el siguiente WBS de tres niveles:

1	**Fase 1: Planificación**
2	Propuesta
3	Aceptación de propuesta
4	Introducción objetivos del proyecto
5	Metodología y estudio de tareas
6	Planificación temporal (Gant)
7	Elaboración documentación
8	Entrega PEC1
9	**Fase 2: Analisis de requisitos**
10	Identificación de actores
11	Especificación requisitos por subsistema
12	Especificación caso de uso
13	Diagrama de clases gestoras, entidades y jerarquia
14	Requisitos de interfaz de usuario
15	Entrega PEC2
16	**Fase 3: Diseño técnico e implementación**
17	Clases del sistema
18	Diseño arquitectura de componentes
19	Diseño persistencia
20	Implementación
21	Elaboración de documentación
22	PEC3
23	**Fase 4: Memoria y presentación**
24	Memoria
25	Elaboración de presentación
26	**Fase 5: Tribunal**
27	Inicio de preguntas
28	Preguntas
29	Entrega respuestas

Planificación del proyecto

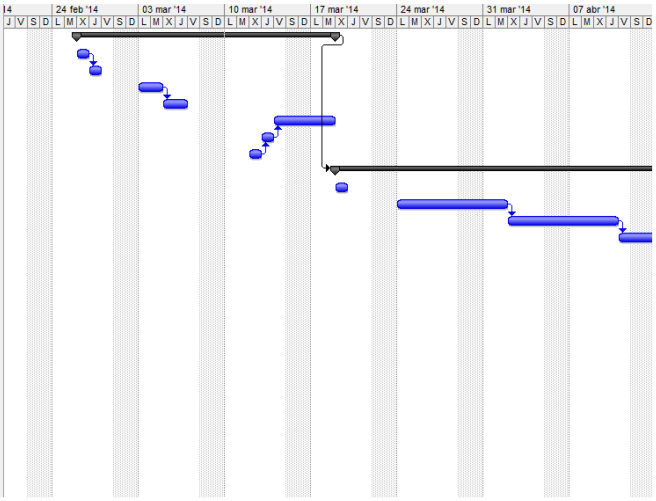
El objetivo de la planificación es obtener la distribución de actividades en el tiempo que garantice las condiciones de entregables exigidos.

Diagrama de Gantt

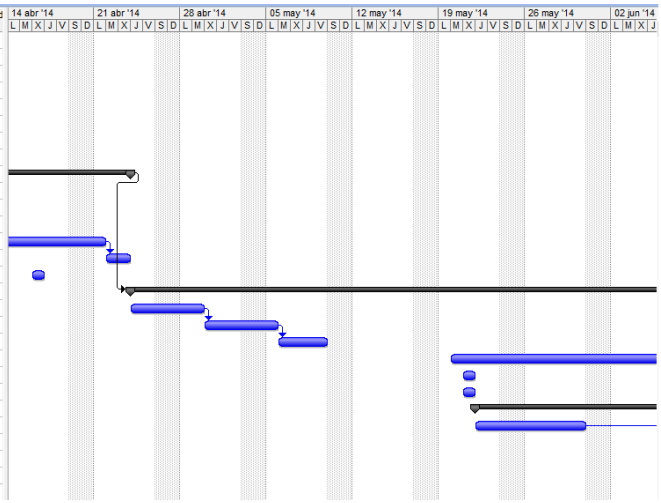
El diagrama de Gantt es una herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

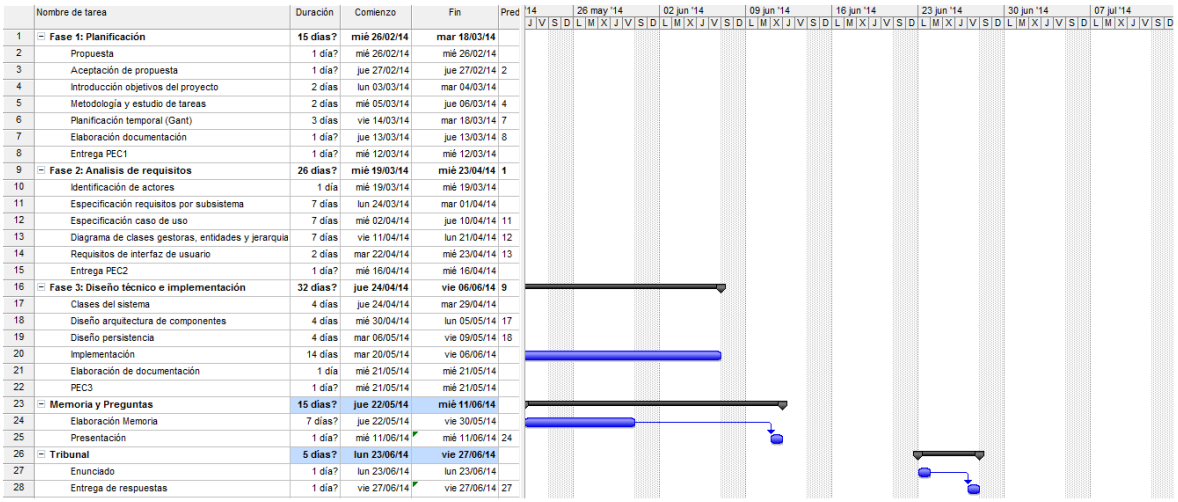
En las siguientes figuras se muestra por un lado la planificación de las tareas y por otro el gráfico Gantt con la planificación y duración de las tareas.

	Nombre de tarea	Duración	Comienzo	Fin	Pred
1	**Fase 1: Planificación**	**15 días?**	**mié 26/02/14**	**mar 18/03/14**	
2	Propuesta	1 día?	mié 26/02/14	mié 26/02/14	
3	Aceptación de propuesta	1 día?	jue 27/02/14	jue 27/02/14	2
4	Introducción objetivos del proyecto	2 días	lun 03/03/14	mar 04/03/14	
5	Metodología y estudio de tareas	2 días	mié 05/03/14	jue 06/03/14	4
6	Planificación temporal (Gant)	3 días	vie 14/03/14	mar 18/03/14	7
7	Elaboración documentación	1 día?	jue 13/03/14	jue 13/03/14	8
8	Entrega PEC1	1 día?	mié 12/03/14	mié 12/03/14	
9	**Fase 2: Analisis de requisitos**	**26 días?**	**mié 19/03/14**	**mié 23/04/14**	**1**
10	Identificación de actores	1 día	mié 19/03/14	mié 19/03/14	
11	Especificación requisitos por subsistema	7 días	lun 24/03/14	mar 01/04/14	
12	Especificación caso de uso	7 días	mié 02/04/14	jue 10/04/14	11
13	Diagrama de clases gestoras, entidades y jerarquia	7 días	vie 11/04/14	lun 21/04/14	12
14	Requisitos de interfaz de usuario	2 días	mar 22/04/14	mié 23/04/14	13
15	Entrega PEC2	1 día?	mié 16/04/14	mié 16/04/14	
16	**Fase 3: Diseño técnico e implementación**	**32 días?**	**jue 24/04/14**	**vie 06/06/14**	**9**
17	Clases del sistema	4 días	jue 24/04/14	mar 29/04/14	
18	Diseño arquitectura de componentes	4 días	mié 30/04/14	lun 05/05/14	17
19	Diseño persistencia	4 días	mar 06/05/14	vie 09/05/14	18
20	Implementación	14 días	mar 20/05/14	vie 06/06/14	
21	Elaboración de documentación	1 día	mié 21/05/14	mié 21/05/14	
22	PEC3	1 día?	mié 21/05/14	mié 21/05/14	
23	**Memoria y Preguntas**	**15 días?**	**jue 22/05/14**	**mié 11/06/14**	
24	Elaboración Memoria	7 días?	jue 22/05/14	vie 30/05/14	
25	Presentación	1 día?	mié 11/06/14	mié 11/06/14	24
26	**Tribunal**	**5 días?**	**lun 23/06/14**	**vie 27/06/14**	
27	Enunciado	1 día?	lun 23/06/14	lun 23/06/14	
28	Entrega de respuestas	1 día?	vie 27/06/14	vie 27/06/14	27



	Nombre de tarea	Duración	Comienzo	Fin	Pred
1	**Fase 1: Planificación**	**15 días?**	**mié 26/02/14**	**mar 18/03/14**	
2	Propuesta	1 día?	mié 26/02/14	mié 26/02/14	
3	Aceptación de propuesta	1 día?	jue 27/02/14	jue 27/02/14	2
4	Introducción objetivos del proyecto	2 días	lun 03/03/14	mar 04/03/14	
5	Metodología y estudio de tareas	2 días	mié 05/03/14	jue 06/03/14	4
6	Planificación temporal (Gant)	3 días	vie 14/03/14	mar 18/03/14	7
7	Elaboración documentación	1 día?	jue 13/03/14	jue 13/03/14	8
8	Entrega PEC1	1 día?	mié 12/03/14	mié 12/03/14	
9	**Fase 2: Analisis de requisitos**	**26 días?**	**mié 19/03/14**	**mié 23/04/14**	**1**
10	Identificación de actores	1 día	mié 19/03/14	mié 19/03/14	
11	Especificación requisitos por subsistema	7 días	lun 24/03/14	mar 01/04/14	
12	Especificación caso de uso	7 días	mié 02/04/14	jue 10/04/14	11
13	Diagrama de clases gestoras, entidades y jerarquia	7 días	vie 11/04/14	lun 21/04/14	12
14	Requisitos de interfaz de usuario	2 días	mar 22/04/14	mié 23/04/14	13
15	Entrega PEC2	1 día?	mié 16/04/14	mié 16/04/14	
16	**Fase 3: Diseño técnico e implementación**	**32 días?**	**jue 24/04/14**	**vie 06/06/14**	**9**
17	Clases del sistema	4 días	jue 24/04/14	mar 29/04/14	
18	Diseño arquitectura de componentes	4 días	mié 30/04/14	lun 05/05/14	17
19	Diseño persistencia	4 días	mar 06/05/14	vie 09/05/14	18
20	Implementación	14 días	mar 20/05/14	vie 06/06/14	
21	Elaboración de documentación	1 día	mié 21/05/14	mié 21/05/14	
22	PEC3	1 día?	mié 21/05/14	mié 21/05/14	
23	**Memoria y Preguntas**	**15 días?**	**jue 22/05/14**	**mié 11/06/14**	
24	Elaboración Memoria	7 días?	jue 22/05/14	vie 30/05/14	
25	Presentación	1 día?	mié 11/06/14	mié 11/06/14	24
26	**Tribunal**	**5 días?**	**lun 23/06/14**	**vie 27/06/14**	
27	Enunciado	1 día?	lun 23/06/14	lun 23/06/14	
28	Entrega de respuestas	1 día?	vie 27/06/14	vie 27/06/14	27





	Nombre de tarea	Duración	Comienzo	Fin	Pred
1	**Fase 1: Planificación**	**15 días?**	**mié 26/02/14**	**mar 18/03/14**	
2	Propuesta	1 día?	mié 26/02/14	mié 26/02/14	
3	Aceptación de propuesta	1 día?	jue 27/02/14	jue 27/02/14	2
4	Introducción objetivos del proyecto	2 días	lun 03/03/14	mar 04/03/14	
5	Metodología y estudio de tareas	2 días	mié 05/03/14	jue 06/03/14	4
6	Planificación temporal (Gant)	3 días	vie 14/03/14	mar 18/03/14	7
7	Elaboración documentación	1 día?	jue 13/03/14	jue 13/03/14	8
8	Entrega PEC1	1 día?	mié 12/03/14	mié 12/03/14	
9	**Fase 2: Analisis de requisitos**	**26 días?**	**mié 19/03/14**	**mié 23/04/14**	**1**
10	Identificación de actores	1 día	mié 19/03/14	mié 19/03/14	
11	Especificación requisitos por subsistema	7 días	lun 24/03/14	mar 01/04/14	
12	Especificación caso de uso	7 días	mié 02/04/14	jue 10/04/14	11
13	Diagrama de clases gestoras, entidades y jerarquia	7 días	vie 11/04/14	lun 21/04/14	12
14	Requisitos de interfaz de usuario	2 días	mar 22/04/14	mié 23/04/14	13
15	Entrega PEC2	1 día?	mié 16/04/14	mié 16/04/14	
16	**Fase 3: Diseño técnico e implementación**	**32 días?**	**jue 24/04/14**	**vie 06/06/14**	**9**
17	Clases del sistema	4 días	jue 24/04/14	mar 29/04/14	
18	Diseño arquitectura de componentes	4 días	mié 30/04/14	lun 05/05/14	17
19	Diseño persistencia	4 días	mar 06/05/14	vie 09/05/14	18
20	Implementación	14 días	mar 20/05/14	vie 06/06/14	
21	Elaboración de documentación	1 día	mié 21/05/14	mié 21/05/14	
22	PEC3	1 día?	mié 21/05/14	mié 21/05/14	
23	**Memoria y Preguntas**	**15 días?**	**jue 22/05/14**	**mié 11/06/14**	
24	Elaboración Memoria	7 días?	jue 22/05/14	vie 30/05/14	
25	Presentación	1 día?	mié 11/06/14	mié 11/06/14	24
26	**Tribunal**	**5 días?**	**lun 23/06/14**	**vie 27/06/14**	
27	Enunciado	1 día?	lun 23/06/14	lun 23/06/14	
28	Entrega de respuestas	1 día?	vie 27/06/14	vie 27/06/14	27

La planificación con fechas es la siguiente:

	Nombre de tarea	Duración	Comienzo	Fin
1	**Fase 1: Planificación**	**15 días?**	**mié 26/02/14**	**mar 18/03/14**
2	Propuesta	1 día?	mié 26/02/14	mié 26/02/14
3	Aceptación de propuesta	1 día?	jue 27/02/14	jue 27/02/14
4	Introducción objetivos del proyecto	2 días	lun 03/03/14	mar 04/03/14
5	Metodología y estudio de tareas	2 días	mié 05/03/14	jue 06/03/14
6	Planificación temporal (Gant)	3 días	vie 14/03/14	mar 18/03/14
7	Elaboración documentación	1 día?	jue 13/03/14	jue 13/03/14
8	Entrega PEC1	1 día?	mié 12/03/14	mié 12/03/14
9	**Fase 2: Analisis de requisitos**	**26 días?**	**mié 19/03/14**	**mié 23/04/14**
10	Identificación de actores	1 día	mié 19/03/14	mié 19/03/14
11	Especificación requisitos por subsistema	7 días	lun 24/03/14	mar 01/04/14
12	Especificación caso de uso	7 días	mié 02/04/14	jue 10/04/14
13	Diagrama de clases gestoras, entidades y jerarquia	7 días	vie 11/04/14	lun 21/04/14
14	Requisitos de interfaz de usuario	2 días	mar 22/04/14	mié 23/04/14
15	Entrega PEC2	1 día?	mié 16/04/14	mié 16/04/14
16	**Fase 3: Diseño técnico e implementación**	**32 días?**	**jue 24/04/14**	**vie 06/06/14**
17	Clases del sistema	4 días	jue 24/04/14	mar 29/04/14
18	Diseño arquitectura de componentes	4 días	mié 30/04/14	lun 05/05/14
19	Diseño persistencia	4 días	mar 06/05/14	vie 09/05/14
20	Implementación	14 días	mar 20/05/14	vie 06/06/14
21	Elaboración de documentación	1 día	mié 21/05/14	mié 21/05/14
22	PEC3	1 día?	mié 21/05/14	mié 21/05/14
23	**Memoria y Preguntas**	**15 días?**	**jue 22/05/14**	**mié 11/06/14**
24	Elaboración Memoria	7 días?	jue 22/05/14	vie 30/05/14
25	Presentación	1 día?	mié 11/06/14	mié 11/06/14
26	**Tribunal**	**5 días?**	**lun 23/06/14**	**vie 27/06/14**
27	Enunciado	1 día?	lun 23/06/14	lun 23/06/14
28	Entrega de respuestas	1 día?	vie 27/06/14	vie 27/06/14

5. Análisis de requisitos

Introducción

De acuerdo con PRESSMAN (2002), el análisis de requisitos es una tarea de ingeniería del software que cubre el hueco entre la definición del software a nivel sistema y el diseño de software. El análisis de requerimientos permite al ingeniero de sistemas especificar las características operacionales del software (función, datos y rendimientos), indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software.

El esta fase, se plasmarán las necesidades y deseos de nuestro cliente, y quedarán todos los requisitos reflejados sin que quede nada que pueda dar lugar a malas interpretaciones o ambigüedades.

El concepto requisito puede definirse como característica observable de un sistema que expresa la necesidad o restricción que un cliente tiene sobre él (Pradel, 2011). La obtención de los mismos puede realizarse según distintas técnicas:

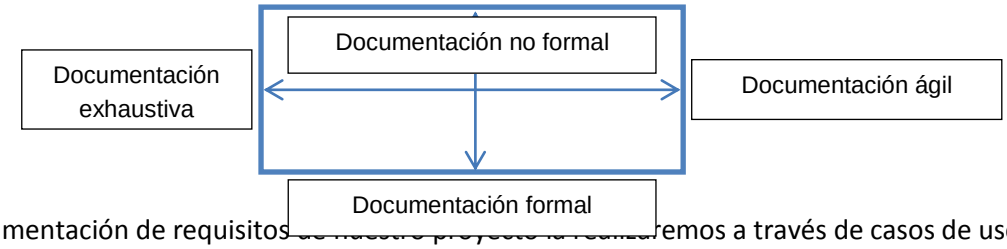
- Lluvia de ideas (Brainstorming).
- Modelado de roles de usuario.
- Entrevistas y cuestionarios.
- Observación y prototipado.
- Listas predefinidas.

En nuestro caso hemos ido realizando distintas entrevistas con el cliente hasta determinar cuál es la problemática existente y perfilar las características que tendrá la solución al mismo.

Especificación de requisitos

La especificación de requisitos recoge el contrato entre los desarrolladores y el cliente y sirve como herramienta de comunicación para los dos grupos. Su estilo y formalidad dependerán del proyecto, pero también del contexto en que se desarrolla y de las personas participantes (Pradel 2011).

