

2. Documento de objetivos del proyecto.

2.1 Descripción del proyecto.

2.1.1 Descripción de lo que se llevará a cabo.

2.1.2 Objetivos del proyecto.

2.2 Planificación.

2.2.1 Recursos: personales y materiales.

2.2.2 Método de trabajo.

2.2.3 Alcance del proyecto.

2.2.4 Entregables.

2.2.5 Fases y tareas: Estimación de los tiempos.

2.2.6 Asignación de recursos y agenda del proyecto (D. Gantt)

2.3 Riesgos

2.3.1 Identificación de riesgos.

2.4 Costos

2.4.1 Estimación de costos: presupuesto.

2. Documento de objetivos del proyecto.

2.1 Descripción del proyecto.

2.1.1 Descripción de lo que se llevará a cabo.

Como se dijo anteriormente en la introducción de la memoria el proyecto consiste en realizar la migración de una aplicación ya existente implementada en Visual Basic 6.0 que se ejecuta en modo local, a una nueva aplicación hecha completamente en Java con una estructura multi-nivel de tres capas: capa de presentación, capa de negocio y capa lógica, tal y como se muestra en la figura 1.

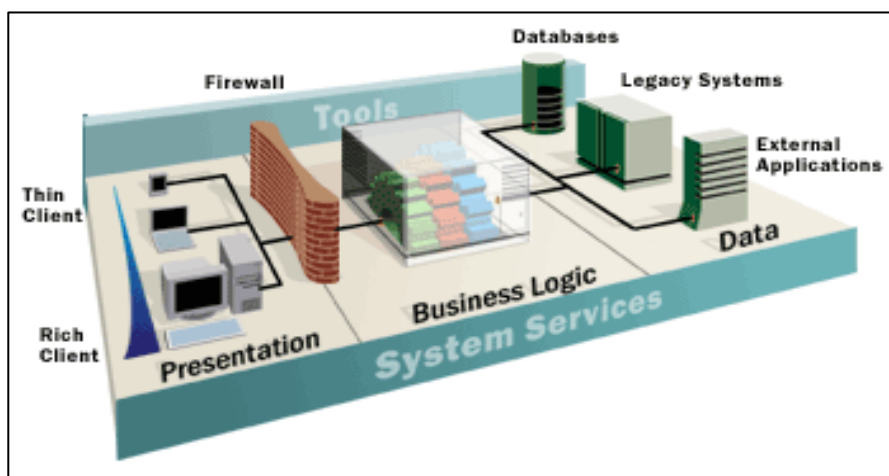


Fig. 1: Arquitectura de tres capas.

Para lograr esta estructura se usarán los Enterprise Java Beans (EJB), uno de los métodos de desarrollo de aplicaciones empresariales de mayor éxito en el mundo. Aunque este tema se explicará con detalle más adelante, estos EJBs permiten construir componentes (unidad de software a nivel de aplicación), ya sean para guardar información o para realizar métodos de negocio, con características de encapsulamiento y reutilización.

La aplicación se desarrollará íntegramente con la herramienta de desarrollo software creada por ORACLE **JDeveloper 9.0.5**. Esta herramienta, tal y como se verá más adelante, permite el desarrollo de sistemas distribuidos con una mayor facilidad, tanto para la parte de presentación como en el resto de capas.

Por tanto, el nuevo sistema constará de una parte cliente, que estará alojada en cada máquina de usuario, donde se implementarán las clases necesarias para la comunicación con el servidor de aplicaciones y procesar los datos obtenidos para su comprensión por el usuario (GUI), y un servidor donde se alojarán los EJBs (por ejemplo, JBoss).

2.1.2 Objetivos del proyecto.

El objetivo del proyecto a nivel genérico es la elaboración de una aplicación cliente-servidor que dé respuesta a los requerimientos del usuario, en este caso la empresa deportiva 'Deportes LSI 2003'.

La aplicación obtenida debe ser desde el punto de percepción del usuario final, similar a la desarrollada por los alumnos de la Universidad Politécnica de Valencia. De este modo, se migrará desde una aplicación desarrollada en Visual Basic 6.0 que funciona localmente y que por tanto la mayor "carga" de software está en la máquina cliente, a una aplicación distribuida hecha completamente en Java, pudiendo de este modo acceder al sistema desde cualquier plataforma y repartir a su vez la "carga" de software entre la máquina cliente y el servidor de aplicaciones.

