

BIKERENT

PLAN DE PROYECTO

Armero Guerra, María
Jiménez Soria, Alberto
Martin Martínez, Javier
Muñoz de la Peña Herranz, Laura
de Pablos Blanco, Guillermo
Sánchez Castellanos, Isaac
Sánchez Jiménez, Jose Luis
Sánchez Sánchez, Adrian
Suarez Solorzano, Sergio
Valenciano López, Javier

2012

INDICE

1.	INTRODUCCION	4
1.1.	Propósito del plan	4
1.2.	Ámbito del proyecto	4
1.2.1.	Declaración del ámbito	4
1.2.2.	Funciones Principales	4
1.2.3.	Aspectos de rendimiento	5
1.2.4.	Restricciones y técnicas de gestión	5
1.3.	Modelo de proceso	5
2.	ESTIMACIONES DEL PROYECTO	6
2.1.	Datos históricos	6
2.2.	Técnicas de estimación	6
2.3.	Estimaciones de coste, esfuerzo y duración	7
3.	ESTRATEGIA DE GESTIÓN DE RIESGOS	12
3.1.	Análisis del riesgo	12
3.2.	Estudio de los riesgos	14
3.3.	Plan de gestión del riesgo	16
3.3.1.	Primer Riesgo: Subestimar el tiempo de desarrollo	16
3.3.2.	Segundo Riesgo: Incompatibilidad de versiones de desarrollo	16
3.3.3.	Tercer Riesgo: Componentes Defectuosos	17
4.	PLANIFICACIÓN TEMPORAL	18
4.1.	Estructura de descomposición del trabajo	18
4.2.	Gráfico Gantt	19
4.3.	Red de tareas	19
4.4.	Tabla de uso de recursos	19
5.	RECURSOS DEL PROYECTO	20
5.1.	Personal	20
5.2.	Hardware y Software	21

6.	ORGANIZACIÓN DEL PERSONAL.....	22
6.1.	Estructura de equipo.....	22
6.2.	Informes de gestión.....	22
7.	MECANISMOS DE SEGUIMIENTO Y CONTROL.....	24
7.1.	Garantía de calidad y control	24
7.2.	Gestión y Control de cambios.....	25

1. INTRODUCCION

1.1. Propósito del plan

El objetivo del plan es establecer los criterios a seguir durante el desarrollo del proyecto junto con una planificación temporal, y una estimación de costes.

Se reflejara también una estrategia de gestión de los posibles riesgos que puedan suceder, así como la organización del personal que se encargue de la gestión y desarrollo del proyecto.

Para el presente plan de proyecto, seguiremos el modelo de Pressman, que se centra en una declaración general de qué y una declaración específica de cuánto y cómo.

1.2. Ámbito del proyecto

1.2.1. Declaración del ámbito

El cliente que solicita el servicio, es una empresa que se dedica al alquiler de bicicletas en diferentes ciudades.

La empresa dispone de un gran número de trabajadores que se encargan de llevar a cabo el alquiler, la atención a clientes y el buen estado de las bicicletas.

El usuario o usuarios principales de la aplicación serán empleados de la propia empresa dedicados expresamente a ello. La aplicación será por tanto una aplicación de escritorio.

1.2.2. Funciones Principales

La aplicación se encargará de organizar todo lo relacionado con el alquiler de bicicletas. Se podrá realizar el alta, baja modificación o consulta de cualquier bicicleta así como de los clientes de la empresa, incidencias y ofertas.

El usuario será el empleado de la empresa y el único que podrá acceder al sistema.

El usuario será el que realice todo tipo de operaciones dentro de la aplicación.

1.2.3. Aspectos de rendimiento

La aplicación no estará activa las 24h del día, ya que no se trata de una aplicación web, sino de una aplicación de escritorio de gestión, por tanto su uso estará disponible únicamente en las horas laborables de la empresa.

La disponibilidad no es, por tanto, un punto crítico para la aplicación.

También se deberá tener en cuenta el número de socios o clientes, y el de bicicletas que podrán estar dados de alta en el sistema.

1.2.4. Restricciones y técnicas de gestión

Los usuarios interactuarán con la aplicación a través de un modulo de escritorio, los ordenadores irán equipados con Windows 7, por lo tanto la aplicación ha de ser compatible con este sistema operativo.