La documentación de requisitos de un proyecto puede seguir distintas vertientes según las necesidades que se tengan sobre dicha documentación. En el siguiente gráfico podemos ver la clasificación de las necesidades de la documentación de requisitos:



La documentación de requisitos de nuestro proyecto la realizaremos a través de casos de uso, ya que son muy flexibles por lo que respecta al nivel de formalismo y la exhaustividad de su descripción (Pradel, 2011). Nuestra documentación puede definirse como ágil y con un nivel medio de formalismo.

Dentro de la documentación que vamos a realizar, podemos encontrar distintos tipos de requisitos. Unos podrán ser recogidos dentro de los casos de uso (requisitos funcionales) y otros definirán las restricciones, usabilidad del sistema…

Definición de actores

El actor en un caso de uso, es quien interactúa con el sistema para conseguir un comportacmiento especifico del mismo. El rol que desempeña un actor un rol que puede ser desde un individuo, un dispositivo hardware, otro sistema...

Dentro de nuestro sistema podemos identificar dos actores principales:

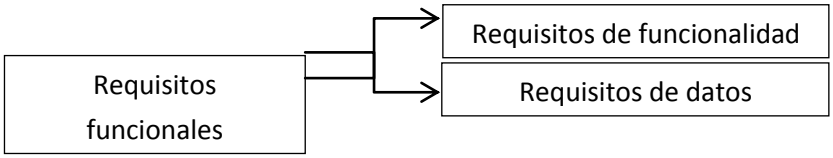
- Gestor de RR.HH: este actor es el que mayor funcionalidad usa del sistema, es el encargado de cargar los datos del currículo de un empleado, mantener actualizado todo lo referente a los mismos y realizar la selección de candidatos para puestos a cubrir en un cliente.
- Empleado: Este actor puede consultar su plan de carrera, ver el estado de sus cursos además de ver a cuales está inscrito, puede entrar en el orientador y seleccionar hacia donde quiere crecer profesionalmente, puede actualizar conocimientos y competencias que vaya adquiriendo por cuenta propia. Conseguimos que nuestro sistema se mantenga activo gracias a las actualizaciones que el empleado va realizando cada cierto tiempo.

Requisitos funcionales

(Pradel, 2011)

Los requisitos funcionales hacen referencia a la funcionalidad que debe proporcionar el sistema y los datos que tiene que conocer y guardar. Nos indican qué cálculos hace el sistema, qué datos mantiene, cómo los manipula, etc.

Podemos distinguir, dentro de los requisitos funcionales, entre los requisitos de funcionalidad y los de datos.



Los requisitos de funcionalidad, en general, describen cuál tiene que ser el comportamiento del sistema explicando qué respuesta debe dar ante los estímulos que le llegan desde el exterior. Por respuesta nos referimos a qué respuesta observable desde fuera, considerando el sistema como una caja negra.

Los requisitos de datos describen qué datos tiene que conocer el sistema. Estos son, típicamente, datos que el sistema guardará de modo persistente.

1. Subsistema curricular

Desde este subsistema se gestiona todo lo referente al currículo de un empleado, alta, modificaciones y bajas.

A este subsistema podrán acceder tanto los empleados para modificar su currículo, pudiendo añadir o eliminar conocimientos y competencias, como los gestores de recursos humanos, los cuales podrán eliminar el currículo y modificar cualquier elemento que lo compone.

En nuevas versiones de software se prevén mejoras como la ordenación de los resultados de la búsqueda de currículo por varios criterios, incluir más filtros en la búsqueda etc.

Alta de currículo

Esta funcionalidad nos permite introducir todos los datos relevantes de un empleado, dichos datos nos permiten tener clasificado al empleado por competencias y conocimientos.

Caso de uso alta de currículo 1.1

Caso de uso: Alta de currículo.

Actor principal: Gestor de recursos humanos.

Caso de uso relacionado: Ninguno.

Precondición: El usuario debe estar logado en el sistema con el rol de gestor.

Postcondición: Se creará un currículo y la aplicación redirigirá al menú por defecto de currículo que es la búsqueda sin parámetros.

Escenario de éxito:

1. El sistema muestra pantalla por defecto del gestor que es la búsqueda de currículo.
2. El usuario selecciona la opción “Alta”.
3. El sistema muestra el formulario donde aparecen los campos de los datos personales del empleado (Figura 1).
4. El usuario introduce el nombre, los apellidos, teléfono, correo electrónico, DNI, perfil profesional, selecciona el estado del empleado guarda y pulsa “siguiente”.
5. El sistema guarda los datos del empleado y nos muestra el formulario de experiencia profesional (Figura 2).
6. El usuario rellena los campos categoría, empresa, fecha de comienzo, fecha de finalización, descripción del puesto, guarda y pulsa “siguiente”.
7. El sistema guarda los datos de la experiencia profesional del empleado y nos muestra el formulario de formación académica (Figura 3).
8. El usuario rellena la titulación, la fecha de comienzo, la fecha de finalización, guarda y pulsa “siguiente”.
9. El sistema guarda la formación académica y nos muestra el formulario de conocimientos (Figura 4).
10. El usuario selecciona la tecnología guarda y pulsa “siguiente”.
11. El sistema guarda el conocimiento seleccionado y muestra el formulario de competencias (Figura 5).
12. El usuario selecciona una competencia guarda y “finaliza”.
13. El sistema guarda la competencia y muestra el menú por defecto del gestor que es la búsqueda de currículo.

Escenarios alternativos:

4ª. El usuario rellena todos los campos.

4ª.1. El sistema le muestra un mensaje de error.

6ª. El usuario rellena los campos categoría, empresa, fecha de comienzo, fecha de finalización, descripción del puesto, guarda y pulsa “añadir experiencia”.

6ª.1. El sistema guarda la experiencia y volvería a mostrar el formulario de experiencia profesional con lo que volveríamos al punto 6.

6b. El usuario no rellena todos los campos.

6b.2. El sistema muestra el mensaje de error.

8ª. El usuario rellena la titulación, la fecha de comienzo, la fecha de finalización, guarda y pulsa “añadir formación”.

8ª.1. El sistema guarda la formación académica y vuelve a mostrar el formulario del punto 8.

8b. El usuario no rellena todos los campos.

8b.1. El sistema muestra un mensaje de error.

10ª. El usuario selecciona la tecnología guarda y pulsa “añadir conocimiento”.

10ª.1. El sistema guarda la tecnología seleccionada y muestra la pantalla del punto 10.

10b. El usuario deja sin seleccionar una tecnología.

10b.1. El sistema muestra un mensaje de error.

10c. El usuario selecciona una tecnología que ya tiene y pulsa “guardar”.

10c.1. El sistema muestra un mensaje de error.

12. El usuario selecciona una competencia guarda y pulsa “añadir competencia”.

12ª. El sistema guarda la competencia y muestra de nuevo el formulario del punto 12.

Datos

En el formulario de formación académica nos encontramos con datos como la fechas comienzo y fin deberá de tener un formato DD/MM/YYYY.

Interfaz gráfica de alta de currículo

La interfaz gráfica de alta de currículo, nos muestra una serie de pantallas a través de las cuales el usuario podrá introducir los datos de cada una de las áreas del currículo de un empleado.

Figura 0.

Figura 1.

Figura 2.

Figura 3.

Figura 4.

Figura 5.

Figura5a

Búsqueda de currículo

Para poder realizar la búsqueda del currículo de un determinado empleado, nos encontramos con que esta funcionalidad queda recogida en el submenú "Búsqueda", aquí el gestor de recursos humanos podrá introducir los filtros para realizar dicha búsqueda y obtendrá como resultado los empleados que cumplen los criterios de filtrado. Sólo serán visibles los empleados que se encuentren en estado activo.

Caso de uso búsqueda de currículo 1.2

Caso de uso: Búsqueda de currículo.

Actor principal: Gestor de recursos humanos.

Caso de uso relacionado: Ninguno.

Precondición: El usuario debe estar logado en el sistema con el rol de gestor.

Postcondición: Se mostrarán los resultados de la búsqueda que se realice según los parámetros introducidos.

Escenario de éxito:

1. El sistema muestra pantalla por defecto del gestor que es la búsqueda de currículo (Figura 6).
2. El usuario rellena los campos nombre, apellidos, DNI y pulsa guardar.
3. El sistema muestra el resultado de la búsqueda.

Escenarios alternativos:

2ª. El usuario no rellena ningún campo.

2ª.1. El sistema le muestra un mensaje de error "Introduzca parámetros de búsqueda".

Extensiones:

3ª. El usuario pulsa el botón de modificar.

3ª.1 El sistema muestra la pantalla de modificación de currículo.

3b. El usuario pulsa el botón de eliminar.

3b.1 El sistema muestra la pantalla de eliminación del currículo.

Datos

El campo DNI no puede tener más de 8 dígitos y debe ir con letra.

Interfaz gráfica de búsqueda de currículo

Figura 6.

Eliminación de currículo

Esta funcionalidad nos permite eliminar creando una baja física del currículo de un empleado borrándolo definitivamente de la base de datos. El gestor de recursos humanos antes de realizar la baja definitiva el sistema le pedirá confirmación. Esta funcionalidad es fundamental para la optimización de sistema, las búsquedas de candidatos serán más rápidas.

Caso de uso eliminar el currículo 1.3

Caso de uso: Eliminar currículo.

Actor principal: Gestor de recursos humanos.

Caso de uso relacionado: Búsqueda de currículo 1.2.

Precondición: El usuario debe estar logado en el sistema con el rol de gestor.

Postcondición: Se podrá inactivo al usuario no saliendo en las búsquedas.

Escenario de éxito:

1.El usuario pulsa eliminar.

2.El sistema muestra la pantalla con los datos del usuario a desactivar.Fig6a.

3.El usuario pulsa eliminar.

4. El sistema desactiva al usuario desapareciendo de las búsquedas de candidatos.

5.El usuario pulsa la tecla de finalizar y vuelve a la búsqueda.

Interfaz gráfica eliminar el currículo

Figura 6ª.

Modificar Currículo

Esta funcionalidad permite tanto al gestor como al empleado mantener actualizado el currículo. El gestor podrá corregir posibles datos mal introducidos, añadir ó eliminar la información que crea relevante para las futuras búsquedas de candidatos. El empleado podrá añadir ó eliminar competencias y conocimientos adquiridos en los nuevos proyectos. Para ello se muestra el currículo con todas las áreas modificables del currículo y desde aquí el usuario puede decidir qué área desea modificar.

Caso de uso modificar el currículo 1.4

Caso de uso: Modificar currículo.

Actor principal: Gestor de recursos humanos.

Caso de uso relacionado: Búsqueda de currículo 1.2.

Precondición: El usuario debe estar logado en el sistema con el rol de gestor.

Postcondición: Se mostrará el currículo del empleado con cada uno de los apartados que se pueden modificar.

Escenario de éxito:

1. El sistema muestra pantalla del currículum vitae (Figura 7).

2. El usuario pulsa modificar alguna de las áreas que tiene el currículo.

3. El sistema muestra la pantalla de modificación del área pulsada.

Extensiones:

2ª. El usuario pulsa el botón de modificar datos personales.

2ª.1. El sistema le muestra una pantalla de modificar datos personales.

2b. El usuario pulsa el botón de modificar experiencia profesional.

2b.1 El sistema muestra la pantalla de modificar experiencia profesional.

2c. El usuario pulsa el botón de modificar la formación académica.

2c.1. El sistema muestra la pantalla de modificación formación académica.

2d. El usuario pulsa el botón de modificar los conocimientos.

2d.1. El sistema muestra la pantalla modificación de conocimientos.

2e. El usuario pulsa el botón de modificar las competencias.

2e.1. El sistema muestra la pantalla modificación de competencias.

Interfaz gráfica modificación de currículo

Figura 7.

Modificación de datos personales

En esta funcionalidad el usuario podrá modificar los datos personales del empleado al que pertenece el currículo.

Caso de uso modificar datos personales 1.5

Caso de uso: Modificar datos personales

Actor principal: Gestor de recursos humanos.

Caso de uso relacionado: Modificar curriculum 1.4

Precondición: El usuario debe estar logado en el sistema con el rol de gestor.

Postcondición: Se mostrará los datos personales que se van a modificar.

Escenario de éxito:

1. El sistema muestra pantalla de los datos personales del empleado seleccionado de la búsqueda (Figura 8).
2. El usuario cambia el contenido de cualquiera de los campos mostrados y pulsa guardar.
3. El sistema guarda los cambios y muestra el botón de finalizar.

Escenario alternativo:

2ª. El usuario deja algún campo vacío.

2ª.1. El sistema le muestra un mensaje de error.

Datos

El campo estado no es editable es un listado de estados, los cuales son activo, baja maternal, baja enfermedad.

Interfaz gráfica modificar datos personales

Figura 8.

Modificación de experiencia profesional

Esta funcionalidad nos permite modificar datos de la experiencia profesional de un empleado.

Caso de uso de experiencia profesional 1.6

Caso de uso: Modificar datos experiencia profesional.

Actor principal: Gestor de recursos humanos.

Caso de uso relacionado: Modificar curriculum 1.4

Precondición: El usuario debe estar logado en el sistema con el rol de gestor.

Postcondición: Se guardarán los datos modificados de la experiencia profesional.

Escenario de éxito:

1. El sistema muestra pantalla de modificación de la experiencia profesional del empleado (Figura 9).

2. El usuario cambia el contenido de cualquiera de los campos mostrados pulsa guardar y finalizar.

Escenario alternativo:

2ª. El usuario deja algún campo vacío.

2ª.1. El sistema le muestra un mensaje de error.

Interfaz gráfico experiencia profesional

Figura 9.

Modificación de formación académica

Esta funcionalidad nos permite seleccionar la acción que queremos llevar a cabo (añadir, eliminar y modificar) sobre la formación académica en el currículo.

Caso de uso modificación formación académica 1.7

Caso de uso: Modificar datos formación académica.

Actor principal: Gestor de recursos humanos.

Caso de uso relacionado: Modificar curriculum 1.4

Precondición: El usuario debe estar logado en el sistema con el rol de gestor.

Postcondición: Se guardará los datos de la formación académica que se han de modificado.

Escenario de éxito:

1. El sistema muestra pantalla de modificación de la formación académica (Figura 10).
2. El usuario cambia el contenido del campo mostrado y pulsa guardar.
3. El sistema guarda la información y muestra el botón de finalizar.

Interfaz gráfico modificación formación académica

Figura 10.

Eliminar formación académica

Esta funcionalidad nos va a permitir eliminar formación académica de un empleado dentro de su currículo, esto nos permitirá corregir posibles errores cometidos al introducir los datos del currículo del empleado.

Caso de uso eliminar formación académica 1.8

Caso de uso: Eliminar formación académica.

Actor principal: Gestor de recursos humanos.

Caso de uso relacionado: Modificar datos formación académica. 1.7.

Precondición: El usuario debe estar logado en el sistema con el rol de gestor.

Postcondición: Se eliminará del currículo los datos referentes a la fomación académica.

Escenario de éxito:

1. El sistema muestra pantalla de eliminación de la formación académica (Figura 11).
2. El usuario pulsa eliminar.
3. El sistema elimina la información y se muestra el botón de finalizar.

Interfaz eliminar formación académica

Figura 11.

Añadir formación académica

La funcionalidad de añadir formación nos permite introducir nueva información sobre titulaciones conseguidas en el campo académico.

Caso uso añadir formación académica 1.9

Caso de uso: Añadir formación académica.

Actor principal: Gestor de recursos humanos.

Caso de uso relacionado: Modificar datos formación académica 1.7.

Precondición: El usuario debe estar logado en el sistema con el rol de gestor.

Postcondición: Se añadirá la formación académica al curriculum.

Escenario de éxito:

1. El sistema muestra pantalla de añadir formación académica (Figura 12).
2. El usuario añade la información de la formación académica y pulsa añadir.
3. El sistema guarda la información y muestra botón de finalizar.

Interfaz añadir formación académica

Figura 12

Eliminar conocimientos.

Esta funcionalidad elimina los conocimientos técnicos de un empleado de su currículo.

Caso de uso eliminar conocimientos 1.11

Caso de uso: Eliminar conocimientos.

Actor principal: Gestor de recursos humanos, empleado.

Caso de uso relacionado: Modificar curriculum 1.4.

Precondición: El usuario debe estar logado en el sistema con el rol de gestor, o de empleado.

Postcondición: Se borraran los conocimientos del curriculum.

Escenario de éxito:

1. El sistema muestra pantalla de eliminar de conocimientos (Figura14).
2. El usuario marca los conocimientos que quiere borrar y pulsa eliminar.
3. El sistema borra el conocimiento y muestra la pantalla de búsqueda de nuevo.

Interfaz gráfico eliminar conocimientos

Figura 14

Añadir conocimientos.

Esta funcionalidad nos permite añadir conocimientos técnicos al currículo del empleado.

Caso de uso añadir conocimientos 1.12

Caso de uso: Añadir conocimientos.

Actor principal: Gestor de recursos humanos, empleado.

Caso de uso relacionado: Modificar curriculum 1.4

Precondición: El usuario debe estar logado en el sistema con el rol de gestor o de empleado.

Postcondición: Se mostrará los conocimientos técnicos añadidos en la pantalla de currículo.

Escenario de éxito:

1. El sistema muestra pantalla de añadir de conocimientos (Figura 15).
2. El usuario selecciona de la lista el conocimiento que le interesa y pulsa añadir.
3. El sistema añade el conocimiento y muestra la pantalla de búsqueda de nuevo.

Interfaz gráfico añadir conocimientos

Figura 15

Eliminar competencias.

Caso de uso eliminar competencias 1.13

Caso de uso: Eliminar competencias.

Actor principal: Gestor de recursos humanos, empleado.

Caso de uso relacionado: Modificar curriculum 1.4

Precondición: El usuario debe estar logado en el sistema con el rol de gestor o de empleado.