La tecnología a utilizar en cuanto a cómo desarrollar la aplicación se dejó a la elección del desarrollador del proyecto. Se eligió desarrollar la parte servidor a través de una tecnología basada en componentes como son los EJBs, mientras que la parte cliente se desarrolla no como una aplicación Web (html) sino a través de objetos GUI (interfaz de usuario gráfica) utilizando para ello los objetos del paquete estándar *javax.swing*.

Por tanto, en un principio se debe efectuar un estudio exhaustivo de la aplicación ya existente, así como de los requerimientos de usuario. El estudio del funcionamiento del programa se hace a base de pruebas y experiencia en el manejo de éste por parte del desarrollador del presente proyecto, ya que no existe manual de usuario. En cuanto a la especificación de requisitos sí está disponible.

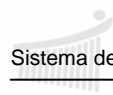
El sistema (en el nivel de iteración que se exige en el proyecto) tendrá tres tipos de usuario: técnico de almacén, representante y operador/a.

El primero de ellos tendrá acceso a asignar stock a los diferentes pedidos, cambiarlos de estado dentro de la estructura organizativa, cancelarlos, etc. El representante tendrá acceso a crear, modificar, cancelar los pedidos que pertenezcan a los clientes que tenga asignados, así como a consultar los pedidos enviados a almacén de éstos. La operadora tendrá los mismos privilegios que el representante, pero podrá acceder a todos los clientes de la base de datos.

La aplicación se desarrollará íntegramente con la herramienta JDeveloper 9.0.5, tanto la parte cliente como la parte servidor (EJBs).

Una vez creada, se deberán empaquetar y trasladar los EJBs a un servidor de aplicaciones que posea las especificaciones J2EE. En este caso se utilizará una herramienta Open-source llamada JBoss.

La base de datos en un principio se creará en Microsoft Access, y estará alojada en la misma máquina donde se encuentre el servidor.



Este sistema permitirá a la empresa informatizar el control de todas sus actividades (gestión de stock en cada almacén, gestión de pedidos, etc.), lo que supondrá un acceso rápido y sencillo a los datos, gracias a interfaces gráficas simples y amigables. Además, los datos accedidos estarán siempre actualizados, lo que es un factor muy importante para poder llevar un control centralizado de los distintos almacenes.

2.2 Planificación.

2.2.1 Recursos personales y materiales.

Como recursos materiales para realizar el proyecto se dispone de un ordenador personal Pentium II 350 MHz, 196 MB de memoria RAM así como una capacidad de almacenaje de 14 GB y sistema operativo Windows 2000 Pro. En esta máquina será donde se realizará casi en su totalidad el desarrollo software del proyecto. Cabe decir que las características técnicas del ordenador son algo limitadas para el desarrollo de este tipo de aplicaciones, y más aún si se utiliza la herramienta JDeveloper, ya que consume muchos recursos de memoria virtual. Por tanto se recomienda utilizar al menos una máquina con estas características: Pentium III 500 MHz y 256 MB RAM.

Se dispone también del aula de informática 1.4 en la Escuela Universitaria de Ingeniería de Vitoria-Gasteiz desde el 16 de julio de 2004 hasta el 30 de septiembre del mismo año. El aula se compone de unas 25 máquinas con sistema operativo Windows 2000 Pro conectadas en red local y acceso a la red de la Universidad del País Vasco. Sus características técnicas superan con creces las mínimas para el desarrollo de este tipo de aplicaciones.

Como recursos personales se dispone del desarrollador del proyecto David Montejo Blanco, estudiante de Ingeniería Técnica en Informática de Gestión en la Escuela Universitaria de Ingeniería de Vitoria-Gasteiz. Se dispone también de la colaboración de la directora del proyecto Mari Carmen Otero Vidal.

2.2.2 Método de trabajo.

El seguimiento y control del proyecto reúne los siguientes aspectos:

Gestión de Requisitos

Los requisitos del sistema son descritos según las especificaciones de los casos de uso que han sido creados al comienzo del proyecto. A la hora de desarrollar el sistema, y haciendo uso de los diferentes prototipos de la aplicación que vayan surgiendo, se harán una serie de comparativas entre las especificaciones y los módulos creados. De este modo se obtendrá un producto final de calidad que satisfaga realmente las pautas marcadas por el usuario.