El usuario ha impuesto las restricciones del uso del modelo en espiral.

El lenguaje para caracterizar el diseño de la aplicación debe ser UML 2 y para ello utilizaremos la aplicación Bouml (versión 4.23).

Habrà una primera entrega al usuario el día 25 de Noviembre, y una segunda entrega que consistirá en el producto final a finales de Enero de 2012.

El lenguaje de implementación será C++.

La persistencia de los datos se realizará en formato de archivos de texto.

1.3. Modelo de proceso

La realización del proyecto se ajustará al modelo de Boston. Es un modelo en espiral que caracteriza la vida de un sistema y se centra en los siguientes puntos:

- **Comunicación con el cliente**
Tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- **Planificación**
Tareas requeridas para definir recursos, tiempo y otras informaciones relacionadas con el proyecto
- **Análisis de riesgos**
Tareas requeridas para evaluar riesgos técnicos y de gestión
- **Ingeniería**
Tareas requeridas para construir una o más representaciones de la Aplicación.
- **Construcción y adaptación**
Tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.
- **Evaluación por el cliente**
Tareas requeridas para obtener la reacción del cliente tras su evaluación.
- **Actividades de soporte**

2. ESTIMACIONES DEL PROYECTO

2.1. Datos históricos

Dado que somos una empresa joven, no se dispone de información histórica. Por lo que la productividad de la empresa será de 10 pm, ya que somos 10 personas en el equipo.

2.2. Técnicas de estimación

Vamos a utilizar la técnica de estimación por puntos función.

La métrica del punto función, definida por Allan Albrecht, de IBM, en 1979, es un método utilizado en ingeniería del software. Este método emplea datos históricos y se usa para:

- ✓ Estimar el costo o el esfuerzo requerido para diseñar, codificar y probar el software.
- ✓ Predecir el número de errores que se encontraran durante la prueba.
- ✓ Pronosticar el número de componentes, de líneas de código proyectadas, o ambas, en el sistema implementado.

Mide dos tipos de características:

- ✓ Los elementos de función.
- ✓ Los factores de complejidad.

Los elementos de función son los siguientes:

- ✓ Entradas (del usuario o sistema externo).
- ✓ Salidas (al usuario o sistema externo).
- ✓ Consultas del usuario (o interacciones).
- ✓ Archivos (o ficheros) lógicos usados por el sistema.
- ✓ Interfaces externas (o ficheros de sistemas externos).

El proyecto quedará dividido en los siguientes módulos:

MGClientes	Módulo de gestión de clientes
MGOferas	Módulo de gestión de Ofertas
MGBicicletas	Módulo de gestión de Bicicletas
MGIncidencias	Módulo de gestión de Incidencias
MGAquileres	Módulo de gestión de Alquileres
MGFechas	Módulo de gestión de Fechas

2.3. Estimaciones de coste, esfuerzo y duración

✓ FICHEROS

- Cliente
 - Id_cliente
 - D.N.I.
 - Nombre
 - Apellidos
 - Dirección
 - Teléfono
- Fecha
 - Día
 - Mes
 - Año
- Alquiler
 - Id_alquiler
 - Fecha_alquiler
 - Fecha_devolución
 - Id_bici
 - Tipo_bici
 - Precio_factura
- Bicicleta
 - Id_bici
 - Sexo {femenino, masculino}
 - Tamaño
 - Tipo {montaña, carretera, paseo}
 - Precio_por_dia
 - Estado {disponible, alquilada, enTaller}
- Oferta
 - Id_oferta
 - Tipo_bici {montaña, carretera, paseo}.
 - Numero_alquileres_para_descuento (por ejemplo cada 5 alquileres de una bici de montaña se descuenta un 20%)
 - Descuento
- Incidencia
 - Id_incidencia
 - Fecha_creación
 - Fecha_resolución
 - Id_bici
 - Concepto
 - Importe
 - Estado {disponible, alquilada, enTaller}

✓ ENTRADAS

- Dar de alta un cliente.
- Dar de alta una bici.
- Añadir una nueva oferta.
- Registrar un alquiler.
- Registrar una incidencia.
- Dar de baja un cliente.
- Dar de baja una bici.
- Dar de baja una oferta.
- Eliminar incidencia

✓ CONSULTAS

- Consideramos consultas aquellos procesos formados por una entrada y una salida, es decir el flujo de datos se produce en ambas direcciones.
- Hemos considerado los casos siguientes como consultas ya que para acceder por ejemplo a los datos de un cliente, una bici o una oferta primero se debe introducir un Id que lo identifique y después se mostrará la información por pantalla.
- Acceder a los datos de un cliente.
- Acceder a los datos de una bici.
- Acceder a los datos de una oferta.