Postcondición: Se eliminara la competencia seleccionada en el combo

Escenario de éxito:

1. El sistema muestra pantalla de eliminar de competencias (Figura 16).
2. El usuario selecciona de la lista la competencia que le interesa y pulsa borrar.
3. El sistema borra la competencia y muestra la pantalla de búsqueda de CV.

Interfaz gráfico eliminar competencia

Figura 17.

Añadir competencias

Esta funcionalidad nos permite añadir competencias al currículo de un empleado.

Caso de uso añadir competencias 1.14

Caso de uso: Añadir competencia.

Actor principal: Gestor de recursos humanos, empleado.

Caso de uso relacionado: Modificar curriculum 1.4

Precondición: El usuario debe estar logado en el sistema con el rol de gestor.

Postcondición: Se mostrará las competencias restantes en la pantalla de currículo.

Escenario de éxito:

1. El sistema muestra pantalla de añadir de competencias (Figura 18).
2. El usuario selecciona de la lista de competencias la que le interesa y pulsa añadir.
3. El sistema añade la competencia y muestra la búsqueda de nuevo.

Interfaz gráfico añadir competencia

Figura 18.

2. Subsistema Orientador.

En el subsistema orientador se ofrece la funcionalidad que permite a un empleado, conocer cuáles son las distintas trayectorias que podría seguir dentro de la compañía, entender de que se encarga cada categoría y así poder elegir de forma determinante cual es la categoría profesional a la que quiere en un futuro dentro de la empresa.

Caso de uso seleccionar categoría profesional 2.1

Caso de uso: Seleccionar categoría profesional.

Actor principal: Empleado.

Caso de uso relacionado: Ninguno.

Precondición: El usuario debe estar logado en el sistema con el rol de empleado.

Postcondición: Se quedará registrada la categoría que seleccionada en su plan de carrera.

Escenario de éxito:

1. El sistema muestra pantalla de orientador (Figura 19).
2. El usuario selecciona del listado de categorías una de ellas.
3. El sistema inscribe al usuario en los cursos que se corresponde a la categoria seleccionada y aparecerán en su plan de carrera y la marca como seleccionada y muestra la pantalla de curriculum que es la pantalla por defecto.

Interfaz gráfica del orientador

Bienvenido **aurora.jimenez** Cerrar sesion

Find your way

Curriculo Orientador Plan de Carrera

Sistemas

Nombre	Descripcion	Accion
Tecnico Microinformatico	El tecnico microinformatico su labor es el mantenimiento de los equipos y programas informáticos, debe poseer una visión global del estado de los sistemas informáticos para resolver las incidencias que surjan	Seleccion
Técnico de Sistemas	Es el encargado del mantenimiento y funcionamiento de todo el equipamiento informático. Ha de tener un amplio conocimiento del conjunto de la red instalada, su arquitectura y localización..	Seleccion
Administrador de Sistemas	El administrador de sistemas es la persona que tiene la responsabilidad de diseñar, implementar, ejecutar y asegurar el correcto funcionamiento de un sistema informático, o algún aspecto de éste..	Seleccion
Administrador de base de datos	El administrador de bases de datos (DBA1) es el profesional que administra las tecnologias de la información y la comunicación, siendo responsable de los aspectos técnicos, tecnológicos, cientificos, inteligencia de negocios y legales de bases de datos.	Seleccion
Responsable de Sistemas	El responsable del sistema se centra en la concepción y el diseño de la toda la infraestructura informática, tanto en el entorno de red local como de comunicaciones, la definición de las politicas de seguridad y de control de acceso a los recursos.	Seleccion

Programacion

Nombre	Descripcion	Accion
Programador	El programador es el responsable de escribir el código fuente a partir del cual se han de generar los programas.	Seleccion
Analista Programador	El analista-programador le corresponde,analisis e implementación d eaquellos que por su estudio,onfección o tratamiento revistan mayor complejidad.	Seleccion
Analista Organico	El analista orgánico se encarga del diseño detallado del sistema respetando la arquitectura definida por el arquitecto.	Seleccion
Analista Funcional	El analista funcional es el responsable de unificar las diferentes visiones del dominio en un único modelo que sea claro, conciso y consistente.	Seleccion
Jefe de Proyecto	El jefe de proyectos es la persona responsable de organizar el proyecto conseguir los objetivos del mismo y coordinar la relación entre el equipo de desarrollo y el resto de la organización .	Seleccion

Figura 19.

3. Subsistema Búsqueda de candidato

El subsistema búsqueda permite buscar candidatos según tres parámetros: perfil, competencias y conocimientos.

Caso de uso búsqueda de candidato 3.1

Caso de uso: Búsqueda de candidato.

Actor principal: Gestor de recursos humanos.

Caso de uso relacionado: Ninguno.

Precondición: El usuario debe estar logado en el sistema como gestor de recursos humanos.

Postcondición: Se mostrará el resultado de la búsqueda por los tres criterios, además se mostrará otro resultado de una segunda búsqueda donde los filtros son por uno u otro criterio dando como resultado los posibles aspirantes al puesto.

Escenario de éxito:

1. El sistema muestra pantalla de búsqueda (Figura 20).
2. El usuario rellena los campos de búsqueda y pulsa buscar.
3. El sistema muestra un listado con los candidatos que cumplen los criterios de búsqueda y un segundo listado que contiene los posibles aspirantes.

Extensiones:

3ª. El usuario pulsa plan de carrera del aspirante.

2ª.1. El sistema muestra la pantalla del plan de carrera.

Interfaz gráfica búsqueda de candidato.

Figura 20.

4. Subsistema Plan carrera.

El subsistema plan de carrera muestra la funcionalidad de visualizar, inscribir y marcar como completado los cursos. El gestor de recursos humanos después de seleccionar un aspirante de la búsqueda de candidatos, podrá en su plan de carrera inscribir al aspirante a los cursos que la plataforma le ofrece. Desde la misma pantalla el gestor podrá completar los cursos que el empleado que ya ha recibido de forma presencial en las aulas habilitadas para ello. Un empleado podrá consultar los cursos a los que está inscrito y el detalle de los mismos.

Caso de uso plan carrera 4.1

Caso de uso: Plan de carrera.

Actor principal: Gestor de recursos humanos.

Caso de uso relacionado: 3.

Precondición: El usuario debe estar logado en el sistema como gestor de recursos humanos.

Postcondición: Se mostrará un listado con todos los cursos que tiene la persona en su plan de carrera, tanto inscritos como completados.

Escenario de éxito:

1. El sistema muestra la pantalla del plan de carrera (Figura 20 y 21).
2. El usuario pulsa inscribir a los cursos que se deben añadir al plan de carrera, de esta forma el aspirante podrá pasar a candidato cuando los complete.
3. El sistema inscribe al curso al empleado y vuelve a mostrar la de búsqueda de candidatos.

Escenarios alternativos:

2ª. El usuario pincha en completar.

2ª.1. El sistema completa el curso que ha marcado y vuelve a cargar la pantalla de búsqueda de candidatos.

2b. El usuario pulsa volver.

2b.1. El sistema muestra la pantalla de búsqueda de candidatos.

Interfaz gráfica plan carrera.

Figura 21.

Figura 22.

Caso de uso búsqueda plan carrera 4.2

Caso de uso: Búsqueda del plan de carrera.

Actor principal: Gestor de recursos humanos.

Caso de uso relacionado: Ninguno.

Precondición: El usuario debe estar logado en el sistema como gestor de recursos humanos.

Postcondición: Se mostrará un listado con todos los planes de carrera de los empleados que cumplan los filtros de búsqueda.

Escenario de éxito:

1. El sistema muestra la pantalla del plan de carrera (Figura 22).
2. El usuario rellena alguno de los filtros de la búsqueda y pulsa buscar.
3. El sistema muestra un listado con todos los empleados que cumplen una condición específica de la búsqueda.
4. El usuario pulsa en el botón de plan de carrera y se le mostrará el plan de carrera del empleado (Figura 21).

Interfaz gráfica plan carrera.

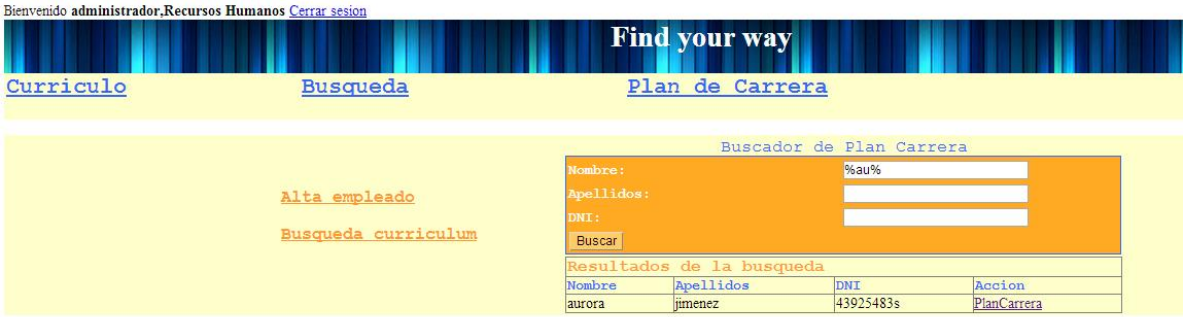


Figura 23.

Subsistema Login

En este subsistema comprueba la existencia de una persona dentro de la aplicación, para ello el usuario para entrar en la aplicación debe logarse con su usuario y su contraseña.

Caso de uso iniciar sesión 5.1

Caso de uso: Iniciar sesión.

Actor principal: Gestor de recursos humanos ó empleado.

Caso de uso relacionado: Ninguno.

Precondición: El usuario debe estar creado previamente en la base de datos

Postcondición: Se la pantalla de inicio del usuario.

Escenario de éxito:

1. El sistema muestra la pantalla de inicio de sesión (Figura 23).
2. El usuario rellena introduce los datos de nombre de **usuario que es su email, password que será "1234" y pulsa "iniciar sesión".**
3. El sistema comprueba la existencia del usuario y muestra la pantalla de home (figura 24 y 25).

Interfaz gráfica inicio de sesión.

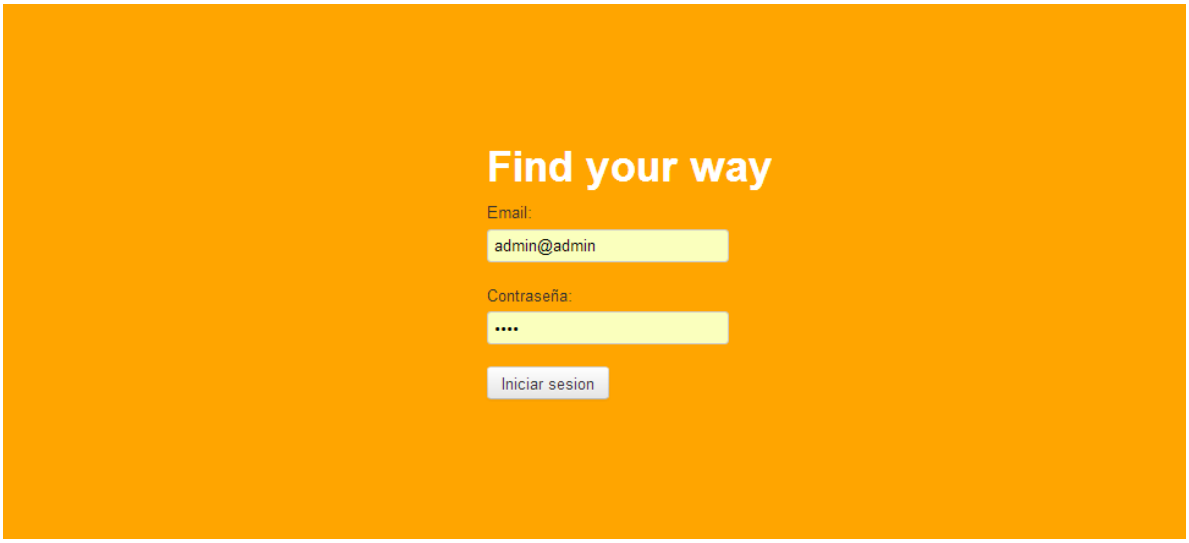


Figura 24.

Figura 25.

Figura 26.

Requisitos no funcionales

Los requisitos no funcionales definen restricciones sobre el sistema a desarrollar, con esto se definen criterios como la usabilidad, fiabilidad, rendimiento o seguridad.

Según Volere los requisitos no funcionales se pueden clasificar en:

De presentación: hacen referencia a la parte visual del sistema

- En la cabecera de la aplicación se mostrará siempre el nombre “Find my way”.
- El diseño de las pantallas debe incluir los colores corporativos.

De usabilidad y humanidad: son los relacionados con un sistema sea ergonómico y usable.

- La aplicación no debe ser intuitiva y de fácil uso

De cumplimiento: Son los requisitos sobre la manera de cumplir las responsabilidades del sistema, incluyendo características como la velocidad, la latencia, la fiabilidad, la robustez.

- La aplicación debe ser distribuida y su acceso on-line.

De mantenimiento y de soporte: Hacen referencia al mantenimiento y soporte a usuarios.

- Los evolutivos o mejoras que se hagan sobre la plataforma siempre se harán respetando la persistencia de los datos.

De seguridad: aquellos sobre los requisitos de acceso, integridad, auditoría, etc.

- El acceso a la plataforma se hará siempre mediante usuario y password.

Legales: son los que hay que satisfacer en virtud de la ley que se aplicará sobre la organización que sea responsable del sistema

- El almacenamiento de los datos cumplirá la LOPD.

6. Diseño técnico

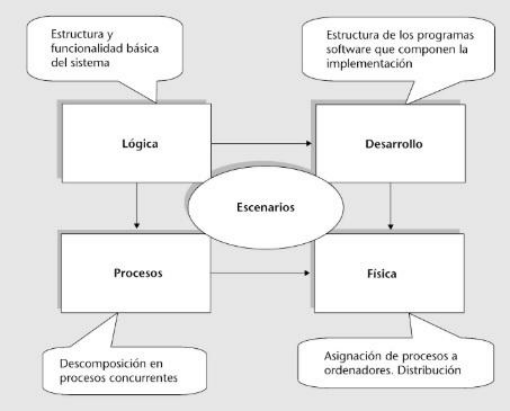
El diseño es el lugar donde los requerimientos y las necesidades de negocio se unen para dar lugar a un producto. El diseño crea un modelo que proporciona detalles sobre la arquitectura del software, estructuras de datos, interfaces y componentes que se necesitan para implementar el sistema.

Los pasos que se siguen suelen ser, primero se representa la arquitectura del sistema. Después se se crean las interfaces que conectan al software con los usuarios, con otros sistemas, y con los propios componentes. Para finalizar se diseñan los componentes de software que componen nuestro sistema.

En el año 2000 se publicó el estándar IEEE 1471 que da las bases para la descripción de la arquitectura software, utilizando una separación modular del diseño en diferentes perspectivas, denominadas puntos de vista.

En el año 2000 el estándar IEEE1471 "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems", "que sienta las bases para la descripción de la arquitectura global de los sistemas software, utilizando una separación modular del diseño en diferentes perspectivas, denominadas puntos de vista.

Entre las distintas propuestas que existen para una especificar la arquitectura de un sistema software hemos elegido el modelo "4+1" vistas de Kruchten. Este modelo define 5 vistas.



(Vacecillo, 2013)

1) La vista lógica corresponde al diseño de la estructura y funcionalidad básica del sistema. En el caso de usar un diseño orientado a objetos, esta vista estaría compuesta por el diagrama de clases, que identifica las entidades principales que componen el sistema, junto con las operaciones básicas que implementa cada uno de los objetos de esas clases.
2) La vista de procesos descompone la funcionalidad del sistema en una serie de procesos o componentes software, que interactúan de forma concurrente. Esta vista captura los aspectos de concurrencia y sincronización del sistema, centrándose por tanto en aspectos como el rendimiento, la integridad y la fiabilidad del sistema.
3) La vista de desarrollo describe cómo están organizados los programas software que implementan el sistema en capas, subsistemas, módulos, etc., así como sus dependencias. De

esta forma pueden capturarse los aspectos de uso, portabilidad y gestión del desarrollo del sistema.

4) La vista física describe cómo se distribuye el software en los diferentes elementos hardware, tanto de procesamiento (ordenadores) como de comunicaciones (LAN, WAN, etc.). Esta vista refleja por tanto los aspectos de despliegue, distribución y replicación, y aborda problemas como la distribución de cargas, el rendimiento global y la escalabilidad.
5) Finalmente, los escenarios sirven para ilustrar los requisitos y funcionalidad principal del sistema, mediante una serie de casos de uso encadenados en lo que Kruchten denomina escenarios (de ahí el nombre de esta vista). Los escenarios describen las secuencias de interacciones entre los objetos y entre los procesos que componen el diseño del sistema, mediante diagramas de casos de uso y de comunicación.

En la propuesta de Kruchten, aquellas vistas que no sean relevantes para un sistema pueden ser omitidas de su descripción arquitectónica.

Modelado clase-responsabilidad-colaborador (CRC)

El modelo CRC en realidad es un conjunto de tarjetas que representan clases. Las tarjetas se dividen en tres secciones. En la parte superior de la tarjeta se escribe el nombre de la clase, en la parte izquierda del cuerpo se enlistan las responsabilidades de la clase y en la derecha los colaboradores.

El modelo CRC hace uso de tarjetas reales o virtuales. El objetivo es desarrollar una representación organizada de las clases. Las responsabilidades son los atributos y operaciones relevantes para la clase. La responsabilidad es cualquier cosa que la clase sepa o haga. Los colaboradores son aquellas clases que se requieren para dar a una clase la información necesaria a fin de completar una responsabilidad. En general, una colaboración implica una solicitud de información o de una cierta acción.