Control de Plazos

El calendario del proyecto tendrá un seguimiento y evaluación semanal por el desarrollador del proyecto. Se procurará seguir el calendario establecido para cada una de las fases y tareas marcadas.

En caso de que se produzca algún retraso o replanificación se deberá añadir según convenga al diagrama de Gantt de planificación, que se muestra en el punto 2.2.6 del presente documento.

2.2.3 Alcance del proyecto: EDT.

EDT o Estructura desagregada del trabajo sirve para separar el proyecto en un conjunto de componentes principales que a su vez se aíslan en componentes cada vez más pequeños. Por tanto la planificación del proyecto quedaría del siguiente modo (figura 2):

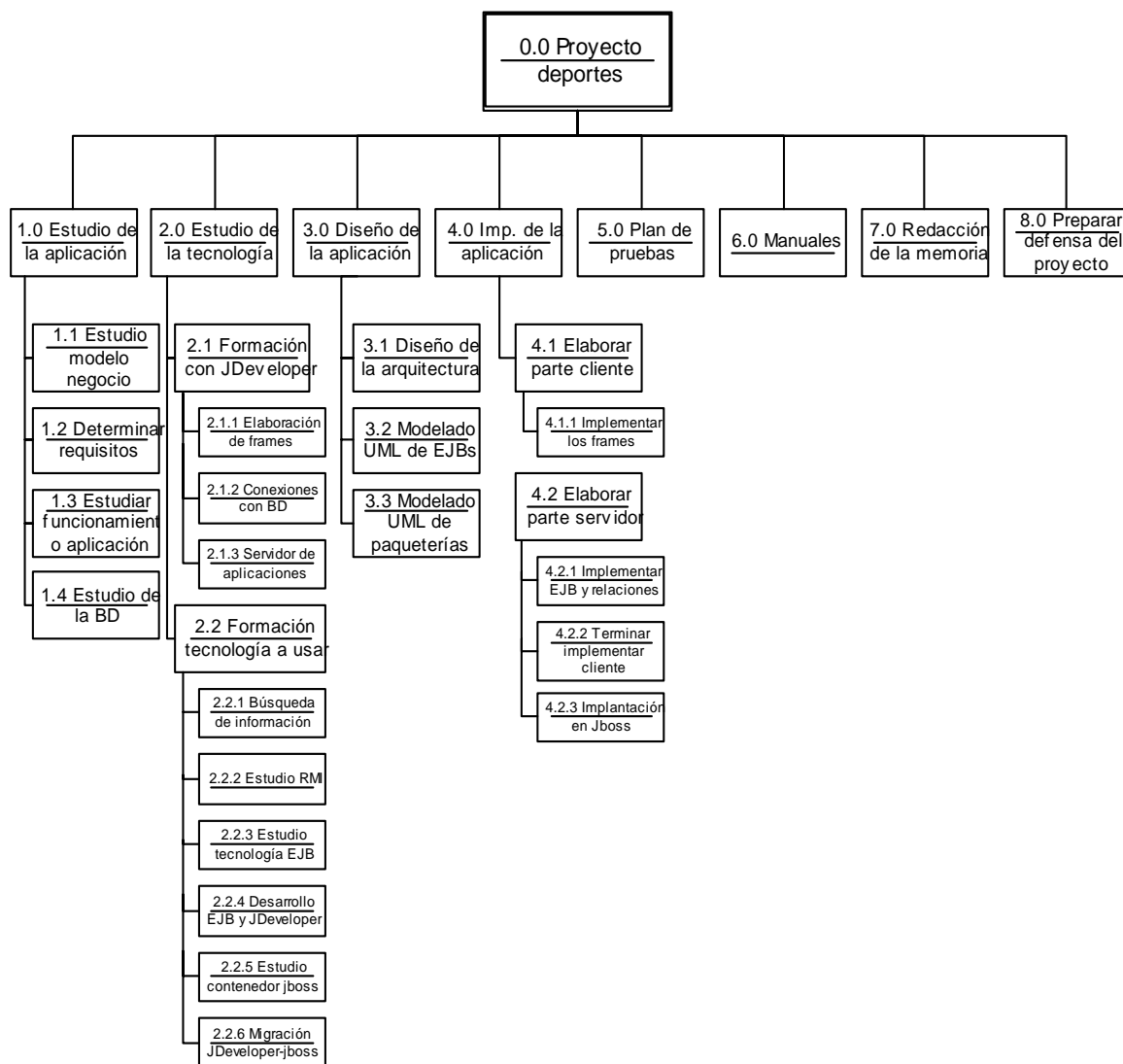


Fig. 2: EDT del proyecto.

2.2.4 Entregables.

A continuación se indican y describen los diferentes documentos generados y utilizados para el proyecto que constituyen los entregables.

1. **Modelado de casos de uso:** El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso. En este caso, debido a que los casos de uso ya están desarrollados se procederá

solamente a su modificación y adecuación a las nuevas especificaciones.

2. **Especificaciones de los casos de uso:** Para los casos de uso que lo requieran se realiza una descripción detallada donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados. Como las especificaciones ya están desarrolladas se procederá solamente a su modificación y adecuación a las nuevas especificaciones.
3. **Modelo de datos:** La base de datos se representará a través de su modelo entidad-relación.
4. **Modelo de análisis y diseño:** Este entregable incluye la representación de las clases que se van a utilizar en UML (*Unified Modelation Language*). También se incluirán la representación de los Enterprise Java Beans así como de las paqueterías que conformarán la aplicación.
5. **Modelo de implementación:** Se trata de entregar las diferentes clases que se van creando a lo largo del proceso de implementación en sus diferentes fases. En cierto modo se trata de prototipos de la aplicación final.
6. **Plan de pruebas y resultados:** Se trata de entregar las pruebas sobre el software creado y sus resultados. Así, se podrá determinar la fiabilidad o calidad de éste y en consecuencia llevar a cabo las acciones necesarias para mejorar estos dos aspectos.