✓ SALIDAS

- Una salida externa se deriva en el interior de la aplicación y proporciona información al usuario.
- Visualizar una oferta.
- Visualizar una incidencia.
- Visualizar los alquileres de un cliente.(Suponemos que ya estamos dentro del cliente y solo elegimos la opción de visualizar alquileres).
- Visualizar el alquiler de una bici.

ANÁLISIS DE LA DIFICULTAD

❖ ENTRADAS

DIFICULTAD	ENTRADA	FICHEROS ACCEDIDOS
Baja	Dar de alta un cliente	1.Cliente(4)
Baja	Dar de alta una bici	1.Bicicleta(6)
Baja	Añadir una nueva oferta	1.Oferta(4)
Media	Registrar alquiler	1.Alquiler(6) 2.Bicicleta(1)-Estado
Media	Registrar incidencia	1.Incidencia(7) 2.Bicicleta(1)-Estado
Baja	Dar de baja un cliente	1.Cliente(1) id
Baja	Dar de baja una bici	1.Bici(1)id
Baja	Dar de baja una oferta	1.Oferta(1)id
Baja	Eliminar incidencia	1.Incidencia(1)id

❖ SALIDAS

DIFICULTAD	SALIDA	FICHEROS ACCEDIDOS
Baja	Visualizar una oferta.	1.Cliente(4)
Baja	Visualizar una incidencia.	1.Bicicleta(6)
Baja	Visualizar los alquileres de un cliente	1.Oferta(4)
Media	Visualizar el alquiler de una bici	1.Bicicleta(1) 2.Alquiler(6)

❖ CONSULTAS

DIFICULTAD	SALIDA	FICHEROS ACCEDIDOS
Media	Acceder a los datos de un cliente.	1.Cliente(4) 2.Alquiler(6)
Baja	Acceder a los datos de una bici.	1.Bicicleta(6)
Baja	Acceder a los datos de una oferta.	1.Oferta(4)

❖ FICHEROS LÓGICOS

Nombre del fichero	Dificultad
Cliente	Baja
Fecha	Baja
Incidencia	Baja
Alquiler	Baja
Oferta	Baja
Bicicleta	Baja

PUNTOS FUNCIÓN SIN AJUSTAR

PUNTOS FUNCION	COMPLEJIDAD						TOTAL
	SIMPLE		MEDIA		COMPLEJA		
	#N	Peso	#N	Peso	#N	Peso	
Entradas	7	3	2	4	0	6	29
Salidas	3	4	1	5	0	7	17
Consultas	2	3	1	4	0	6	10
Ficheros Log	6	7	0	10	0	15	42
Interf.Extern	0	5	0	7	0	10	0
Total PF sin ajustar							98

CÁLCULO DEL FACTOR DE COMPLEJIDAD TOTAL

<u>Id</u>	<u>Factor de Complejidad</u>	<u>Valor (1..5)</u>
1	<u>Comunicación de datos</u>	<u>2</u>
2	<u>Proceso Distribuido</u>	<u>4</u>
3	<u>Objetivos de rendimiento</u>	<u>4</u>
4	<u>Integración de la aplicación</u>	<u>1</u>
5	<u>Tasa de transacciones</u>	<u>3</u>
6	<u>Entrada de datos interactiva</u>	<u>5</u>
7	<u>Eficiencia para el usuario final</u>	<u>3</u>
8	<u>Actualizaciones interactivas</u>	<u>5</u>
9	<u>Lógica de proceso interna compleja</u>	<u>1</u>
10	<u>Reusabilidad del código</u>	<u>1</u>
11	<u>Conversión e instalación</u>	<u>1</u>
12	<u>Facilidad de operación</u>	<u>1</u>
13	<u>Instalaciones múltiples</u>	<u>1</u>
14	<u>Facilidad de Cambios</u>	<u>5</u>
<u>Factor de complejidad total</u>		<u>37</u>

PUNTOS FUNCIÓN AJUSTADOS

$$\text{PFA} = \text{PFSA} * (0,65 + (0,01 * \text{FCT})) = 98 * (0,65 + (0,01 * 37)) = 99,96$$

ESFUERZO TOTAL

$$\text{E.T} = \text{PFA} * \text{Media de productividad} = 99,96 * 10 = 999.6 \text{ horas.}$$

Se adopta el valor del esfuerzo obtenido a partir de la medición realizada en puntos de función, adoptando una estrategia de peso medio.