- Clases de entidad, también llamadas clases modelo o de negocio, se extraen directamente del enunciado del problema. Es común que estas clases representen cosas almacenadas en una base de datos y que persistan mientras dure la aplicación.
- Clases de frontera se utilizan para crear la interfaz que el usuario mira y con la que interactúa cuando utiliza el software. Los objetos de entidad contienen información que es importante para los usuarios pero no se muestran por sí mismos. Las clases de frontera se diseñan con la responsabilidad de administrar la forma en la que se presentan a los usuarios los objetos de entidad.
- Clases controlador administran una unidad de trabajo de principio a fin. Es decir, las clases de controlador están diseñadas para administrar 1) la creación o actualización de objetos de entidad, 2) las instancias de los objetos de frontera en tanto obtienen información de los objetos de entidad, 3) la comunicación compleja entre conjuntos de objetos y 4) la validación de datos comunicados entre objetos o entre el usuario y la aplicación. En general, las clases de controlador no se consideran hasta haber comenzado la actividad de diseño.

Las clases entidad detectadas:

CategoriaConocimiento	
Descripción:	Representa los conocimientos que debe tener un empleado con una categoria profesional
Tipo:	Frontera
Características:	Concreta, persistente
Responsabilidades:	Mantiene los datos conocimientos asociados a una categoria
Colaboraciones:	FormaciónAcademica,ExperienciaProfesional, Conocimientos, Competencias
Constructor: +CategoriaConocimiento() + CategoriaConocimiento (Categoria idC, String nombreCto, Conocimiento c)	
Atributos: -idCategoria: Identificador de la categoria -idCategoriaConocimiento: Identificador de la categoría conocimiento. -idConocimientoCurso:Identificador de conocimiento curso.	
Métodos: -getIdCategoria():Categoria -setIdCategoria(Categoria c) -getIdConocimientoCurso():Curriculo -setIdConocimientoCurso(ConocimientoCurso c) -getNombreConocimiento():String -setNombreConocimiento(String c) -getIdCategoriaConocimiento(Long) -setIdCategoriaConocimiento(Long c)	

Curriculo	
Descripción:	Representa el conjunto de datos académicos y profesionales de un empleado.
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Mantiene los datos del currículo de un empleado
Colaboraciones:	FormaciónAcademica,ExperienciaProfesional, Conocimientos, Competencias
Constructor: +Curriculo()	

Curriculo
+Curriculo(Usuario user)
Atributos: -idUsuario: Identificador de usuario -idcurriculo: Identificador único
Métodos: -getIdUsuario():Usuario -setIdUsuario(Usuario user) -getIdcurriculo():Curriculo -setIdcurriculo()

FormacionAcademica	
Descripción:	Representa la formación académica de un empleado
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Almacena y proporciona los datos de la formación académica de un empleado
Colaboraciones:	Currículo
Constructores: +FormacionAcademica() +FormacionAcademica(String fechaInicio, String fechaFin, Curriculo idcv, String titulo)	
Atributos: -idcurriculo: Curriculo //Identificador único del currículo de un empleado -titulo: String //Valor del título académico -idFormaciónAcademica: String // Identificador único de la formación académica. -fechaInicio: String -fechaFin:String	
Métodos: //Métodos de acceso a los atributos privados -getIdFormacionAcademica(): Long -setIdFormacionAcademica (Long idFormacionAcadémica)	

FormacionAcademica
-getIdcurriculo(Curriculo curri) -setIdcurriculo():Curriculo -getTitulo(): String -setTitulo(String tituloAcademica) -getFechaInicio():String -setFechaInicio(String fechaInicio) -getFechaFin():String -setFechaInicio(String fechaFin)

ExperienciaProfesional	
Descripción:	Representa la experiencia profesional de un empleado
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Almacena y proporciona los datos la experiencia profesional de un empleado.
Colaboraciones:	Curriculo
Constructores: +ExperienciaProfesional() +ExperienciaProfesional(String idCategoria, Curriculo idCurriculo, String fechaInicio, String fechaFin, String empresa, String descripcionPuesto)	
Atributos: -idexperienciaprof: String //Identificador del usuario propietario de la búsqueda. -Curriculo: Curriculo//Objeto contenedor de las experiencia. -categoriaProfesional: String //Valor de la categoria profesional. -empresa: String //Nombre de la empresa. -fechaInicio: String //Fecha de inicio del contrato. -fechaFin: String //Fecha de fin del contrato. -descripcionPuesto: String //Descripción de las funciones realizadas durante la relación laboral con la empresa.	
Métodos: //Métodos de acceso a los atributos privados -getIdexperienciaprof(): Long -setIdexperienciaprof (Long idExp) -getIdcurriculo(Curriculo currri)	

ExperienciaProfesional
-setIdcurriculo():Curriculo -getCategoriaProfesional(): String -setCategoriaProfesional(String categoria) -getEmpresa(): String -setEmpresa(String empresa) -getFechaInicio(): String -setFechaInicio(String inicio) -getFechaFin(): String -setFechaFin(String fechaFin) -getFechaFin(): Date -setFechaFin(Date fechaFin)

ConocimientoCurriculum	
Descripción:	Representa los conocimientos de un empleado.
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Almacena y proporciona los datos los conocimientos de un empleado.
Colaboraciones:	Currículo
Constructores: +ConocimientoCurriculum() +ConocimientoCurriculum(Curriculum cv, String nombreConocimiento)	
Atributos: -idConocimiento: Long //Identificador de la tecnología. -idcurriculo: Curriculum//Objeto que contiene la tecnología. -nombreConocimiento: String //nombre tecnología.	
Métodos: //Métodos de acceso a los atributos privados -getIdConocimiento(): Long -setIdConocimiento(Long idConocimiento) -getCurriculo(): Curriculo -setCurriculo(Curriculo curri) -getNombreConocmiento(): String -setNombreConocimiento(String tecnología)	

ConocimientoCurso	
Descripción:	Representa la entidad de una técnología y su curso correspondiente
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Almacena datos curso asociado a un conocimiento.
Colaboraciones:	Curriculo
Constructores: +ConocimientoCurso() +ConocimientoCurso(String nombreConocimento, String nombreCurso)	
Atributos: -idConocimientoCurso: Long //Identificador de la tabla -nombreCtoCurso: String //nombre del conocimiento técnico -nombreCurso: String //Nombre del curso.	
Métodos: //Métodos de acceso a los atributos privados -getIdConocimientoCurso: Long -setIdConocimientoCurso(Long id) -getNombreCtoCurso (): String -setNombreCtoCurso(String cto) -setNombreCurso(String curso) -get nombreCurso ():String	

Competencia	
Descripción:	Representa una competencia del empleado
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Almacena y proporciona los datos de una competencia.
Colaboraciones:	Curriculo
Constructores: +Competencia() +Competencia(Curriculo idcurriculo, String nombreCompetencia)	
Atributos:	

Competencia
-idCompetencia: Long //Identificador de la competencia -idcurriculo: Curriculo //Curriculo al que pertenece la competencia -nombreCompetencia: String //Nombre de la comeptencia
Métodos: //Métodos de acceso a los atributos privados -getIdCompetencia(): Long -setIdCompetencia(Long idUsuario) -getIdcurriculo(): Curriculo -setIdcurriculo(Curriculo c) -setNombreCompetencia (String competencia) -getNombreCompetencia ():String

Usuario	
Descripción:	Define las características fundamentales de un usuario
Tipo:	Principal
Características	Persistente
Responsabilidades:	Modela los datos personales de un usuario.
Colaboraciones:	
Constructores: +Persona()	
Atributos: -dni: String//Identificador único de un usuario -nombre: String //Nombre del usuario -apellidos: String //Apellidos del usuario -telefono: String //Teléfono del usuario -email: String //Email del usuario -estado:String//Estado de la persona -rol:String//Definicion de rol -idUsario:Long -isAdmin:boolean	
Métodos: //Métodos de acceso a los atributos privados -getDni(): String -setDni(String dni) -getNombre(): String	

Usuario
-setNombre(String nombre) -getApellidos(): String -setApellidos(String apellidos) -getTelefono(): String -setTelefono(String tlfno) -getEmail(): String -setEmail(String email) -getStatus():String -setStatus(String status) -getRol():String -setRol(String rol)

PlanCarrera	
Descripción:	Representa el plan de carrera de un empleado
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Mantiene los datos del plan de carrera de un empleado
Colaboraciones:	Persona,Inscripción
Constructores: +PlanCarrera() +PlanCarrera(Inscripcion in)	
Atributos: -id: Long//identificador unico del plan -inscripcionId:Inscripcion//identificador de inscripcion	
Métodos: //Métodos de acceso a los atributos privados -getId():Long -setId(Long id) -setinscripcionId(Inscripcion ins) -getinscripcionId():Inscripcion	

Orientador

Orientador	
Descripción:	Representa la entidad del rol marcado.
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Inscribir al usuario en los cursos que se corresponden a la categoria que seleccionan.
Colaboraciones:	Persona,Cursos
Constructores: +Curso()	
Atributos: -idorientador: Long//identificador unico de orientador -usuarioid:Long//identificador del usuario que te inscribes -nombre:String//nombre conocimiento	
Métodos: //Métodos de acceso a los atributos privados -getIdorientador(): String -setIdorientador(String idCurso) -getNombre():String -setNombre(String nombreCurso) -getUsuarioid():Long -setUsuarioid(Long cc)	

Inscripcion	
Descripción:	Representa la inscripcion de una persona a un curso
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Mantiene los datos de la inscripcion
Colaboraciones:	Persona,Cursos
Constructores: +Inscripcion() +Inscripcion(String fInscripcion,String estado,Usuario idUsr, Conocimiento idconoci)	
Atributos: -estado: String//identificador único del curso	

Inscripcion
-fechaInscripcion:String//fecha de inscripción -id:String//estado de la inscripción -idConocimientoCurso: Persona -idUsuario:Usuario
Métodos: //Métodos de acceso a los atributos privados -getId(): Long -setId(Long id) -getFechaInscripcion():String -setFechaInscripcion(String fecha) -getEstado():String -setEstado(String estado) -geIdUsuario():Usuario -setIdUsuario(Usuario usu) -getIdConocimientoCurso():ConocimientoCurso -setIdConocimientoCurso(ConocimientoCurso c)

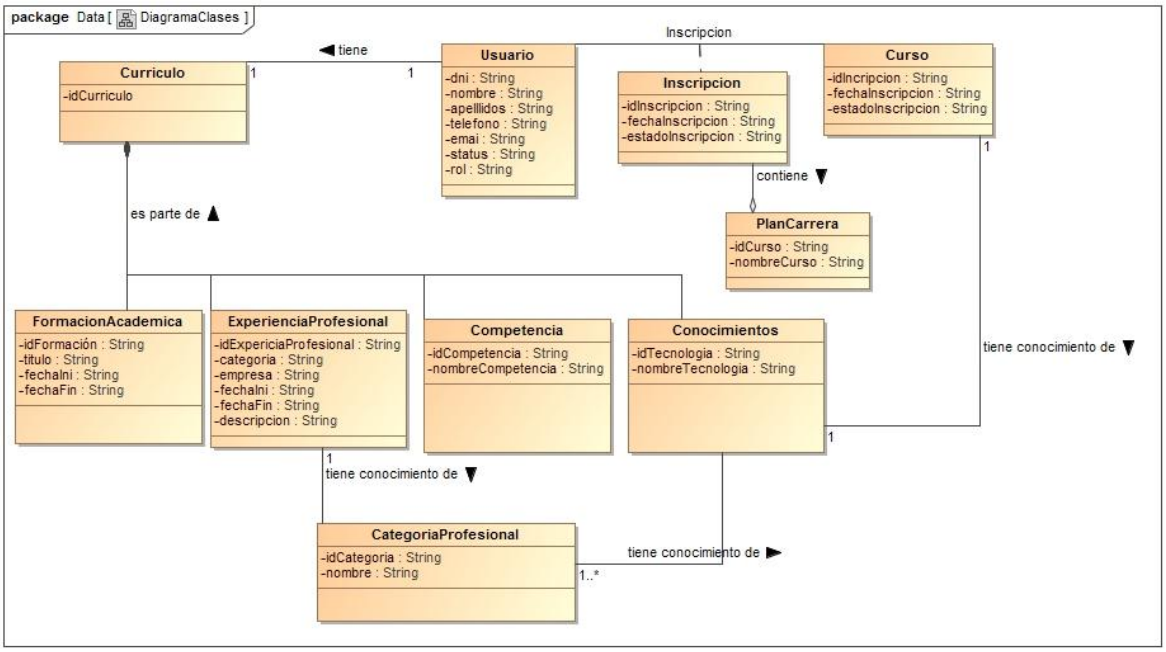
Categoria	
Descripción:	Representa la categoría de una persona
Tipo:	Principal
Características:	Concreta, persistente
Responsabilidades:	Mantiene los datos de una categoria
Colaboraciones:	Conocimientos
Constructores: +Categoria()	
Atributos: -idCategoria:String -nombre: String//identificador único del curso	
Métodos: //Métodos de acceso a los atributos privados -get IdCategoria (): String -set IdCategoria (String categoria) -getNombre():String	

Categoria
-setNombre(String categoria)

Modelo del dominio

Según Pradel[página 31] un modelo del dominio es una representación de clases conceptuales del mundo real en un dominio de interés. Este modelo ha de servir a las personas que intervienen en el desarrollo del software para entender mejor el dominio del sistema.

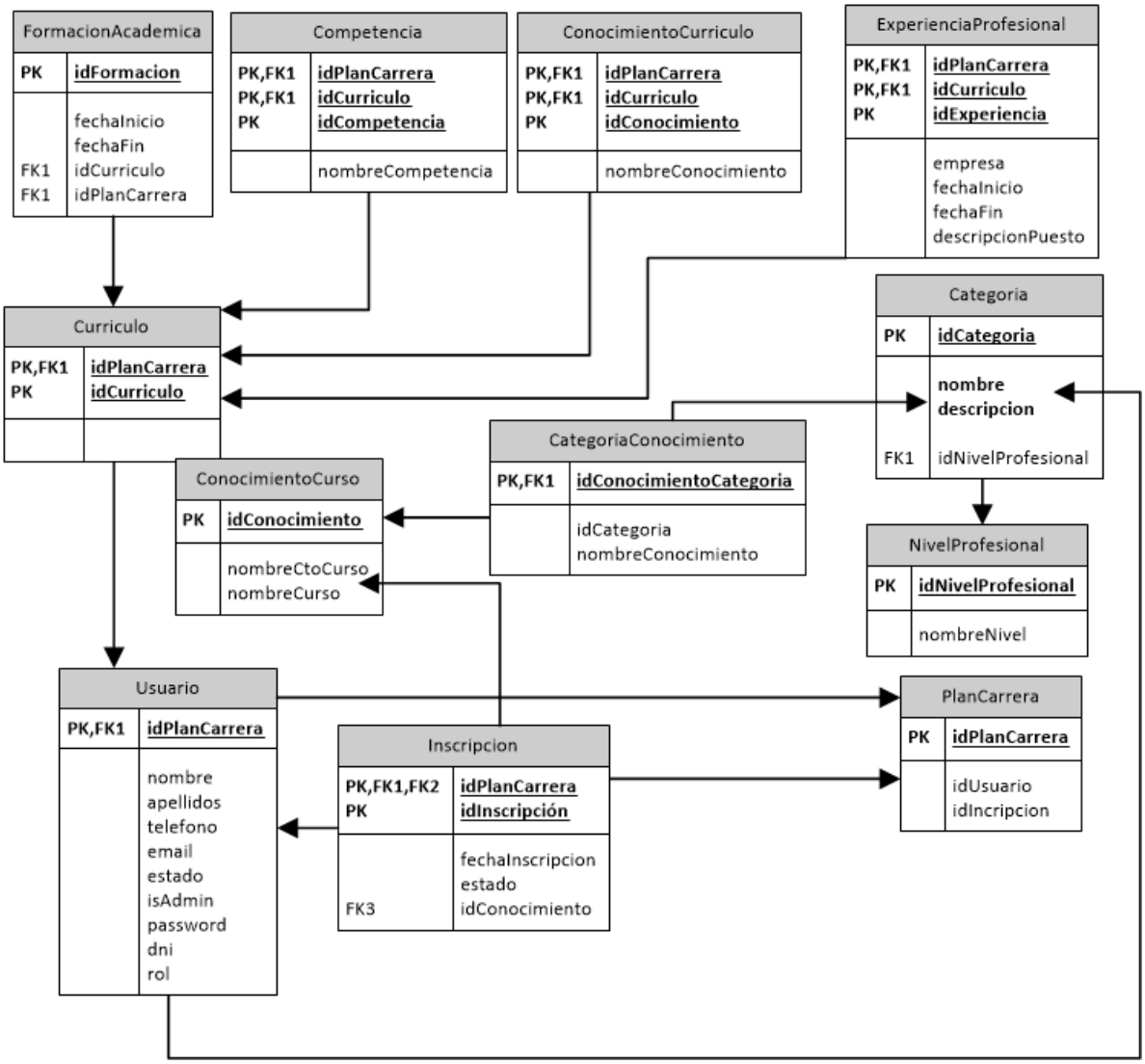
Las clases que representan nuestro dominio y las relaciones entre ellas se representan en el siguiente diagrama:



Como nuestras clases son persistentes y por tanto pueden traducirse en un modelo Entidad Relación.

Modelo entidad-relación

El modelo entidad-relación ER es un modelo de datos que permite representar cualquier abstracción, percepción y conocimiento en un sistema de información formado por un conjunto de objetos denominados entidades y relaciones, incorporando una representación visual conocida como diagrama entidad-relación.



Arquitectura software del sistema

La arquitectura software consiste en un conjunto de decisiones, patrones, abstracciones coherentes que nos sirven de referencia para guiarnos en la construcción del software.

La arquitectura puede describirse por medio de modelos, dichos modelos estarán compuestos por componentes. Estos últimos son unidades abstractas que ofrecen una serie de servicios a través de sus interfaces y se comunican con el resto de componentes del sistema por medio de ellas. Los componentes en si encapsulan una funcionalidad especifica.

Estilos arquitectónicos

Los más relevantes son:

- Estilo de flujo de datos: Tuberías y filtros, es una arquitectura que conecta componentes computacionales (filtros) a través de conectores (pipes), de forma que los datos se transportan a través de tuberías entre los filtros, transformando gradualmente las entradas en salidas. En este estilo secuencial por lotes los componentes son programas independientes.