7. **Plan de iteración:** Se trata de realizar la planificación del proyecto, es decir, la definición de sus fases y tareas, recursos asignados, dependencias y su orden de elaboración. Se realizará también un diagrama de Gantt.
8. **Evaluación de la iteración:** Se trata de realizar una comparativa entre lo planificado en un principio y lo que realmente ha sucedido a lo largo del desarrollo del proyecto. Es decir, analizar los retrasos y replanificaciones, comparativa de horas planificadas y reales, conclusiones, etc.
9. **Manuales:** Se entregarán tanto el manual de usuario, donde se explicará el funcionamiento de la aplicación a nivel de usuario, y manual técnico, donde se explicará cómo instalar la aplicación, su despliegue en el servidor de aplicaciones, etc.
10. **Producto:** Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación. Se entregará documentación, clases java, programas necesarios para el despliegue de la aplicación, etc.

2.2.5 Fases y tareas. Estimación de tiempos.

Listado de las fases y tareas del proyecto:

- 1 Estudio de la aplicación**
 - 1.1 Estudio del modelo de negocio
 - 1.2 Determinar requisitos de la aplicación
 - 1.3 Estudiar el funcionamiento de la aplicación
 - 1.4 Estudio de la base de datos
- 2 Estudio de la tecnología a utilizar**
 - 2.1 Formación con JDeveloper**
 - 2.1.1 Elaboración de frames
 - 2.1.2 Conexiones con base de datos
 - 2.1.3 Estudio de su servidor de aplicaciones
 - 2.2 Formación de la tecnología a utilizar**
 - 2.2.1 Búsqueda de información
 - 2.2.2 Estudio de RMI (Remote Method Invocation)
 - 2.2.3 Estudio de la tecnología EJB
 - 2.2.4 Desarrollo de EJB con Jdeveloper
 - 2.2.5 Estudio contenedor J2EE Jboss
 - 2.2.6 Migración de Jdeveloper a Jboss
- 3 Diseño de la aplicación**
 - 3.1 Diseño de la Arquitectura
 - 3.2 Modelado UML de los EJB
 - 3.3 Modelado UML de las paqueterías
- 4 Implementación de la aplicación**
 - 4.1 Elaborar parte cliente (Presentación)**
 - 4.1.1 Implementar los frames(ventanas)
 - 4.2 Elaborar parte servidor (parte lógica y negocio)**
 - 4.2.1 Implementar EJB y sus relaciones
 - 4.2.2 Terminar de implementar parte cliente
 - 4.2.3 Implantación en JBoss
- 5 Pruebas de la aplicación**
- 6 Manuales**
- 7 Redacción de la memoria**
- 8 Preparación de la defensa del proyecto**