CÁLCULO DEL TIEMPO TOTAL EN BASE A LAS PERSONAS IMPLICADAS

Las jornadas serán de 8h.Trabajando de lunes a viernes.

En el proyecto estarán trabajando 10 personas.

En base a esto calculamos:

$$8\text{h/día} * 5 \text{ días/semana} = 40 \text{ h/semana}$$

$$40\text{h/semana} * 4 \text{ semanas/mes} = 160\text{h/mes}$$

$$160\text{h/semana} * 10 \text{ personas} = 1600\text{h}$$

$$999,6\text{h} / 1600 \text{ h/mes} = 0,6 \text{ meses}$$

Aproximadamente unos 20 días.

3. ESTRATEGIA DE GESTIÓN DE RIESGOS

3.1. Análisis del riesgo

En nuestro proyecto vamos a distinguir los siguientes riesgos iniciales:

Riesgo	Descripción	Técnica de Reducción
Cambio en los requisitos	Modificación en los requisitos expuestos al inicio por el cliente.	Hablar con el cliente y explicarle la importancia de tener claro el alcance final del proyecto, y el impacto que supone un cambio.
Componentes defectuosos	Errores en los componentes software y/o hardware del proyecto.	Utilizar componentes de eficiencia comprobada por los trabajadores.
Baja de algún componente del equipo	Ausencia del trabajo de un componente del equipo debido a cualquier circunstancia externa al trabajo.	Diversificación del conocimiento funcional y documentación de tareas asignadas y trabajo realizado por cada componente.
Subestimar el tiempo de desarrollo.	Estimación del tiempo de desarrollo inferior a la que realmente merece el proyecto.	Estudio, medición y comparación previa de todas las tareas que debemos realizar.
Desarrollo de las funciones y propiedades erróneas	Errores en el estudio previo de las funciones a realizar	Realizar constantemente algunas de las siguientes técnicas: análisis de organización, prototipado, manuales de usuarios preliminares, análisis de rendimiento sin nombre
Indisponibilidad del hardware	Retraso en la entrega del hardware por parte del proveedor.	Hablar con el proveedor de la importancia que tiene la pronta entrega y acordar la penalización por el retraso

Riesgos	Descripción	Técnica de reducción
Competencia del equipo de desarrollo	Las necesidades del proyecto requieren de conocimientos que el personal desconoce o no es capaz de utilizar correctamente.	Determinar las necesidades de formación requeridas para este proyecto. Contar, en la medida de lo posible, con recursos que hayan participado en proyectos de similares características.
Problemas eléctricos	Fallos en el suministro de electricidad	Guardar los cambios realizados en el proyecto frecuentemente
Desarrollo erróneo del interfaz de usuario	Error en la elaboración del interfaz de usuario, como el no incluir elementos que el usuario necesita	Conviene una participación constante con el usuario, realizar prototipados, escenarios...
Alteraciones en el rendimiento de tiempo real	Ejecutar una operación de la aplicación una vez finalizada y comprobar que tarda más tiempo del esperado.	Se debe de realizar pequeñas pruebas durante el desarrollo del proyecto.
Mala coordinación entre los trabajadores	Falta de entendimiento entre los responsables del proyecto, sin saber cada uno cuál es su labor a realizar	Análisis previo de todas las tareas que hay que realizar y asignación de cada una de ellas antes de empezar a trabajar.
Incompatibilidad de versiones de desarrollo	Empleo de herramientas de desarrollo cuya versión sea incompatible con el entorno de producción previsto.	Definición de las versiones a emplear de cada herramienta de desarrollo, para evitar incompatibilidades.

A la hora de clasificar los riesgos anteriores nos vamos a ayudar de la siguiente tabla.