- Estilo centrado en Datos:

  Arquitecturas de pizarra o Repositorio: en esta arquitectura hay dos componentes principales: una estructura de datos que representa el estado actual y una colección de componentes independientes que operan sobre él[SG96]. Normalmente el centro de esta arquitectura es un almacén de datos y los componentes aceden a ella para actualizar, insertar, eliminar, consultar datos.
- Estilo de llamada y retorno: Esta familia de estilos enfatiza la modificabilidad y la escalabilidad. Son los estilos más generalizados en sistemas en gran escala.

  Arquitectura en capas: El estilo en capas se define como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inferior.

  Arquitecturas orientadas a objetos distribuidos: Según define Moreno (2013) una arquitectura orientada a objetos estructura el sistema en objetos débilmente acoplados con interfaces bien definidas. Los objetos llaman a servicios ofrecidos por otros objetos. Así, la interacción entre objetos tiene lugar a través de invocaciones de sus operaciones.
- Estilo peer to peer:

  Arquitectura basadas en eventos: Según define Moreno(2013) la idea dominante en las arquitecturas basadas en eventos es que, en lugar de invocar un procedimiento directamente, como se haría en un estilo orientado a objetos, un componente puede anunciar, mediante difusión, uno o más eventos(invocación explicita).

Definición de nuestra arquitectura software

Dentro de los requisitos no funcionales el cliente se requería que la plataforma tuviera acceso online y fuera distribuida, por todo lo anterior se decide llevar a cabo una arquitectura cliente servidor organizada y en cada capas y estructurada componentes los cuales se organizan siguiendo una arquitectura orientada a objetos. El resultado es una arquitectura heterogénea que es fácilmente mantenible, escalable y poco acoplada.

Dentro de nuestra arquitectura cliente servidor se organiza en tres capas, el capa presentación, capa lógica de negocio y el capa de persistencia.

Si realizamos una abstracción de los componentes funcionales nos encontraremos con cinco componentes funcionales dentro de cada uno de las capas de la arquitectura lógica.

Capa de presentación

En este apartado nos centraremos en la tecnología y los componentes que define JavaEE. En esta capa el objetivo primordial es hacer accesible la lógica de negocio a los clientes que acceden a través de HTTP.

Los principales objetivos de la capa de presentación son los siguientes:

- Permitir a los clientes llamar a la lógica de negocio y que el servidor les presente resultados.

- Gestionar el flujo de presentación, es decir decidir cuál es la siguiente pantalla que hay que presentar al usuario.
- Realizar ciertas validaciones de los datos de entrada.
- Mantener el estado de ciertas peticiones sobre el protocolo HTTP que no tiene estado.

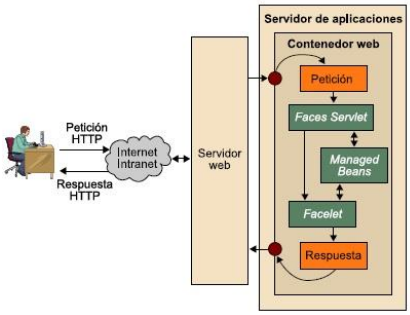
Nuestra arquitectura lógica a ya la hemos representado desdeun punto de vista de componentes de alto nivel, nos interesa realizar un refinamiento de componentes y aplicar el perfil de arquitectura Java EE.
Para ello hemos decidido aplicar ciertos patrones de diseño.Siguiendo una arquitectura Java EE y realizando un refinamiento de los componentes,despues del refinamiento nos decanremos por utilizar el framework JSF, ya que se basa en el patrón de diseño MVC, permimiendonos poder separar la parte de diseño de la parte que maneja las llamadas a la logica de negocioparte del modelo de dicho patrón se implementará en la capa de negocio.
Siguiendo el refinamiento anterior tomamos la decision de utilizar el patrón command.El controlador será el responsable de coordinar todo el proceso, pero la implementación será desacoplada.La acción del usuario con la interfaz gráfica se hará un mapeo 1:1 entre la acción y la vista.
Como controlador usaremos el patrón del nivel presentación de Java EE FrontControler el cual nos permitirá manejar las peticiones de una forma centralizada(incluye la invocación de servicios de seguridad como la autenticación y autorización, delega el procesamiento de negocio, controlar la elección de una vista apropiada).
A continuación describimos la técnología utilizada para despues detallar cada uno de los componenetes que integran nuestra arquitectura.

Java server faces

Capms(2011)

Java server faces es un framework de componentes de servidor que implementa el patrón MVC y sirve para construir aplicaciones web con java.

Las aplicaciones JSF “viven” dentro del paradigma petición-respuesta bajo el protocolo HTTP.



Como implementación del patrón MVC, Java Server Faces consta de vistas, modelo y controlador. Analizaremos como se implementa cada una de estas partes:

1. Las vistas proporcionan la interfaz de usuario de la aplicación web:

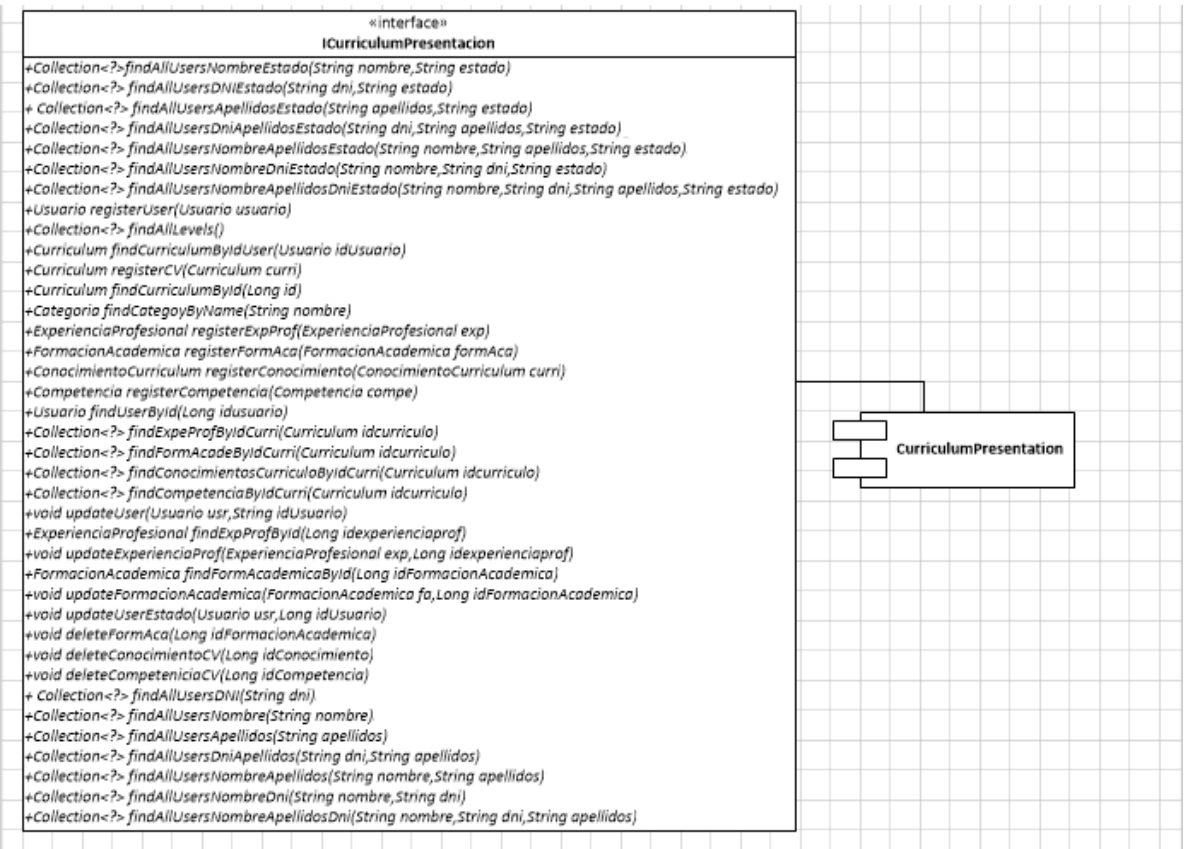
- Las vistas de una aplicación JSF se implementan con lenguajes de declaración de páginas. Lo más habitual es utilizar *facelets*.
- Las vistas permiten vincular los componentes de interfaz de usuario con el modelo , mediante un lenguaje de expresiones. Esta vinculación se hace mediante eventos.

Los *facelets* son el sistema por defecto para crear vistas con JSF. Son documentos similares a una página HTML, pero que pueden contener etiquetas de diferentes tipos(propias de *facelets*, etiquetas de JSF, etiquetas de JSLT, etc.) para generar componentes de interfaz y llevar a cabo otras tareas.

2. El modelo es el responsable de la lógica de la aplicación:
    - El modelo está implementado con *maneged bean*, objetos Java gestionados por el framework de JSF y que se encargan de implementar la lógica de negocio, responder a los eventos generados por los componentes de la interfaz de usuario y controlar la navegación entre páginas de los métodos de acción.
3. El controlador se encarga de gestionar la lógica de presentación:
    - El controlador esta implementado con un servlet llamado *Faces Servlet* y un conjunto de métodos de acción de los *maneged bean.*
    - Todas las peticiones HTTP del usuario son recogidas por el Faces Servlet,que s encarga de actualizar la interfaz de usuario, de los datos de los *mangedbean* y de llamar a los gestores de eventos y a los métodos de acción.

Una vez tenemos clara cual es la forma de la que trabaja el framework que vamos a utilizar para la capa de presentación, la representación de nuestro diagramas de componentes quedaria clara cual es sería su implementación las acciones se implementarán con *Managed Bean* el controlador ya viene implementado, y las vistas serán implementadas con *Facelets.*
Detalle de la capa de presentacion:

El componente Curriculo en su capa de presentación:
Estará compuesto por una interfaz definida un un componente CurriculumPresentation.
La interfaz será la siguiente:



El componente CurriculumPresentation estará compuesto por dos componentes uno denominado CurriculumView y otro CurriculumControler.

El componente <u>CurriculumControler:</u>

Contendrá:

- Componente FrontControler : el cual vendrá implementado por servlet del framework JSF.
- Componete AnadirCompe: el cual estará impletado el ManagedBean llamado AnadirCompe.java.
- Componente AnadirCompeCV: el cual estará impletado el ManagedBean AnadirCompeCV.java.
- Componente AnadirConoc: el cual estará impletado el ManagedBean AnadirConoc.java.
- Componente AnadirConocCV: el cual estará impletado el ManagedBean AnadirConocCV.java.
- Componente AnadirDataPerson: el cual estará impletado el ManagedBean AnadirDataPerson.java.
- Componente AnadirExpProf: el cual estará impletado el ManagedBean AnadirExpProf.java.
- Componente AnadirFormAca: el cual estará impletado el ManagedBean AnadirFormAca.java.
- Componente BorrarCompe: el cual estará impletado el ManagedBean BorrarCompe.java.
- Componente BorrarConoc: el cual estará impletado el ManagedBean BorrarConoc.java.

- Componente BorrarFormAca: el cual estará impletado el ManagedBean BorrarFormAca.java.
- Componente ModifDataPerson: el cual estará impletado el ManagedBean ModifDataPerson.java.
- Componente ModifFormAca: el cual estará impletado el ManagedBean ModifFormAca.java.
- Componente ModifiExpeProfe: el cual estará impletado el ManagedBean ModifiExpeProfe.java.
- Componente ShowCurriculum: el cual estará impletado el ManagedBean ShowCurriculum.java.
- Componente AnadirFormAca: el cual estará impletado el ManagedBean AnadirFormAca.java.
- Componente AnadirFormAca: el cual estará impletado el ManagedBean AnadirFormAca.java.

El componente CurriculumView:Contendrá las vistas

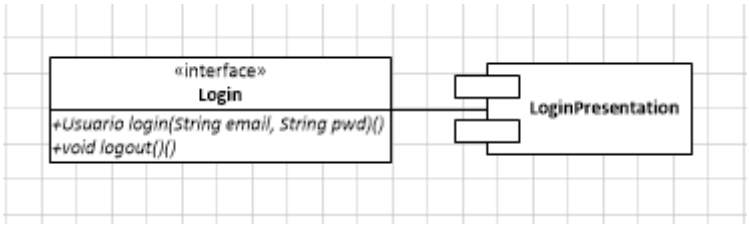
Contendrá:

- Componente anadirCompe: el cual estará implementado por el facelet anadirCompe.xhtml
- Componente anadirCompeCV: el cual estará implementado por el facelet anadirCompeCV.xhtml
- Componente anadirConoc: el cual estará implementado por el facelet anadirConoc.xhtml
- Componente anadirConocCV: el cual estará implementado por el facelet anadirConocCV.xhtml
- Componente anadirCompe: el cual estará implementado por el facelet anadirCompe.xhtml
- Componente anadirDataPerson: el cual estará implementado por el facelet anadirDataPerson.xhtml
- Componente anadirFormAca: el cual estará implementado por el facelet anadirFormAca.xhtml
- Componente anadirFormAcaCV: el cual estará implementado por el facelet anadirFormAcaCV.xhtml
- Componente borrarCompet: el cual estará implementado por el facelet borrarCompet.xhtml
- Componente borrarCompet: el cual estará implementado por el facelet borrarCompet.xhtml
- Componente borrarFormAcade el cual estará implementado por el facelet borrarFormAcade.xhtml
- Componente buscarCV: el cual estará implementado por el facelet buscarCV.xhtml
- Componente clientDetailView: el cual estará implementado por el facelet clientDetailView.xhtml

- Componente clientListView: el cual estará implementado por el facelet clientListView.xhtml
- Componente modifCompet: el cual estará implementado por el facelet modifCompet.xhtml
- Componente modifExpeProfe: el cual estará implementado por el facelet modifExpeProfe.xhtml
- Componente modifFormAca: el cual estará implementado por el facelet modifFormAca.xhtml
- Componente modifiDataPerson: el cual estará implementado por el facelet modifiDataPerson.xhtml
- Componente showCurriculum: el cual estará implementado por el facelet showCurriculum.xhtml
- Componente usuarioBajaLogicaCV: el cual estará implementado por el facelet usuarioBajaLogicaCV.xhtml

El componente Login en su capa de presentación:

Estará compuesto por una interfaz definida un un componente LoginPresentation.
La interfaz será la siguiente:



El componente LoginControler:
Contendrá:

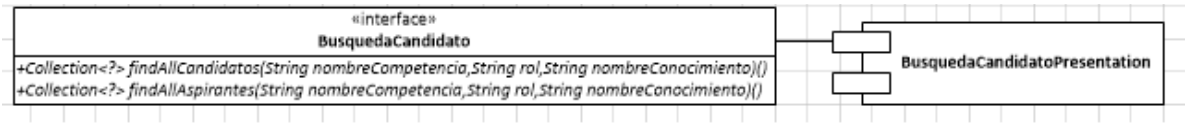
- Componente FrontControler : el cual vendrá implementado por servlet del framework JSF.
- Componete Login: el cual estará impletado el ManagedBean llamado Login.java.

El componente LoginView:
Contendrá:

- Componente loginView: el cual vendrá implementado por el facelet loginView.xhtml.

El componente Buscar Candidato en su capa de presentación:

Estará compuesto por una interfaz definida un un componente BuscarCandidatoPresentation.
La interfaz será la siguiente:



El componente BuscarCandidatoControler:

Contendrá:

- Componente FrontControler : el cual vendrá implementado por servlet del framework JSF.
- Componete BuscarCandidato: el cual estará impletado el ManagedBean llamado BuscarCandidato.java.

El componente BuscarCandidato View:

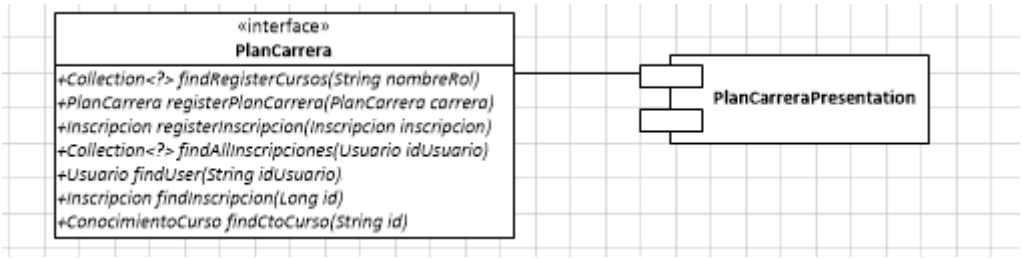
Contendrá:

- Componente buscarCandidato: el cual vendrá implementado por el facelet buscarCandidato.xhtml.

El componente Plan de carrera en su capa de presentación:

Estará compuesto por una interfaz definida un un componente PlanCarreraPresentation.

La interfaz será la siguiente:



El componente PlanCarreraControler:

Contendrá:

- Componente FrontControler : el cual vendrá implementado por servlet del framework JSF.
- Componete PlanCarreraShow: el cual estará impletado el ManagedBean llamado PlanCarreraShow.java.java.
- Componente PlanCarreraShowfromBuscador: el cual estará implemnetado por un ManagedBean llamado PlanCarreraShowfromBuscador.java.

El componente PlanCarreraView:

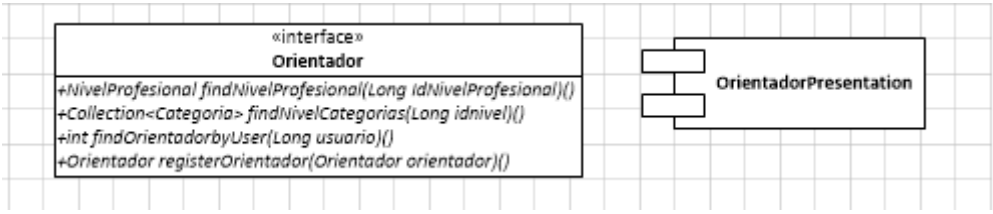
Contendrá:

- Componente planCarreraShow: el cual vendrá implementado por el facelet planCarreraShow.xhtml.
- Componente planCarreraShowfromBuscador: el cual vendrá implementado por el facelet planCarreraShowfromBuscador.xhtml

El componente Orientador en su capa de presentación:

Estará compuesto por una interfaz definida un un componente OrientadorPresentation.