A continuación se generará una ficha para cada tarea identificada, donde se anotará su número, nombre, una breve descripción y el esfuerzo estimado:

Especificación de la tarea:

Número: 1.1

Nombre: Estudio del modelo de negocio

Descripción: Determinar en qué estructura organizativa se va a desenvolver el sistema a desarrollar y tener una visión más detallada de sus necesidades reales.

Esfuerzo Estimado: 1 día

Especificación de la tarea:

Número: 1.2

Nombre: Determinar requisitos de la aplicación.

Descripción: Determinar cuáles son los deseos de la empresa deportiva en cuanto a las funciones que debe desempeñar la aplicación a obtener. Esta tarea se centra principalmente en el estudio del documento donde vienen especificados los casos de uso, así los diagramas que los representan.

Esfuerzo Estimado: 3 días

Especificación de la tarea:

Número: 1.3

Nombre: Estudiar el funcionamiento de la aplicación.

Descripción: En esta tarea se trata de conocer a fondo el funcionamiento de la aplicación implementada en Visual Basic 6.0 ya existente y sacar patrones de diseño para su desarrollo en Java.

Esfuerzo Estimado: 5 días

Especificación de la tarea:

Número: 1.4

Nombre: Estudio de la base de datos.

Descripción: Se hará un estudio en profundidad del diseño de la base de datos. Esta tarea es importante, ya que a la hora de desarrollar los EJBs es fundamental tener un conocimiento profundo de la base de datos.

Esfuerzo Estimado: 1 día

Entregables: Modelo entidad-relación

Especificación de la tarea:

Número: 2.1.1

Nombre: Elaboración de frames.

Descripción: Se trata de realizar un estudio sobre la elaboración de Frames (ventanas) dentro del entorno de desarrollo JDeveloper. Esto se llevará a cabo haciendo pequeñas aplicaciones donde se utilizarán los distintos componentes que se utilizan en la elaboración de GUI en Java. En concreto, se utilizarán los objetos definidos en el paquete *javax.swing*.

Esfuerzo Estimado: 3 días

Especificación de la tarea:

Número: 2.1.2

Nombre: Conexiones con bases de datos.

Descripción: Determinar cómo desde la herramienta JDeveloper se puede acceder a la estructura y contenido de una base de datos cualquiera.

Esfuerzo Estimado: 1 día

Especificación de la tarea:

Número: 2.1.3

Nombre: Estudio de su servidor de aplicaciones.

Descripción: Se trata de tener una idea básica del funcionamiento del servidor de aplicaciones que se incluye en la herramienta JDeveloper.

Esfuerzo Estimado: 2 días

Especificación de la tarea:

Número: 2.2.1

Nombre: Búsqueda de información.

Descripción: Se produce la búsqueda de información para la elaboración de la aplicación para así determinar cuál va a ser la tecnología a utilizar y la que mejor se adecue a las especificaciones del usuario.

Esfuerzo Estimado: 5 días

Especificación de la tarea:

Número: 2.2.2

Nombre: Estudio de RMI (Remote Method Invocation).

Descripción: Una vez determinada la tecnología a utilizar se procederá a estudiar los fundamentos en los que se basa. En este caso la tecnología seleccionada fueron los EJBs, que se basa en la invocación a métodos de objetos distribuidos o remotos.

Esfuerzo Estimado: 2 días

Especificación de la tarea:

Número: 2.2.3

Nombre: Estudio de la tecnología EJB.

Descripción: Una vez recopilada la información necesaria, se lleva a cabo la familiarización con la tecnología a desarrollar y la realización de pequeñas aplicaciones como prueba.

Esfuerzo Estimado: 5 días

Especificación de la tarea:

Número: 2.2.4

Nombre: Desarrollo EJB con JDeveloper.