Atributo	Valor	Descripción
Consecuencia o impacto	Catastrófico	Más de 30 días de retraso en la planificación Reducción de la funcionalidad del producto superioral 10%
	Crítico	Menos de 30 días de retraso en la planificación Reducción de la funcionalidad del producto inferioral 10%
	Serio	Menos de 15 días de retraso en la planificación Reducción de la funcionalidad del producto inferioral 5%
	Menor	Menos de 7 días de retraso en la planificación Reducción de la funcionalidad del producto inferioral 2%
	Despreciable	Impacto despreciable en el proyecto
Probabilidad	Improbable	Menor del 10% Probabilidad cercana a cero
	Remota	Del 10% al 25% Poco probable, aunque posible
	Ocasional	Del 25% al 75% Puede ocurrir alguna vez
	Probable	Del 75% al 90% Ocurre repetidamente
	Frecuente	Mayor del 90% Ocurrirá varias veces

3.2. Estudio de los riesgos

Con la ayuda de las dos tablas anteriores vamos a clasificar los riesgos, para posteriormente poder gestionar los riesgos más importantes. Los distintos niveles de riesgo que vamos a abordar en función del impacto y la probabilidad son en grado ascendente: tolerable, bajo, medio, alto, intolerable.

Probabilidad Impacto	Frecuente	Probable	Ocasional	Remoto
Catastrófico	Intolerable	Intolerable	Alto	Medio
Crítico	Intolerable	Alto	Medio	Bajo
Severo	Alto	Medio	Bajo	Tolerable
Menor	Medio	Bajo	Tolerable	Tolerable

Ahora ya podemos completar la siguiente tabla de clasificación de riesgos:

Riesgo	Impacto	Probabilidad	Nivel
Cambio en los requisitos	Crítico	Ocasional	Medio
Componentes defectuosos	Crítico	Ocasional	Medio
Baja de algún componente del equipo	Severo	Ocasional	Bajo
Subestimar el tiempo de desarrollo.	Crítico	Frecuente	Intolerable
Desarrollo de las funciones y propiedades erróneas	Crítico	Remoto	Bajo
Indisponibilidad del hardware	Severo	Ocasional	Bajo
Competencia del equipo de desarrollo	Severo	Remoto	Tolerable
Problemas eléctricos	Menor	Ocasional	Tolerable
Desarrollo erróneo del interfaz de usuario	Crítico	Remoto	Bajo
Alteraciones en el rendimiento de tiempo real	Severo	Remoto	Tolerable
Mala coordinación entre los trabajadores	Severo	Ocasional	Bajo
Incompatibilidad de versiones de desarrollo	Crítico	Probable	Alto

Una vez que hemos analizado los riesgos vamos a seguir con su estudio decidiendo cuales van a ser los planes de actuación.

Riesgo	Decisión
Subestimar el tiempo de desarrollo	Controlar
Incompatibilidad de versiones de desarrollo	
Componentes defectuosos	
Desarrollo de las funciones y propiedades erróneas	Evitar
Indisponibilidad del hardware	
Cambio en los requisitos	
Desarrollo erróneo del interfaz de usuario	
Problemas eléctricos	Asumir
Competencia del equipo de desarrollo	
Alteraciones en el rendimiento de tiempo real	
Mala coordinación entre los trabajadores	
Baja de algún componente del equipo	

3.3. Plan de gestión del riesgo

Una vez que hemos analizado y estudiado los riesgos que podemos sufrir durante el proceso de nuestro proyecto, vamos a tratar de gestionar los riesgos más importantes con el fin de evitar que estos sucedan o si no los podemos evitar por lo menos reducirlos. Para ello vamos a describir los pasos de reducción, supervisión y gestión de riesgos que se han decidido controlar.

3.3.1. Primer Riesgo: Subestimar el tiempo de desarrollo

Reducción: Ver las posibilidades que tenemos de comprar nuevos componentes que nos ahorren tiempo de trabajo (ej. Generadores de código), y si estas posibilidades son escasas debido al tema económico negociar con el cliente el alcance del proyecto o ampliar el plazo de la entrega.

Supervisión: Se comparará la fecha de entrega de cada uno de los entregables, con respecto de la fecha prevista, con el fin de detectar retrasos de consideración. Analizar el esfuerzo requerido para emplear los componentes reutilizables, con el fin de determinar si realmente compensa su utilización para agilizar el desarrollo.