La interfaz será la siguiente:



El componente OrientadorControler:

Contendrá:

- Componente FrontControler : el cual vendrá implementado por servlet del framework JSF.
- Componete OrientadorMB : el cual estará impletado el ManagedBean llamado OrientadorMB.java.

El componente OrientadorView:

Contendrá:

Componente Orientador: el cual vendrá implementado por el facelet Orientador.xhtml

Capa de negocio

En esta apartado hablaremos de los componentes Java EE llamados EJB, estos son un tipo de componentes distribuidos y son los que utilizaremos para implementan nuestra lógica de negocio (en su versión 3.0.).

Según Camps Enterprise Java Beans es una tecnología de componentes de servidor que permite el desarrollo y el despliegue de aplicaciones empresariales distribuidas basadas en componentes.

Las aplicaciones desarrolladas con EJB nos aportan una serie de características técnicas que son imprescindibles para el desarrollo nuestro software, estas son : escalables, seguras, transaccionales , multiusuario y distribuidas.

Además la arquitectura EJB nos aporta una serie de principios :

- Los EJB tienen una interfaz pública y los usuarios interactúan únicamente con los métodos de dicha interfaz.
- Las llamadas de los usuarios a los métodos de los EJB son interceptadas por el contenedor de EJB antes de que pasen el control al componente.
- El contenedor gestiona las transacciones, la seguridad, y todos los temas relacionados con la concurrencia.
- Las instancias de los EJB se pueden almacenar en bancos de instancias por temas de rendimiento, de esta forma se evita el tener que instanciar un EJB cada vez que lo llama un cliente.

EJBs

Pradel (2011)

Los ejbs se definen como objetos distribuidos, de forma que se pueden llamar remotamente. Los EJBs utilizan RMI como tecnología subyacente para hacer las llamadas remotas.

Las comunicación distribuida nos soluciona el problema de la distribución de objetos y nos permite hacer llamadas a objetos remotos como si estuviéramos haciendo llamadas locales.

Podemos encontrar distintos tipos de EJBs:

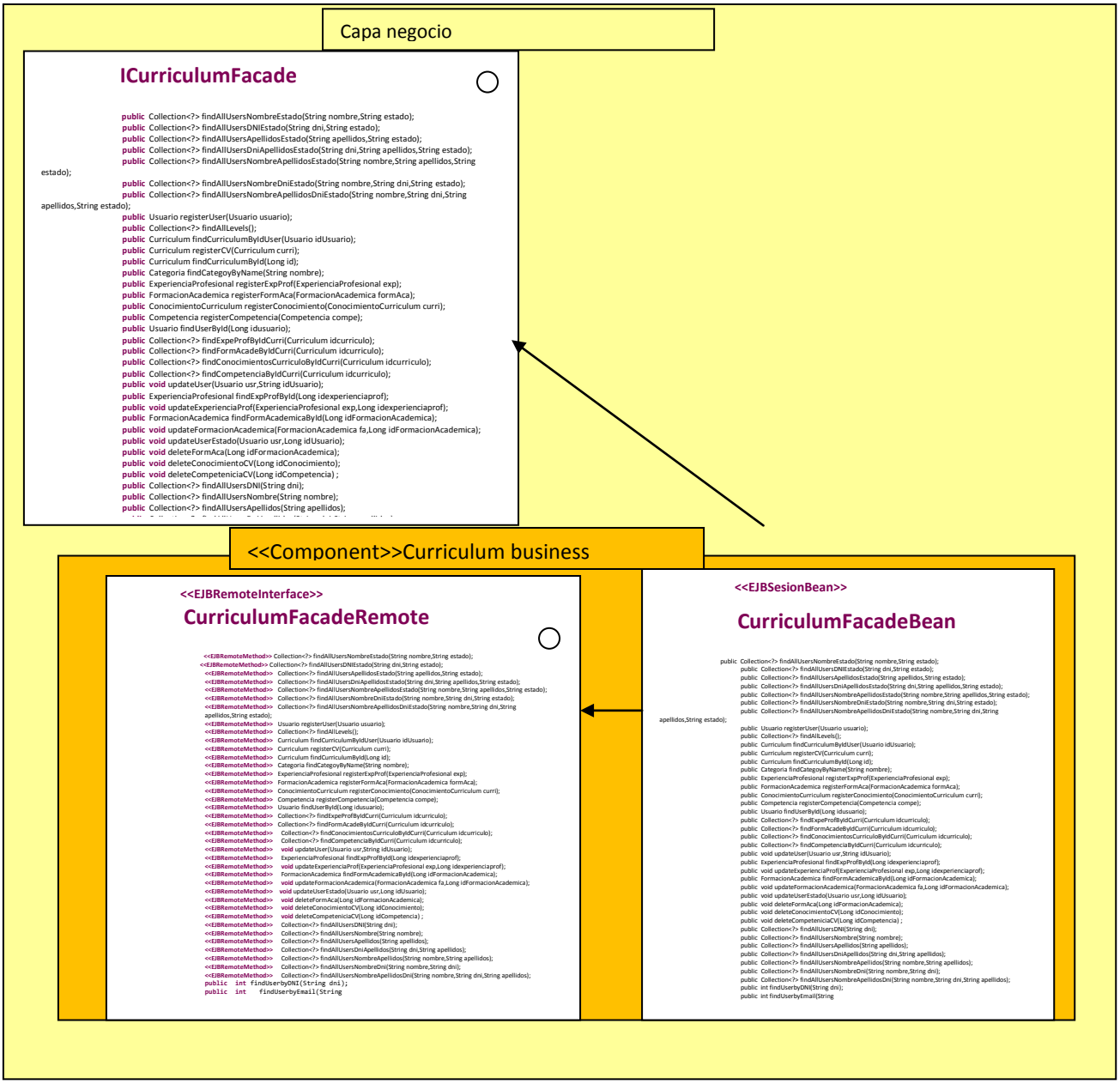
Los ejbs de sesión :que son objetos no persistentes, que se limita su ciclo de vida a una interacción con el cliente, en una o varias llamadas. Existen dos tipos de EJB de sesión.

- EJB de sesión sin estado: representan procesos de negocio que atienden peticiones de cliente de forma síncrona de una única invocación, es decir el cliente llama al método del EJB de sesión sin estado y el servidor devuelve la respuesta que ha procesado el EJB de sesión sin estado. Estos EJB ofrecen un rendimiento bastante aceptable, además de poder ser sustituidos fácilmente por servicios web, en el marco de una arquitectura SOA.
- EJB de sesión con estado: representan procesos de negocio que realizan varias interacciones entre el cliente y el servidor. Este tipo de EJB almacena el estado conversacional del cliente entre varias llamadas en los atributos del propio EJB. Cada cliente tiene asociado una instancia de un EJB. La gestión de este tipo de EJBs requiere de un mayor consumo de recursos por parte del servidor por eso su rendimiento es peor que el de los EJB se sesión sin estado.

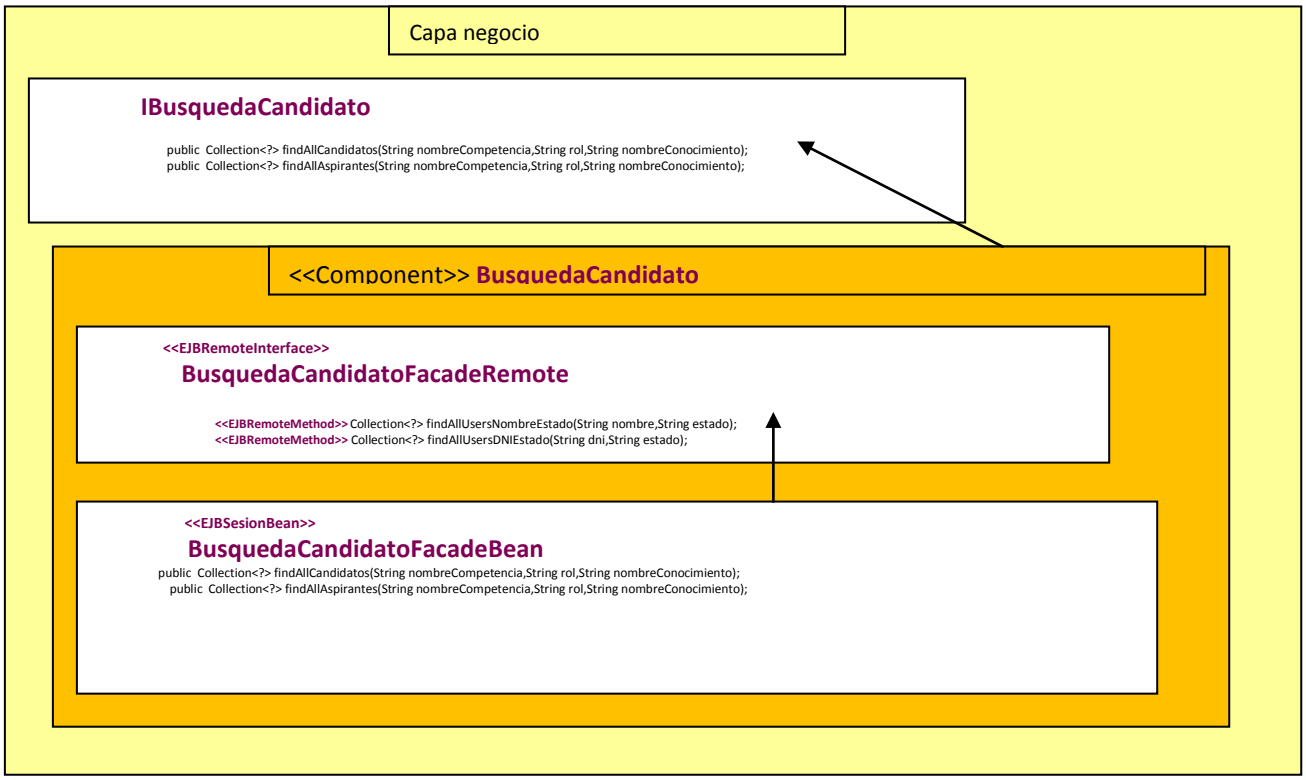
EJB de mensaje: Según Camps (2011)Son componentes transaccionales sin estado, de servidor, que sirven para responder a mensajes asíncronos.

Por todo lo comentado anteriormente y por las características no funcionales del software requeridas por el cliente, tomamos la decisión de utilizar EJBs de sesión sin estado para representar nuestra capa de lógica de negocio bajo la arquitectura Java EE.
Nuestra la arquitectura lógica de nuestra capa de negocio vendrá definida por diferente componentes y su representación grafica es la siguiente:

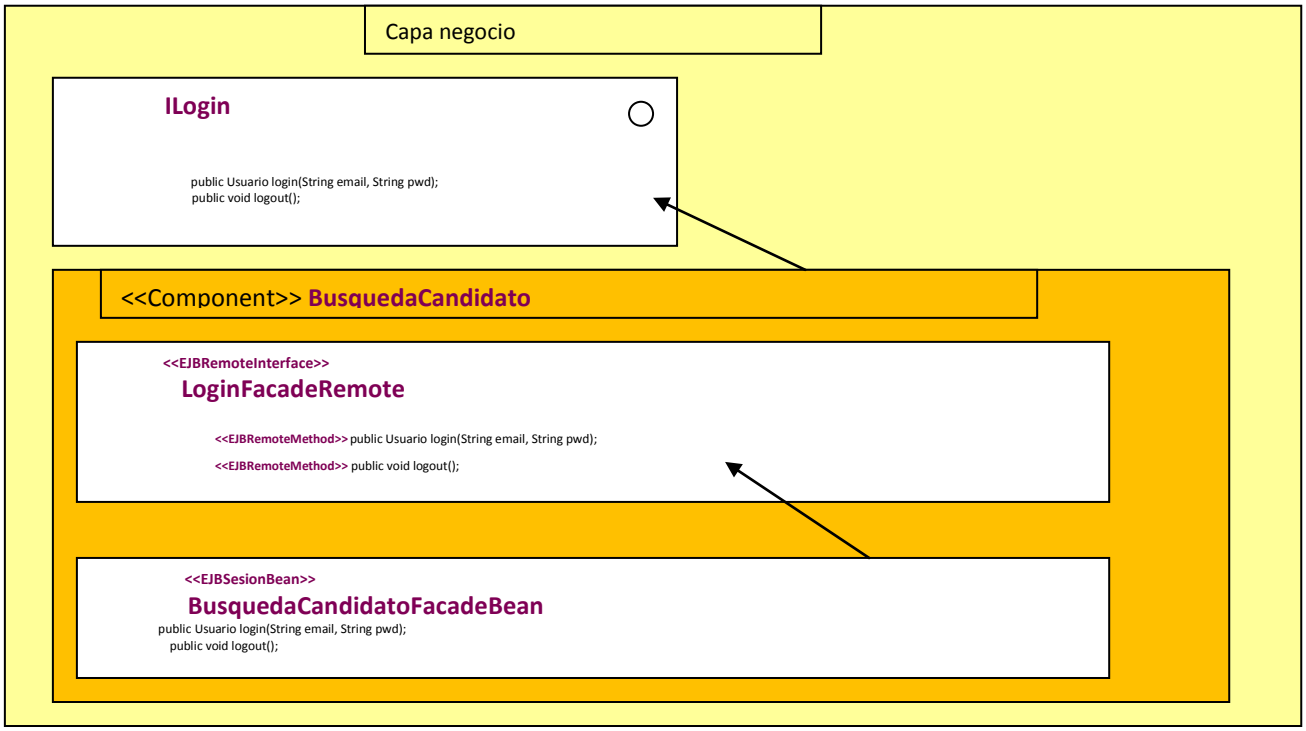
Capa de negocio componente Curriculum.



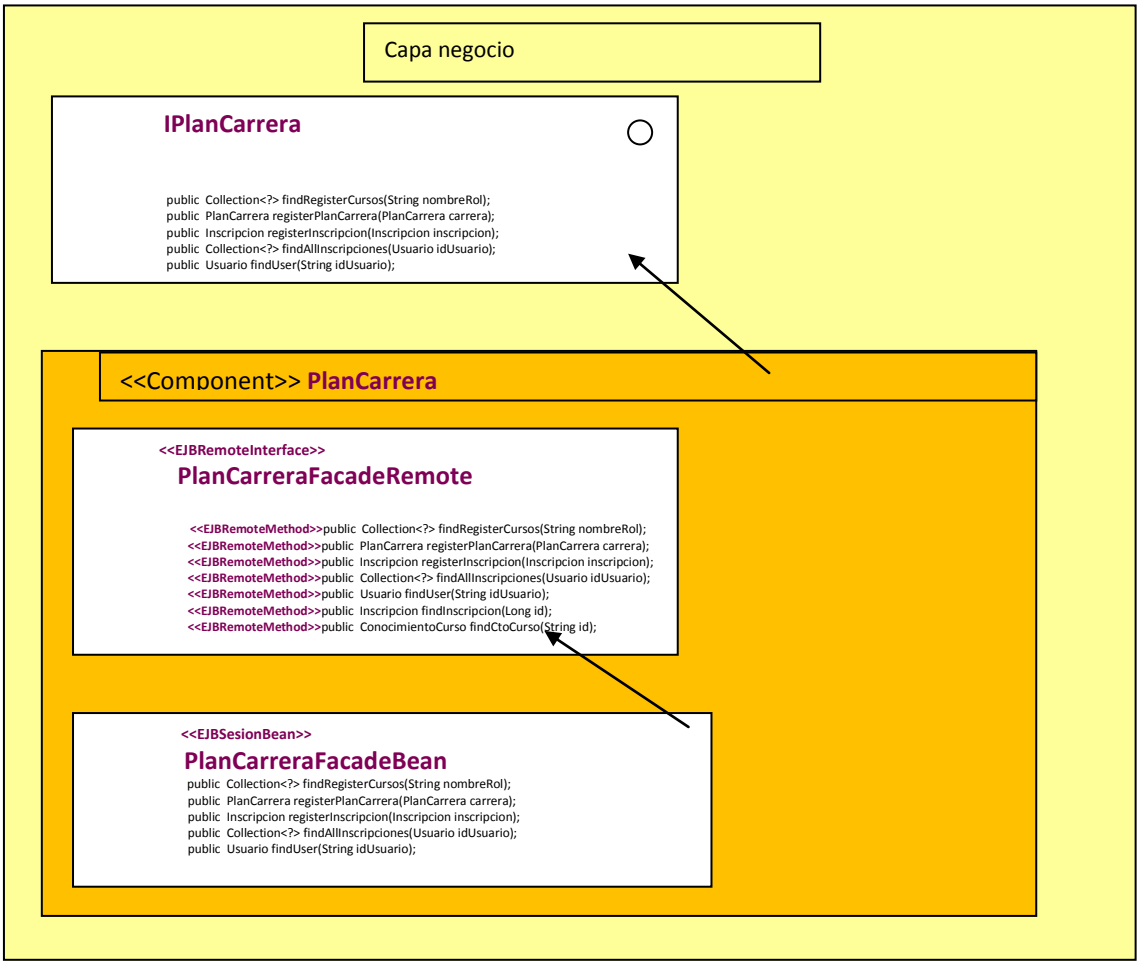
Capa de negocio componente Búsqueda de candidato.