Descripción: Se procede a estudiar cómo realizar y desplegar EJBs dentro de JDeveloper.

Esfuerzo Estimado: 2 días

Especificación de la tarea:

Número: 2.2.5

Nombre: Estudio del contenedor J2EE Jboss.

Descripción: Al realizar la aplicación con JDeveloper, ésta deberá ser probada dentro de otro servidor de aplicaciones (en este caso Jboss). Por tanto se procederá a su estudio.

Esfuerzo Estimado: 3 días

Especificación de la tarea:

Número: 2.2.6

Nombre: Migración de JDeveloper a Jboss.

Descripción: Determinar cómo realizar el paso de la aplicación una vez acabada a su entorno "real".

Esfuerzo Estimado: 3 días

Especificación de la tarea:

Número: 3.1

Nombre: Diseño de la arquitectura.

Descripción: Se trata de determinar cuál será el modo en que estarán distribuidos los diferentes tipos de EJBs dentro del servidor de aplicaciones así como su relación con la base de datos.

Esfuerzo Estimado: 2 días

Especificación de la tarea:

Número: 3.2

Nombre: Modelado UML de los EJB.

Descripción: Crear los diagramas necesarios para el diseño de los Enterprise Java Beans.

Esfuerzo Estimado: 2 días

Entregables: Diagramas correspondientes.

Especificación de la tarea:

Número: 3.3

Nombre: Modelado UML de las paqueterías.

Descripción: Se trata de diseñar a través de la elaboración de diagramas en UML la distribución de clases, tanto en la parte servidor como en la parte cliente a través de sus paqueterías.

Esfuerzo Estimado: 1 día

Entregables: Diagramas correspondientes.

Especificación de la tarea:

Número: 4.1.1

Nombre: Implementar los frames (ventanas)

Descripción: Implementación de la interfaz de usuario a semejanza de la aplicación realizada en Visual Basic 6.0.

Esfuerzo Estimado: 10 días

Entregables: Clases Java correspondientes.

Especificación de la tarea:

Número: 4.2.1

Nombre: Implementar los EJBs y sus relaciones.

Descripción: Implementación de los distintos EJBs necesarios, así como de sus métodos de negocio y sus relaciones.

Esfuerzo Estimado: 15 días

Entregables: Clases Java correspondientes.

Especificación de la tarea:

Número: 4.2.2

Nombre: Terminar de implementar la parte cliente.

Descripción: Una vez desarrollados los EJBs, se procederá a implementar la parte cliente para que pueda acceder a los métodos de negocio de los beans de Sesión, así como al procesamiento de los datos obtenidos y su visualización por pantalla.

Esfuerzo Estimado: 1 día

Entregables: Clases Java correspondientes.

Especificación de la tarea:

Número: 4.2.3

Nombre: Implantación en Jboss.

Descripción: Hacer los cambios y crear los ficheros necesarios para que la aplicación funcione bajo Jboss.

Esfuerzo Estimado: 2 días

Especificación de la tarea:

Número: 5

Nombre: Pruebas de la aplicación.

Descripción: Llevar a cabo el plan de pruebas para determinar si la aplicación es fiable o no y determinar los posibles fallos del sistema.

Esfuerzo Estimado: 3 días

Entregables: Plan de pruebas y sus resultados.

Especificación de la tarea:

Número: 6

Nombre: Manuales.

Descripción: Realización de los manuales de usuario y técnico sobre la aplicación.

Esfuerzo Estimado: 3 días

Entregables: Manual de usuario y técnico.

Especificación de la tarea:

Número: 7

Nombre: Redacción de la memoria.

Descripción: Realización de la memoria del proyecto.

Esfuerzo Estimado: 5 días

Entregables: Memoria del proyecto.

Especificación de la tarea:

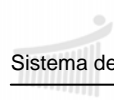
Número: 8

Nombre: Preparación de la defensa del proyecto.

Descripción: Hacer una presentación del proyecto y preparar su exposición y preguntas eventuales.

Esfuerzo Estimado: 7 días

Entregables: Presentación del proyecto en Microsoft Power Point.