Gestión: Hablar con el cliente, y comunicarle con más de un mes de antelación el retraso que va a sufrir en la entrega, otra opción sería proponerle minimizar el alcance del proyecto.

3.3.2. Segundo Riesgo: Incompatibilidad de versiones de desarrollo

Reducción: Definición de las versiones a emplear de cada herramienta de desarrollo, tanto para las fases de desarrollo como para la preparación del entorno de producción, para evitar incompatibilidades.

Supervisión: Realizar varias reuniones con los trabajadores donde podamos recoger los problemas de incompatibilidad que estén sufriendo para poder solucionarlos.

Gestión: Contactar con el proveedor de las versiones de desarrollo y pedirle la versión más estable y actualizada en función del trabajo que vamos a utilizar.

3.3.3. Tercer Riesgo: Componentes Defectuosos

Reducción: Consultar con el proveedor los componentes de mayor y menor rendimiento, para poder elegir correctamente.

Supervisión: Realizar multitud de pruebas regularmente para probar estos componentes.

Gestión: Consultar con el proveedor cuál sería la mejor opción si reparar el componente defectuoso o comprar uno nuevo, y en función de lo que nos explique decidir.

El resto de riesgos, tienen un impacto menor, por lo que intentaremos evitarlos pero nuestro esfuerzo por conseguirlo ha de ser menor, por ello no es necesario indicar en este punto el plan de actuación.

4. PLANIFICACIÓN TEMPORAL

4.1. Estructura de descomposición del trabajo

En función de la estimación informada en el punto 2 de este documento, se establece la siguiente tabla de descomposición del trabajo. Para facilitar la tarea de convertir el esfuerzo estimado en persona/mes en días, se han tomado veinte días hábiles por mes (trabajando de lunes a viernes).

AE-> Acción	Planificación y estudio de requisitos	Ingeniería		Const. Y Adapt.	
		Análisis	Diseño	Código	Prueba
Proyecto	1.1 I:18.11.11 F:25.11.11 R: Laura, Jose Luis, María, Javier V., Sergio, Isaac. E:Planificación y estudio de requisitos				
MGClientes		2.1 I:05.12.11 F:07.12.11 R:Laura E:Análisis MGClientes	2.2 I:13.12.11 F:13.12.11 R:Alberto E:Diseño MGClientes	2.3 I:10.01.12 F:12.01.12 R:Alberto E:Código MGClientes	2.4 I:24.01.12 F:25.01.12 R:Laura E:Ejecutar pruebas
MGOferas		3.1 I:07.12.11 F:08.12.11 R:Sergio E:Análisis MGOferas	3.2 I:14.12.11 F:14.12.11 R:Adrián E:Diseño MGOferas	3.3 I:10.01.12 F:17.01.12 R:Adrián E:Código MGOferas	3.4 I:24.01.12 F:24.01.12 R:Sergio E:Ejecutar pruebas
MGBicicletas		4.1 I:05.12.11 F:06.12.11 R:Isaac E:Análisis MGBicicletas	4.2 I:13.12.11 F:14.12.11 R:Javi M. E:Diseño MGBicicletas	4.3 I:12.01.12 F:16.01.12 R:Javi M. E:Código MGBicicletas	4.4 I:24.01.12 F:25.01.12 R:Isaac E:Ejecutar pruebas

AE-> Acción	Planificación y estudio de requisitos	Ingeniería		Const. Y Adapt.	
		Análisis	Diseño	Código	Prueba
MGIncidencias		5.1 I:05.12.11 F:07.12.11 R:Javier V. E:Análisis MGIncidencias	5.2 I:13.12.11 F:14.12.11 R:Adrián E:Diseño MGIncidencias	5.3 I:10.01.12 F:16.01.12 R:Isaac E:Código MGIncidencias	5.4 I:25.01.12 F:25.01.12 R:Javier V. E:Ejecutar pruebas
MGAquileres		6.1 I:05.12.11 F:06.12.11 R:Jose Luis E:Análisis MGAquileres	6.2 I:14.12.11 F:14.12.11 R:Guillermo E:Diseño MGAquileres	6.3 I:11.01.12 F:17.01.12 R:Guillermo E:Código MGAquileres	6.4 I:24.01.12 F:25.01.12 R:Jose Luis E:Ejecutar Pruebas
MGFechas		7.1 I:05.12.11 F:06.12.11 R:María E:Análisis MGFechas	7.2 I:13.12.11 F:13.12.11 R:Guillermo E:Diseño MGFechas	7.3 I:10.01.12 F:13.01.12 R:María E:Código MGFechas	7.4 I:24.01.12 F:24.01.12 R:María E:Ejecutar Pruebas