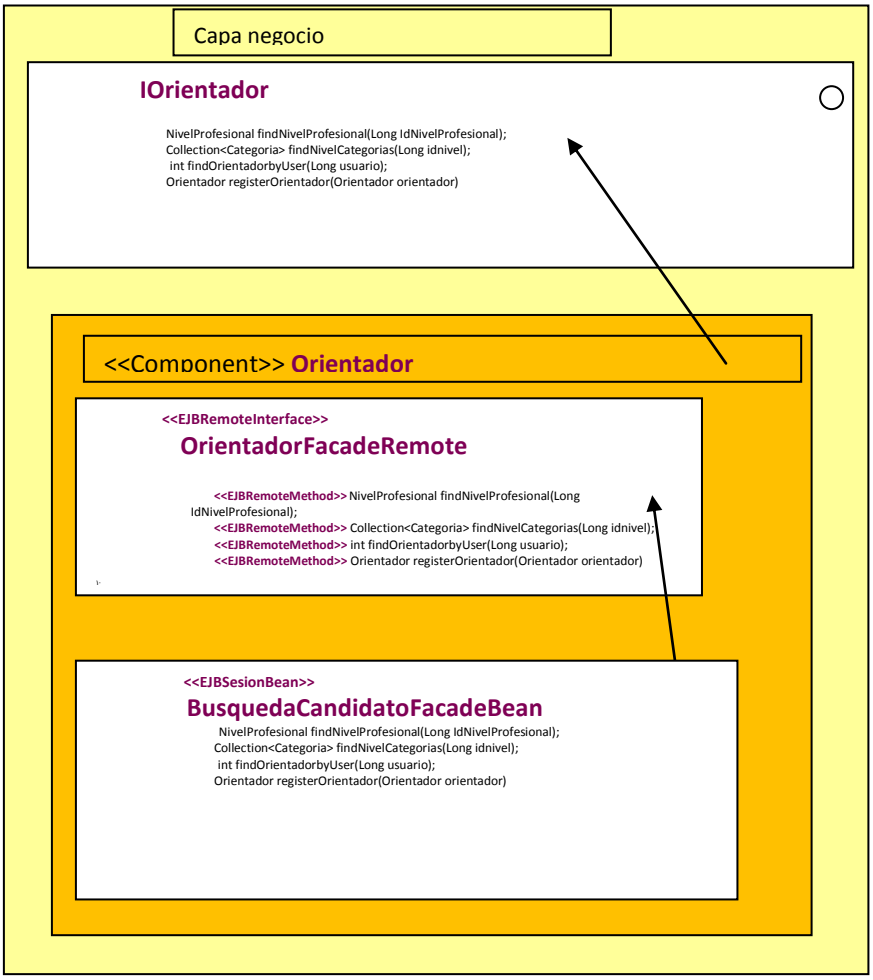
Capa de negocio componente Login.



Capa de negocio componente PlanCarrera.



Capa de negocio componente Orientador



Capa de persistencia

En esta capa hablaremos de la tecnología que usa java para implementar la capa de persistencia, para ello utilizaremos JPA (Java Persistence API).

Camps (2011)

JPA ofrece un framework de mapeo objeto-relación entre objetos Java y las tablas de una base de datos relacional; y presenta las siguientes características:

- Simplifica el desarrollo de aplicaciones que tengan que acceder a datos almacenados en base de datos relacionales y manipularlos.
- Ofrece persistencia de objetos de manera transparente.
- Hace que nuestro código no sea dependiente de un sistema gestor de base de datos concreto.
- Define cómo hay que describir el mapeo entre objetos y tablas.
- Define la interfaz con el proveedor de persistencia; este proveedor será el que interactúe con la base de datos.

Las entidades JPA son objetos persistentes porque duran los mismo que los datos que representan, además no tienen lógica de negocio, salvo manipulaciones simples de los datos que representan.
Una entidad JPA representa una tabla en una base de datos relacional, una instancia de una entidad JPA representa una tabla en una base de datos relacional, una instancia de una fila de la tabla y cada campo persistente de la entidad, una columna.
Según todo lo anterior detallamos nuestro modelo de persistencia.
Tenemos los siguientes JPA:
Categoria: Esta entidad representa las categorías profesionales del sistema.
Atributos:

- idCategoria: Identificador único
- descripcion: Información sobre la categoría.
- nombre: nombre de la categoría

CategoriaConocimiento: Esta entidad representa los conocimientos que se deben poseer por tener una determinada categoria profesional.
Atributos:

- idCategoria:Identificador de la categoria a la que se referencia.
- idCategoriaConocimiento:Identificador único.
- idCategoriaCurso: Identificador del curso correspondiente al conocimiento.
- nombreConocimiento: Nombre del conocimiento técnico.

Competencia: Es la entidad representa las competencias que tiene un empleado en su curriculum.

- IdCompetencia:identificador único.
- IdCurriculo:Id del curriculo al que pretenece la competencia.
- nombreCompetencia:Nombre de la competencia.

ConocimientoCurriculum: Es la entidad que representa los conocimientos de un empleado en su curriculo.

- idConocimiento: Identificador único.
- idCurriculo:Identificador del curriculo al que pertenece.
- nombreConocimiento: Nombre del conocimiento.

ConocimientoCurso: Es la entidad que guarda el nombre del conocimiento asociado a su curso.

- idConocimientoCurso: Identificador único.
- nombreCtoCurso:Nombre del conocimeinto técnico
- nombreCurso: Nombre del curso asociado al conocimiento técnico.
-

Curriculum: Entidad que representa a el curriculum de un empleado.

- idUsuario:Es el id usuario al que pertenece el curriculum.
- idCurriculum:Identificar único.

ExperienciaProfesional: Entidad que identifica la experiencia profesional de un empleado

- idCurriculo:Identificador del curriculum al que pertenece la experiencia profesional.
- fechaInicio:Fecha en que comienza la experiencia laboral.
- fechaFin:Fecha que termina la relacion laboral.

- empresa: Empresa en la que se realiza el trabajo.
- descripcionPuesto: Descripcion del puesto de trabajo.
- idexperienciaprof:Identificador único.

FormacionAcademica: Entidad que identifica la formación académica del empleado.

- idCurriculo: Identificador del curriculum que pertenece la formación academica.
- fechaInicio:Fecha en la que comenzo a estudiar.
- fechaFin:Fecha en la que finalizo los estudios.
- titulo: nombre del titulo conseguido.

Inscripcion: Entidad que recoge la inscripcion de un empleado a un curso.

- id:identificador único.
- fechaInscripcion:Fecha en la que se crea la inscripción.
- idUsuario:id del usuario que se inscribe.
- idConocimientoCurso: id del curso al que te inscriben.

NivelProfesional: Es una entidad que define el area profesional.

- nombre:Nombre del nivel profesional
- descripcion:Descripcion del puesto.
- idNivelProfesional:Identificador unico.
- listCategorias:Categorias de un mismo nivel.

PlanCarrera: Define los cursos que ha realizado ó debe realizar el empleado.

- id:Identificador único.
- inscripcionid:Recoge la inscripcion de un empleado a un curso.

Usuario: Es una entidad que define al empleado.

- idusuario:identificador unico.
- apellidos:apellidos del empleado.
- dni:numero identificativo y único del empleado.
- email:correo electronico y usuario de entrada del usuario
- estado:Estado en el que se encuentra el usuario puede tener los valores Activo, Inactivo, BajaMaternal, BajaEnfermedad, Jubilado.
- idCurriculo: Define la entidad de curriculo del empleado.
- isadmin: campo que define su un usuario es empleado o gestor de rrhh.
- nombre: Nombre del empleado.
- password: Guarda el password del empleado.
- rol: que despeña el empleado y puede tener estos valores:
  Programador, Analista Organico, Analista Funcional, Tecnico Microinformatico, Jefe de Proyecto, Administrador de base de datos, Analista Programador, Administrador de Sistemas, Consultor, Tecnico de Sistemas
- telefono:Campo que define el telefono del empleado.

7. Pruebas de software

[Xavier Escudero]

Niveles de prueba

Las pruebas se pueden realizar en diferentes niveles, según que queramos pro- bar de forma aislada un componente, la integración entre componentes o el funcionamiento del sistema como un todo. Esta clasificación es la denomina- da de niveles de prueba, y en ella podemos diferenciar:

1)Pruebas unitarias o de componente. Tienen como objetivo probar cada componente (módulo, clase...) de forma aislada del resto del sistema. Están consideradas como una actividad dentro del proceso de construcción, y, por lo tanto, las realiza el equipo de desarrollo.

2)Pruebas de integración. Se prueban las interfaces entre componentes, di- ferentes partes del sistema (entre un componente y el sistema operativo, por ejemplo), o bien entre sistemas (entre un sistema de nóminas y un sistema externo bancario, por ejemplo). Suele realizarlas el equipo de pruebas, aunque en algunos casos también las efectúa el equipo de desarrollo.

3)Pruebas de sistema. Tienen como objetivo encontrar defectos y validar que el sistema cumple las especificaciones, comparando los resultados obtenidos con los requisitos del sistema. Son realizadas por el equipo de pruebas.

4)Pruebas de aceptación. Es el último nivel, en el que el cliente tiene que validar que el desarrollo ha cumplido lo que se requería de él y no hay riesgos en la puesta en marcha del aplicativo.

Cuanto mejor se haya realizado un nivel de pruebas, mayor será la eficiencia de los siguientes niveles y menores defectos se encontrarán en ellos.

En todos los niveles se pueden validar requisitos tanto funcionales como no funcionales. Así, por ejemplo, podemos comprobar una funcionalidad en un componente o en el sistema y, también, el rendimiento del componente o del sistema.

Nuestro software después de ser sometido a pruebas de componentes se sometió a pruebas integradas y la documentación de las mismas es la siguiente:

	Curriculum- Alta de curriculum		
Subsistemas Implicados:	Ninguno		
	Crear un curriculum		
Nº	Entrada	Acción Esperada	**Verificar**
1	Seleccionar del menú de la izquierda Alta	Muestre pantalla de alta de datos personales	**ok**
2	Dejar los campos en blanco y pulsar guardar	Muestran pantallas de alta datos personales con mensajes de error	Ok

3	Introducir todos campos pero introducimos un email o un dni que ya existe en la base de datos	Muestra mensaje de error "Existe el dni en la base de datos" Existe el email en la base de datos	Ok
4	Introducimos todos los datos correctamente y pulsamos guardar	Se crea los datos personales el plan de carrera con las inscripciones correspondientes a la categoria profesional que tiene y muestra la pantalla de experiencia profesional	Ok
5	No rellenamos ningún campo pulsamos guardar.	Se muestra los mensajes de error de validación de cada uno de los campos	Ok
6	Rellenamos todos los campos y pulsamos guardar.	Se crea la experiencia profesional y te muestra el botón de añadir experiencia profesional. Y el botón de siguiente.	Ok
7	Pulsamos añadir experiencia profesional	Se muestra de nuevo la pantalla de añadir experiencia profesional	Ok
8	Rellenamos todos los campos y pulsamos siguiente.	Se guarda la experiencia profesional y se muestra la pantalla de formación profesional	Ok
7	No rellenamos ningún campo y pulsamos guardar	Se muestran mensajes de error	Ok

8	Rellenamos los campos y pulsamos guardar	Se muestran los botones de añadir formación académica y siguiente	Ok
9	Se pulsa de botón de añadir formación académica	Se muestra de nuevo la pantalla de formación académica.	Ok
10	Rellenamos todos los campos pulsamos guardar	Se muestran los botones de siguiente y añadir formación académica y se guarda la formación académica.	Ok
11	Pulsamos el botón de siguiente	Se muestra la pantalla de conocimiento.	Ok
12	Debemos de seleccionar un conocimiento del combo y pulsar guardar	Se guarda el conocimiento y se muestra los botones de guardar conocimiento	Ok
13	Pulsamos añadir conocimiento	Se muestra de nuevo la pantalla de conocimiento.	Ok
14	Seleccionamos del combo un conocimiento y pulsamos guardar y siguiente	Se guarda el conocimiento y se muestra la pantalla de competencia	Ok
15	Seleccionamos del combo una competencia pulsamos guardar	Se guarda la comeptencia y se muestran los botones de añadir competencia y finalizar	Ok

16	Pulsamos añadir nueva competencia y pulsamos guardar y finalizar	Se guarda la competencia y se muestra la pantalla de curriculo.	Ok

	Curriculum- Búsqueda de curriculum		
Subsistemas Implicados:	Ninguno		
Nº	Entrada	Acción Esperada	**Verificar**
1	Pinchar en el botón Curriculum del link superior	Muestra el buscador de curriculum vitae	Ok
2	Rellenar alguno de los campos y pulsar buscar	Se muestra un listado de empleados que cumplen el filtro de búsqueda y se encuentran activos	Ok
3	Chequeamos "*Todos los usuarios*" y pulsamos buscar	Se muestra un listado de todos los empleados que cumplen los filtros de búsqueda y además están activos.	Ok

	Curriculum- Búsqueda de curriculum		
Subsistemas Implicados:	Ninguno		
Nº	Entrada	Acción Esperada	**Verificar**
1	Pinchar en el botón Plan de carrera del link superior y le hacemos link	Muestra el buscador de plan de carrera	Ok

2	En el lado derecho aparece un link que pone Búsqueda de curriculum	Se muestra el buscador de curriculum	Ok
3	Rellenamos alguno o todos los campos pulsamos buscar	Se muestra un listado de empleados que cumplen el filtro de búsqueda y se encuentran activos	Ok
4	Chequeamos "*Todos los usuarios*" y pulsamos buscar	Se muestra un listado de todos los empleados que cumplen los filtros de búsqueda y además están activos.	Ok

	Curriculum- Búsqueda de curriculum		
Subsistemas Implicados:	Ninguno		
Nº	Entrada	Acción Esperada	**Verificar**
1	Pinchar en el botón buscador del link superior	Muestra el buscador de plan de candidatos	Ok
2	En el lado derecho aparece un link que pone Búsqueda se le hace link	Se muestra el buscador de curriculum	Ok
3	Rellenamos alguno o todos los campos pulsamos buscar	Se muestra un listado de empleados que cumplen el filtro de búsqueda y se encuentran activos	Ok
4	Chequeamos "*Todos los usuarios*" y pulsamos buscar	Se muestra un listado de todos los empleados que cumplen los filtros de búsqueda y además están activos.	Ok

	Curriculum- Eliminación de curriculum		
Subsistemas Implicados:	Búsqueda curriculum 1.2		
Nº	Entrada	Acción Esperada	**Verificar**
1	Después de realizar la búsqueda aparece un listado de empleados y en cada una de las filas un link que pone Eliminar, pulsamos este link	Muestra una pantalla con los datos del empleado a desactivar	Ok
2	Pulsa eliminar y finalizar	El sistema desactiva al empleado dejando de aparecer en la búsqueda de empleados activos(Sin chequear el check "*Todos los usuarios*")	Ok

	Curriculum- Modificación de curriculum		
Subsistemas Implicados:	Búsqueda curriculum 1.2		
Nº	Entrada	Acción Esperada	**Verificar**
1	Después de realizar la búsqueda aparece un listado de empleados y en cada una de las filas un link que pone Modificar, pulsamos este link	Muestra una pantalla el curriculum vitae del empleado del que queremos modificar los datos	Ok
2	Podemos pinchar en cualquier parte del Curriculum del empleado y modificarlo	El sistema muestra la pantalla del detalle de los datos a modificar del link de modificar clickeado. Por ejemplo Datos Personales modificar.	Ok

	Curriculum- Modificación de datos personales
Subsistemas Implicados:	Modificar curriculum 1.4

Nº	Entrada	Acción Esperada	Verificar
1	Desde el detalle del CV de un empleado pulsamos modificar en el área de Datos personales	Muestra una pantalla con el detalle de los datos personales del empleado	Ok
2	Modificamos el campo que queremos cambiar y pulsamos guardar y finalizar	El sistema guarda los cambios y vuelve a la búsqueda de curriculum	Ok

	Curriculum- Modificación de datos personales		
Subsistemas Implicados:	Modificar curriculum 1.4		
Nº	Entrada	Acción Esperada	**Verificar**
1	Desde el detalle del CV de un empleado pulsamos modificar en el área de Datos personales	Muestra una pantalla con el detalle de los datos personales del empleado	Ok
2	Modificamos el campo que queremos cambiar y pulsamos guardar y finalizar	El sistema guarda los cambios y vuelve a la búsqueda de curriculum	Ok

	Curriculum- Modificación de experiencia profesional		
Subsistemas Implicados:	Modificar curriculum 1.4		
Nº	Entrada	Acción Esperada	**Verificar**
1	Desde el detalle del CV de un empleado pulsamos modificar en el área de expediente académico	Muestra una pantalla con el detalle de la experiencia profesional del empleado	Ok
2	Modificamos el campo que queremos cambiar y pulsamos guardar y finalizar	El sistema guarda los cambios y vuelve a la búsqueda de curriculum	Ok

	Curriculum- Modificación de formación académica

Subsistemas Implicados:	Modificar curriculum 1.4		
Nº	Entrada	Acción Esperada	**Verificar**
1	Desde el detalle del CV de un empleado pulsamos modificar en el área de formación académica.	Muestra una pantalla con el detalle de la formación académica del empleado	Ok
2	Modificamos el campo que queremos cambiar y pulsamos guardar y finalizar	El sistema guarda los cambios y vuelve a la búsqueda de curriculum	Ok

	Curriculum- Eliminar formación académica		
Subsistemas Implicados:	Modificar curriculum 1.4		
Nº	Entrada	Acción Esperada	**Verificar**
1	Desde el detalle del CV de un empleado pulsamos eliminar en el área de formación académica.	Muestra una pantalla con el detalle de la formación académica del empleado	Ok
2	Pulsamos borrar y finalizar	El sistema borra los datos y vuelve a la búsqueda de curriculum	Ok

	Curriculum- Añadir formación académica		
Subsistemas Implicados:	Modificar curriculum 1.4		
Nº	Entrada	Acción Esperada	**Verificar**
1	Desde el detalle del CV de un empleado pulsamos añadir en el área de formación académica.	Muestra una pantalla con los campos vacios de la formación académica.	Ok
2	Dejamos los campos vacios y pulsamos guardar	El sistema muestra mensajes de error	Ok

3	Rellenamos todos los campos y pulsamos guardar y finalizar	El sistema guarda los datos y muestra de nuevo la búsqueda de curriculo.	Ok