2.2.6 Asignación de recursos y agenda del proyecto (Gantt).

(Aquí falta el Diagrama de Gantt de la planificación inicial)

A continuación se muestran los recursos asignados a cada tarea:

Desarrollador	746h
Estudio del modelo de negocio	2h
Determinar requisitos de la aplicación	6h
Estudiar el funcionamiento de la aplicación	40h
Estudio de la base de datos	2h
Elaboración de frames	24h
Conexiones con base de datos	8h
Estudio de su servidor de aplicaciones	16h
Búsqueda de información	40h
Estudio de RMI (Remote Method Invocation)	16h
Estudio de la tecnología EJB	30h
Desarrollo de EJB con Jdeveloper	4h
Estudio contenedor J2EE Jboss	24h
Migración de Jdeveloper a Jboss	24h
Diseño de la Arquitectura	16h
Modelado UML de los EJB	12h
Modelado UML de las paqueterías	2h
Implementar los frames(ventanas)	80h
Implementar EJB y sus relaciones	120h
Terminar de implementar parte cliente	120h
Implantación en Jboss	16h
Pruebas de la aplicación	24h
Manuales	24h
Redacción de la memoria	40h
Preparar defensa del proyecto	56h
Ordenador Personal	19
Estudiar el funcionamiento de la aplicación	1
Estudio de la base de datos	1
Elaboración de frames	1
Conexiones con base de datos	1
Estudio de su servidor de aplicaciones	1
Búsqueda de información	1
Estudio de RMI (Remote Method Invocation)	1
Estudio de la tecnología EJB	1
Desarrollo de EJB con Jdeveloper	1
Estudio contenedor J2EE Jboss	1
Modelado UML de los EJB	0.5
Modelado UML de las paqueterías	0.5
Implementar los frames(ventanas)	1
Implementar EJB y sus relaciones	1
Terminar de implementar parte cliente	1
Implantación en Jboss	1
Pruebas de la aplicación	1
Manuales	1
Redacción de la memoria	1
Preparar defensa del proyecto	1
Aula de Informática 1.4	2
Pruebas de la aplicación	1
Preparar defensa del proyecto	1

2.3 Riesgos.

El objetivo de este documento es el de realizar un análisis exhaustivo de los riesgos que pueden aparecer a lo largo del proceso de desarrollo del proyecto. Se pretende identificar los distintos riesgos potenciales del proyecto y presentar planes para reducir su probabilidad de ocurrencia (estrategia de prevención), para minimizar el impacto del riesgo sobre el producto (estrategia de minimización), y para tomar las medidas pertinentes en caso de que el riesgo haya ocurrido (plan de contingencia). Existen tres tipos de riesgos: de planificación, que afectan a la temporización o recursos del proyecto, de producto, que afectan a la calidad o rendimiento del software en desarrollo y riesgos de negocio, que afectan a la organización que suministra o desarrolla el software¹.

2.3.1 Identificación de riesgos.

Riesgos de proyecto:

1. **Dedicación no exclusiva al proyecto:** En determinadas fases del proyecto el desarrollador realiza otras actividades: asignaturas, cursos, etc. y eso provoca la dedicación parcial no exclusiva al proyecto, lo que puede dar lugar a problemas en cuanto al retraso en la realización de algunas actividades.

Probabilidad: Alta.

Efecto: Tolerable.

Estrategia de prevención: Respetar los plazos planificados con la mayor exactitud posible.

Estrategia de minimización:-

Plan de contingencia: Replanificar el proyecto o aumentar la carga de trabajo al desarrollador.

2. **Bajo número de miembros en el equipo:** El proyecto sólo cuenta con el desarrollador y el director del proyecto. Esta poca variedad de opiniones puede dar lugar a una toma de decisiones equivocada.

Probabilidad: Baja.

Efecto: Tolerable.

Estrategia de prevención: Hacer numerosas reuniones desarrollador-director para concretar o aclarar temas.

Estrategia de minimización:-

Plan de contingencia: Revisar objetivos del proyecto y hacer una nueva reunión.

3. **Cambios en los requisitos:** Introducir nuevos requisitos cuando el proyecto está en su fase de desarrollo.

¹ Este tipo de riesgo no se tendrá en cuenta en el presente proyecto, ya que no se tiene ningún tipo de compromiso ni contrato con empresa alguna.

Probabilidad: Alta.

Efecto: Serio.