4.2. Gráfico Gantt

Dado el tamaño de la planificación, no se ha incluido dentro de este documento. Puede ser consultada en el fichero "Punto4 Projects.mpp"

4.3. Red de tareas

Dado el tamaño de la planificación, no se ha incluido dentro de este documento. Puede ser consultada en el fichero "diagramadegant.mpp"

4.4. Tabla de uso de recursos

Dado el tamaño de la planificación, no se ha incluido dentro de este

5. RECURSOS DEL PROYECTO

5.1. Personal

Analista funcional

Es el encargado de mantener informado del estado del proyecto al jefe, siendo la persona que relaciona al cliente con los empleados del proyecto. Se encarga de revisar que el proyecto siga la planificación propuesta diariamente. Marca la planificación y directrices del proyecto. También se encargara de elaborar los planes de pruebas.

Analista programador

Se encarga de realizar los diseños técnicos de los distintos módulos que componen el producto, generando los diagramas destinados a ser interpretados por los programadores.

Programadores

Son los encargados de codificar la lógica interna del programa y la implementación de las interfaces realizadas por el diseñador. También se encargan de la ejecución de las pruebas unitarias y la elaboración de los instalables.

Testers

Se encargan de ejecutar los planes de prueba especificados para los distintos módulos, y certificar que cumplen con las garantías de calidad establecidas, con los requisitos definidos inicialmente.

5.2. Hardware y Software

Recursos Hardware

Necesidad	Especificaciones	Cantidad
Estaciones de trabajo para desarrolladores	PC CPU Intel Core i7, 4GB de RAM	5
Servidor de pruebas	PC CPU Intel Core i7, 4GB de RAM	1
Servidor del repositorio y de herramientas de pruebas	Dropbox que une las diversas estaciones de trabajo para un repositorio común.	1
Servidor de base de datos de producción.	PC CPU Intel Core i7, 4GB de RAM	1
Servidor web de producción	PC CPU Intel Core i7, 4GB de RAM	1

Recursos Software

Necesidad	Producto
Sistema operativo de los servidores	Windows 7
Sistema operativo de las máquinas de desarrollo	Windows 7
Gestor de base de datos	MySQL
Herramientas de desarrollo C++	Visual Studio 2010
Herramientas de modelado UML	Bouml y ArgoUML

6. ORGANIZACIÓN DEL PERSONAL

6.1. Estructura de equipo

Dado el tamaño del producto a desarrollar, se establece un único equipo de desarrollo. Su estructura jerárquica estará basada en la organización Mantei "descentralizado democrático".

NOMBRE	TAREAS
Armero Guerra, María	PLAN DE PROYECTO
Jiménez Soria, Alberto	CODIGO
Martin Martínez, Javier	CODIGO
Muñoz de la Peña Herranz, Laura	DOCUMENTO REQUISITOS
de Pablos Blanco, Guillermo	CODIGO
Sánchez Castellanos, Isaac	PLAN DE PROYECTO
Sánchez Jiménez, Jose Luis	DOCUMENTO REQUISITOS
Sánchez Sánchez, Adrian	CODIGO
Suarez Solorzano, Sergio	PLAN DE PROYECTO
Valenciano López, Javier	PLAN DE PROYECTO

Leyenda:

DOCUMENTO REQUISITOS
PLAN DE PROYECTO
DIAGRAMAS
CODIGO

6.2. Informes de gestión

Un miembro del equipo se encargará de llevar una memoria del estado del proyecto, aportando información obtenida de las reuniones semanales realizadas con el resto del equipo, y que será almacenado en el repositorio del mismo.

En este informe se aportarán datos de:

- Grados de avance de las tareas, y comparativa con la planificación inicial
- Actividades realizadas desde la fecha de elaboración del último informe
- Actividades previstas hasta la fecha de elaboración de próximo informe
- Incidencias producidas
- Seguimiento de los riesgos del proyecto

Miembros del Equipo:

Armero Guerra, María

Programadora sénior, con conocimientos de C++ y pascal, encargada del desarrollo y actualizaciones del plan de proyecto.