	Curriculum- Añadir de conocimientos		
Subsistemas Implicados:	Modificar curriculum 1.4		
Nº	Entrada	Acción Esperada	**Verificar**
1	Desde el detalle del CV de un empleado pulsamos añadir en el área de conocimientos	Muestra una pantalla con una lista desplegable.	Ok
2	Seleccionamos de la lista desplegable un conocimiento técnico, pulsamos guardar y finalizar	El sistema guarda los datos y muestra de nuevo el buscador de cv.	Ok

	Curriculum- Eliminar de conocimientos		
Subsistemas Implicados:	Modificar curriculum 1.4		
Nº	Entrada	Acción Esperada	**Verificar**
1	Desde el detalle del CV de un empleado pulsamos borrar en el área de conocimientos	Muestra una pantalla con una serie de checks con el nombre de los conocimientos a borrar.	Ok
2	Seleccionamos los checks que queremos borrar pulsamos guardar y finalizar	El sistema borra los datos y muestra de nuevo el buscador de cv.	Ok

	Curriculum- Añadir Competencias		
Subsistemas Implicados:	Modificar curriculum 1.4		
Nº	Entrada	Acción Esperada	**Verificar**

1	Desde el detalle del CV de un empleado pulsamos añadir en el área de competencias.	Muestra una pantalla con una lista desplegable con las competencias a elegir.	Ok
2	Seleccionamos una competencia de la lista pulsamos guardar y finalizar	El sistema añade los datos y muestra de nuevo el buscador de cv.	Ok

	Curriculum- Orientador		
Subsistemas Implicados:	Ninguno		
Nº	Entrada	Acción Esperada	**Verificar**
1	Después de logados en el sistema con empleado a través de nuestro email pinchamos en Orientador	Muestra una pantalla con todas las categorías disponibles	Ok
2	Seleccionamos una de las categorías.	El sistema añade al plan de carrera todos los cursos correspondientes a la categoria y deja seleccionada la categoría marcada.	Ok

	Curriculum- Búsqueda candidato/Mostrar Plan Carrera empleado		
Subsistemas Implicados:	Ninguno		
Nº	Entrada	Acción Esperada	**Verificar**
1	Después de logados en el sistema como admin@admin pinchamos	El sistema nos muestra la búsqueda de candidato.	Ok

	en Búsqueda		
2	Seleccionamos una de cada uno delo combos un conocimiento una categoria y una competencia.	Muestra el resultado del candidato o candidatos que cumples los filtros y de los aspirantes que cumplen algunas de los filtros de la búsqueda.	Ok
3	En el listado de candidatos hacemos un link a plan de carrera de uno de los candidatos	El sistema muestra el plan de carrera del candidato	Ok
4	Aparece el listado histórico de cursos y uno de ellos tiene un botón de completar y lo pulsamos	El sistema completa la inscripcion y muestra de nuevo la búsqueda.	Ok

	Curriculum- Búsqueda candidato/Mostrar Plan Carrera empleado		
Subsistemas Implicados:	Ninguno		
Nº	Entrada	Acción Esperada	**Verificar**
1	Después de logados en el sistema como admin@admin pinchamos en Búsqueda	El sistema nos muestra la búsqueda de candidato.	Ok
2	Seleccionamos una de cada uno delo combos un conocimiento una categoria y una competencia.	Muestra el resultado del candidato o candidatos que cumples los filtros y de los aspirantes que cumplen algunas de los filtros de la búsqueda.	Ok
3	En el listado de candidatos hacemos un link a plan de carrera de uno de los candidatos	El sistema muestra el plan de carrera del candidato	Ok

4	Aparece un listado de cursos que podemos inscribir y cada uno de ellos tiene un botón de inscribir y lo pulsamos	El sistema inscribe al usuario en el curso y muestra de nuevo la búsqueda.	Ok

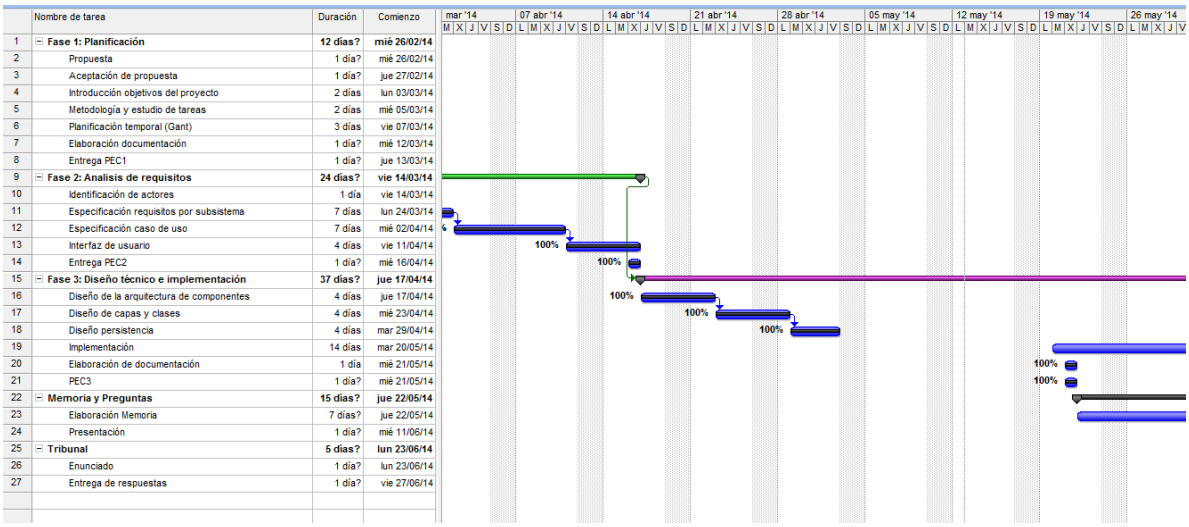
	Curriculum- Búsqueda Plan Carrera/Mostrar Plan Carrera Empleado		
Subsistemas Implicados:	Ninguno		
Nº	Entrada	Acción Esperada	**Verificar**
1	Después de logados en el sistema como admin@admin pinchamos en Búsqueda	El sistema nos muestra la búsqueda de candidato.	Ok
2	Seleccionamos el link de Plan de Carrera	Muestra el buscador de planes de carrera.	Ok
3	Se rellenan los datos de lo que buscas y se pulsa buscar	Se muestra el resultado del buscador los empleados que cumplen los filtros de búsqueda	Ok
4	Se pulsa el link de Plan de carrera	Se muestra el plan de carrera del empleado los cursos que tiene realizados	OK

	Curriculum- Búsqueda Plan Carrera/Mostrar Plan Carrera Empleado		
Subsistemas Implicados:	Ninguno		
Nº	Entrada	Acción Esperada	**Verificar**
1	Después de logados en el sistema como admin@admin pinchamos en Búsqueda	El sistema nos muestra la búsqueda de candidato.	Ok

2	Seleccionamos el link de Plan de Carrera	Muestra el buscador de planes de carrera.	Ok
3	Se rellenan los datos de lo que buscas y se pulsa buscar	Se muestra el resultado del buscador los empleados que cumplen los filtros de búsqueda	Ok
4	Se pulsa el link de Plan de carrera	Se muestra el plan de carrera del empleado los cursos que tiene realizados	OK

8. Seguimiento del proyecto

En esta fase durante el diseño técnico se nos platearon una serie de dificultades técnicas que se han ido solventando habiendo repercutido en el tiempo que había para esta tarea.
Además hemos encontrado una nueva forma más intuitiva de gestionar el borrado del Curriculum creando una baja lógica en vez de fisica pudiendo el gestor de RRHH realizar una búsqueda de Curriculum añadiendo los usuarios "Inactivos " y modificar su estado por medio de modificar Curriculum/ modificar datos personales y aqui ponerle el estado Activo.
Además la navegación y presentación de algunas pantallas ha variado con respecto a diseño original para adaptarnos a los componentes UI que nos brindaba el framework.



	Nombre de tarea	Duración	Comienzo
1	**Fase 1: Planificación**	**12 días?**	**mié 26/02/14**
2	Propuesta	1 día?	mié 26/02/14
3	Aceptación de propuesta	1 día?	jue 27/02/14
4	Introducción objetivos del proyecto	2 días	lun 03/03/14
5	Metodología y estudio de tareas	2 días	mié 05/03/14
6	Planificación temporal (Gant)	3 días	vie 07/03/14
7	Elaboración documentación	1 día?	mié 12/03/14
8	Entrega PEC1	1 día?	jue 13/03/14
9	**Fase 2: Analisis de requisitos**	**24 días?**	**vie 14/03/14**
10	Identificación de actores	1 día	vie 14/03/14
11	Especificación requisitos por subsistema	7 días	lun 24/03/14
12	Especificación caso de uso	7 días	mié 02/04/14
13	Interfaz de usuario	4 días	vie 11/04/14
14	Entrega PEC2	1 día?	mié 16/04/14
15	**Fase 3: Diseño técnico e implementación**	**37 días?**	**jue 17/04/14**
16	Diseño de la arquitectura de componentes	4 días	jue 17/04/14
17	Diseño de capas y clases	4 días	mié 23/04/14
18	Diseño persistencia	4 días	mar 29/04/14
19	Implementación	14 días	mar 20/05/14
20	Elaboración de documentación	1 día	mié 21/05/14
21	PEC3	1 día?	mié 21/05/14
22	**Memoria y Preguntas**	**15 días?**	**jue 22/05/14**
23	Elaboración Memoria	7 días?	jue 22/05/14
24	Presentación	1 día?	mié 11/06/14
25	**Tribunal**	**5 días?**	**lun 23/06/14**
26	Enunciado	1 día?	lun 23/06/14
27	Entrega de respuestas	1 día?	vie 27/06/14

Figura 27.

10. Conclusión

Elaborar este producto software nos ha permitido pasar por todas las fases del ciclo de vida del software, pudiendo tomar decisiones sobre que metodologías usar, realizar una planificación detallada de todas las tareas a afrontar, hacer una especificación de requerimientos, el análisis de los mismos a través de casos de uso y diseño de interfaces ...

Una de las decisiones que barajó durante el proceso de evaluación de que metodología de desarrollo del software usar se pensó en la posibilidad de usar una metodología incremental , ya que es una de la metodologías más utilizadas porque ofrece la posibilidad de evolucionar el producto con cada una de la iteraciones pudiendo añadir nuevos requisitos y así obtener un producto más completo. En nuestro caso resultaba imposible segmentar el proyecto en iteraciones y nos dimos cuenta que era más natural usar una metodología secuencial por tener los requerimientos claros y cerrados desde una primera instancia.

A lo largo del proceso de análisis de requisitos en un primer lugar se realizaron diseño de interfaces GUI muy simplistas, algunas representaban la funcionalidad de una forma muy poco usable. Después de realizar el análisis técnico se rectificaron pudiendo darles un aspecto más acorde con la apariencia final del producto y se les otorgó una funcionalidad con usabilidad acertada.

Durante el proceso de testing del software nos encontramos con cosas que no funcionaban según lo recogido en los requerimientos y se corrigió la funcionalidad.

Con todo lo que nos hemos ido encontrado, hemos podido ver como se afrontan y solucionan los problemas relacionados con el ciclo de vida del desarrollo de un producto software, hemos asentado todo lo aprendido tanto a nivel teórico como práctico a lo largo de las asignaturas del itinerario de ingeniería del software.

Hemos asimilado que el desarrollo de un proyecto software puede tener distintas caras según el rol que desempeñes dentro del desarrollo del mismo, y nos encontramos con los fundamentos suficientes para afrontar en el futuro con más confianza cualquier reto profesional.

11. Licencia

El tipo de licencia para el software utilizado es Licencia Pública General más conocida por sus siglas en ingles GPL.

12. Anexo

Para correcta implementación del software se necesitan el siguiente software y estas versiones:

- JDK 1.7 versión 1.7.0-10
- Ant, versión 1.8.4
- PostgreSQL versión 9.2.2-1
- Conector Java Postgresql-9.2-1002.jdbc4
- JBoss, versión 7.1.1.Final
- Eclipse, versión Juno SR1
- JBoss Tools 4.0

- En primer lugar debemos instalar el jdk 1.7.
- Crear la variable de entorno JAVA_HOME que su valor que debe contener es la ruta física del jdk. Por ejemplo: *"c:\Archivos de programa\Java\jdk1.7.0_10"*
- Buscamos la variable de entorno PATH y le añadimos *";%JAVA_HOME%\BIN"*
- Buscamos la variable de usuario CLASSPATH y le añadimos *";%JAVA_HOME%\lib\tools.jar;."*
- Instalamos el apache Ant la versión 1.8.4 se descargará un zip lo descomprimimos en C:\
- Creamos la variable de entorno ANT_HOME que contendrá la ruta fisica de donde está instalado Por ejemplo: *"C:\apache-ant-1.9.2"*
- Buscamos la variable de entorno "PATH" y le añadimos al final *";%ANT_HOME%\BIN"*
- Instalamos el PostgreSQL versión 9.2 si SO es win vista ó 7 debemos realizar lo siguiente *Panel de control->Cambiar configuración de control de cuentas de usuario->"no notificar nunca"*
- Instalamos PostgreSQL con todas las opciones por defecto debemos apuntar el puerto de conexión.
- Una vez instalado el PosgreSQL buscamos pgAdmin y nos conectamos al servidor de postgre (Botón derecho del ratón Connect)
- En Login botón derecho del ratón "*New Log Role*" le ponemos en la pestaña de *Properties* el nombre de USUARIO y en la pestaña de *Definition* el PASSWORD.
- En la pestaña *Role privileges* marcar todos los privilegios.
- Creamos un nuevo esquema llamado MAC.
- Instalamos JBoss, para ello nos descargamos el fichero *jboss-as-7.1.1.Final.zip* por ejemplo en C:\
- Creamos la variable de entorno JBOSS_HOME ponemos el valor de la ruta donde hemos descomprimido el fichero. Por ejemplo *C:\jboss-as-7.1.1.Final*
- Reiniciar el sistema.
- Para ejecutar JBoss nos vamos a la carpeta \bin ejecutamos stanalone.bat.
- Para parar el Jboss Ctrl +C en la consola
- Nos bajamos el conector java jdbc de postgreSQL (*postgresql-9.2-1002.jdbc4.jar*) .
- Configuramos el conector jdbc para jboss creamos una estructura de carpetas *\postgresql\main* en la carpeta *JBOSS_HOME\modules\org*, así la estructura debe de ser, por ejemplo: *C:\jboss-as-7.1.1.Final\modules\org\postgresql\main* JBOSS_HOME: debe contener el valor del directorio donde está instalado Jboss. Por ejemplo *C:\jboss-as-7.1.1.Final*

- Copiamos *postgresql-9.2-1002.jdbc4.jar* en la carpeta *C:\jboss-as-7.1.1.Final\modules\org\postgresql\main*
- En la misma carpeta creamos un fichero con el nombre module.xml con el siguiente texto.

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="org.postgresql">
<resources>
<resource-root path="postgresql-9.2-1002.jdbc4.jar"/>
</resources>
<dependencies>
<module name="javax.api"/>
<module name="javax.transaction.api"/>
</dependencies>
</module>
```

- Para definir el datasource añadimos al fichero *standalone.xml* colgando de <datasources> el siguiente texto:
  <datasource jta="false" jndi-name="java:jboss/postgresDS" pool-name="postgresDS"

```
enabled="true" use-java-context="true" use-ccm="false">

<connection-url>jdbc:postgresql://localhost:5432/postgres</connection-url>
<driver-class>org.postgresql.Driver</driver-class>
<driver>postgresql</driver>
<pool/>
<security>
<user-nameUSERser-name>
></u <password>PASSWORD</password>
</security>
<statement/>
</datasource>
```

- Y dentro del mismo fichero colgando de <drives> el siguiente texto:

```
<driver name="postgresql" module="org.postgresql">

<xa-datasource-class>org.postgresql.xa.PGXADataSource</xa-datasource-class>

</driver>
```

- Y el ear de la aplicación se colocará en la carpeta: C:\jboss-as-7.1.1.Final\standalone\deployments
- Una vez arrancado Jboss y con la base de datos arrancada podremos acceder a la aplicación a través del link: http:\\localhost:8080\MAC

- Una vez desplegado el ear se pasará el script situado en esta ruta: MAC\doc\Scritps\ NivelProf-Categorias-Admin-ConocimientoCurso.sql
- **La entrada a la plataforma** después de la instalación se hará con el usuario **admin@admin** y **la contraseña de todos los usuarios** en esta primera fase es **1234.**

13. Bibliografía

Pradel, J.; Raya, J. (2011) "Introducción a la ingeniería de los requisitos", "Documentación de requisitos" En: *Ingeniería de los requisitos*.

Rodríguez, J.R; Pere, J. (2011) "Planificación del Proyecto" En: *Gestión de proyectos*

Pradel, J; Raya, J. (2011) "Requisitos" En: *Ingeniería del software.*

Rodríguez, J.R; Pere, J. (2011) "Planificación del Proyecto" En: *Gestión de proyectos*

Moreno, N.; Vallecillo, A.; Romero J.R.; Durán F.J. (2013) "Arquitectura del software" En: Ingeniería Sofware Componentes Sistemas Distribuidos

Reinoso, C.; Kicillof, N. "*Estilos y Patrones en la estrategia de Arquitectura de Microsoft*" [artículo en línea] http://www.slideshare.net/bjjuarez/estilos-de-software

Robertson, J.; Robertson, S. "*Volere Requirements Specification Template*" [artículo en línea] http://www.volere.co.uk/template.htm

Vallecillo, A; Romero, J.R; Moreno, A; Durán, F.J. (2011) "Diseño de aplicaciones distribuidas" En: *Ingeniería del software de componentes y sistemas distribuidos.*

Camps, J.M. (2011) "Java EE, Una plataforma de componentes distribuida" En: *Ingeniería del software de componentes y sistemas distribuidos.*

Escudero, X. (2011) "Calidad del software: técnicas de prevención, detención y corrección de defectos" En: Proyecto de desarrollo del software.