Estrategia de prevención: Entrevistas lo más detalladas posibles con el director del proyecto.

Estrategia de minimización: Diseño flexible de la aplicación.

Plan de contingencia: Modificación de la planificación.

- 4. Baja temporal de algún miembro:** Tanto si el director del proyecto como el desarrollador sufre una baja, esto tendrá efectos seguros sobre la planificación del proyecto.

Probabilidad: Baja.

Efecto: Serio.

Estrategia de prevención: No hay prevención posible debido a que la existencia de accidentes es imprevisible.

Estrategia de minimización:-

Plan de contingencia: Replanificar el proyecto y hacer las tareas que no precisen de la colaboración del director y desarrollador conjuntamente.

- 5. Retrasos en la entrega:** Debido a una mala planificación, o a la aparición de algún otro riesgo, se pueden producir retrasos.

Probabilidad: Baja.

Efecto: Serio.

Estrategia de prevención: Realizar una buena planificación.

Estrategia de minimización: Trabajar más horas de las planificadas.

Plan de contingencia: Renegociar los plazos de entrega del proyecto.

- 6. Personal poco cualificado:** El desarrollador se encuentra con una tecnología desconocida y el estudio de ésta se puede alargar, produciendo retrasos en lo planificado.

Probabilidad: Alta.

Efecto: Serio.

Estrategia de prevención: Recopilar numerosa información sobre la tecnología a utilizar.

Estrategia de minimización: Realización gradual de ejemplos con la tecnología para obtener un cierto nivel de conocimientos antes de proceder a realizar el programa final.

Plan de contingencia: Replanificar el proyecto y aumentar los plazos en el estudio de la tecnología.

Riesgos del producto:

1. **Recursos hardware limitados:** Se dispone para la mayor parte de la realización del proyecto de un ordenador personal que no llega a cubrir los requisitos marcados por la herramienta JDeveloper.

Probabilidad: Media.

Efecto: Serio.

Estrategia de prevención: -.

Estrategia de minimización:-.

Plan de contingencia: En el caso en que el desarrollo de la aplicación se haga insostenible, acudir al aula 1.4 para su desarrollo.

2. **Mala elección de la tecnología:** La tecnología no es capaz de abordar todas las especificaciones marcadas al inicio del proyecto.

Probabilidad: Baja.

Efecto: Muy serio.

Estrategia de prevención: Realizar un estudio exhaustivo de las especificaciones y buscar la información necesaria para determinar que la tecnología seleccionada es la correcta.

Estrategia de minimización: Cambio de los requisitos.

Plan de contingencia: Renegociar plazos de entrega.

3. **Bajo rendimiento de la base de datos:** Debido al SGBD utilizado o al diseño de las bases de datos, puede que éstas tengan un rendimiento inferior al deseable.

Probabilidad: Media.

Efecto: Tolerable.

Estrategia de prevención: Realizar un buen diseño de la base de datos.

Estrategia de minimización:-.

Plan de contingencia: Hacer los cambios oportunos sobre la base de datos o cambiar de SGBD.

4. **Mala elección de la herramienta de desarrollo:** La elección de una herramienta de desarrollo de uso complicado o que no de los resultados esperados puede influir en la calidad del producto final.

Probabilidad: Baja.

Efecto: Tolerable.

Estrategia de prevención: Barajar varias hipótesis (herramientas) y elegir la mejor.

Estrategia de minimización: Cambiar requisitos.

Plan de contingencia: Profundizar en el estudio de la herramienta y buscar nuevas soluciones.

2.4 Costos.

2.4.1 Estimación de costos.

A continuación se muestra en detalle el conjunto de tareas planificadas para el proyecto junto a su costo. Para realizar la estimación de costos se toman como referencia estos dos recursos: un desarrollador que cobra 10€/hora como tasa estándar y 12€/hora como tasa de horas extra y un ordenador personal al que se le asigna un costo por uso de 1€, es decir, cada vez que se utiliza se cobra un euro, en concepto de luz y amortizaciones. En la realización de este documento no se tiene en cuenta el margen de beneficios para el proyecto ni el pago de licencias software (ver figura 4).

(Aquí falta la tabla de estimación de costos de la planificación inicial)

Por tanto, el costo estimado del proyecto es de **7480€**