Jiménez Soria, Alberto

Analista programador, encargado del desarrollo del software y tester. Conocimientos en C++ y pascal

Martin Martínez, Javier

Analista programador, encargado del desarrollo del software y tester. Conocimientos en C++ y pascal

Muñoz de la Peña Herranz, Laura

Programadora sénior, encargada de redactar los requisitos específicos del software y su continua actualización durante el desarrollo, con conocimiento de C++ y pascal, y herramientas de edición de texto.

de Pablos Blanco, Guillermo

Analista programador, encargado del desarrollo del software y tester. Conocimientos en C++ y pascal.

Sánchez Castellanos, Isaac

Programador sénior, con conocimientos de C++ y pascal, encargado del desarrollo y actualizaciones del plan de proyecto.

Sánchez Jiménez, Jose Luis

Programador sénior, encargado de logística y redactar los requisitos específicos del software y su continua actualización durante el desarrollo, con conocimiento de C++ y pascal, y herramientas de edición de texto.

Sánchez Sánchez, Adrian

Analista programador, encargado del desarrollo del software y tester. Conocimientos en C++ y pascal.

Suarez Solorzano, Sergio

Programador sénior, con conocimientos de C++ y pascal, encargado del desarrollo y actualizaciones del plan de proyecto.

Valenciano López, Javier

Programador sénior, con conocimientos de C++ y pascal, encargado del desarrollo y actualizaciones del plan de proyecto.

7. MECANISMOS DE SEGUIMIENTO Y CONTROL

7.1. Garantía de calidad y control

Para garantizar un software de calidad, se realizarán una serie de actividades planificadas y sistemáticas para aportar la confianza en que el software satisfará los requisitos. Estas actividades son utilizadas para cumplir con dos puntos importantes:

1. Mantener bajo control un proceso
2. Eliminar las causas de defecto.

Esta garantía se aplicará durante toda la vida de desarrollo del software hasta la entrega al cliente. Este control estará presente en:

- Estrategias de prueba.
- Métodos y herramientas de análisis, diseño, programación y prueba.
- Inspecciones técnicas formales en todos los pasos en la elaboración del software.
- Control de la documentación del software y de los cambios registrados. Estos documentos serán:
 - Especificación de requisitos.
 - Plan de proyecto.
 - Planificación.
 - Diseño funcional, técnico y de arquitectura.
 - Puntos acordados en una reunión de trabajo.
 - Memoria
 - Verificación y validación de las distintas fases de vida del software.

Tendremos muy en cuenta durante la elaboración del software los siguientes factores:

1. Corrección (¿Hace lo que se le pide?): comprobación en que lo que se hace, satisface sus especificaciones y consigue los objetivos.
2. Fiabilidad (¿Es fiable?): llevar a cabo las operaciones especificadas.
3. Eficiencia (¿Qué software y hardware necesito?): la cantidad de recursos que necesito para realizar las operaciones con unos tiempos esperados.
4. Integridad (¿Controlo su uso?): el grado con que puede controlarse el acceso a datos del personal.
5. Facilidad de uso (¿Es fácil de manejar?): el esfuerzo requerido para aprender a manejar el software.
6. Facilidad de mantenimiento (¿Son los fallos localizables?): el esfuerzo en localizar y solucionar errores.
7. Flexibilidad (¿Se podría añadir nuevas funciones?): el esfuerzo en poder modificar el software.
8. Facilidad de prueba (¿Se puede probar todas las opciones?): la dificultad en poder probar el software para que cumpla los requisitos.
9. Portabilidad (¿Podré usar el software en otro sistema?): el grado de esfuerzo en poder usar el software en otro sistema.
10. Reusabilidad (¿Son reutilizables algunos módulos del software?): el grado en que se puede reutilizar partes del software en otros proyectos.
11. Interoperabilidad (¿Podrá comunicarse con otro software?): el esfuerzo necesario para poder comunicar nuestro software con otros.

7.2. Gestión y Control de cambios

Control de cambios

Se realizará un informe sobre los cambios que se realizan durante el todo el proyecto.

Informes de estado

Estos informes nos proporcionan el estado actual de las diferentes partes del proyecto para poder ver el estado general del mismo. Con ellos priorizamos las partes correspondientes para poder cumplir con los plazos acordados.