

Plan de proyecto Software

v4.0.1

Ingeniería del Software



PiKey Team- P_KT :

Jesús Aguirre Pemán
Enrique Ballesteros Horcajo
Jaime Dan Porras Rhee
Ignacio Iker Prado Rujas
Alejandro Villarín Prieto

3 de Junio de 2013

Índice general

I. Introducción	1
1. Propósito del plan	1
2. Ámbito del proyecto y objetivos	1
2.1. Declaración del ámbito	1
2.2. Funciones principales	2
2.3. Aspectos de rendimiento	2
2.4. Restricciones y técnicas de gestión	2
3. Modelo de proceso	3
II. Estimaciones del proyecto	5
1. Datos históricos	5
2. Técnicas de estimación	5
3. Estimaciones de esfuerzo, coste y duración	6
III. Estrategia de gestión del riesgo	7
IV. Planificación temporal	9
1. Estructura de descomposición del trabajo, Planificación temporal	9
2. Gráfico de Gantt	11
3. Red de tareas	12
4. Tabla de uso de recursos	13
V. Recursos del proyecto	15
1. Personal	15
2. Hardware y software	15
2.1. Componentes reutilizables de software	15
2.2. Recursos de entorno	16
3. Lista de recursos	16
VI. Mecanismos de seguimiento y control	19
1. Garantía de calidad y control	19
2. Gestión y control de cambios	19

Parte I

Introducción

1. Propósito del plan

El propósito del Plan de Desarrollo de Software es ofrecer toda la información necesaria para controlar el desarrollo de nuestro proyecto KIKE HOSTELERIA.

El Plan de Proyecto es un modelo sistemático que se elabora antes de realizar una acción cuyo objetivo principal es dirigir el proyecto para que este vaya por un buen camino y así lograr los resultados deseados es decir el cumplimiento de los objetivos.

Dicho documento, además de explicar a qué usuarios va dirigido y las funciones que ejecuta, proporciona una visión global del enfoque de desarrollo propuesto.

En nuestro proyecto no existirá la figura del Jefe de Proyecto, por lo que la responsabilidad de la planificación de recursos y el control de progresos recaerá sobre todo el equipo.

2. Ámbito del proyecto y objetivos

2.1. Declaración del ámbito

La comunicación entre el personal de un restaurante o de un hotel es vital, y es lo que determina la velocidad y la eficiencia en la realización de tareas del negocio. En un restaurante, los clientes valoran sobremanera la rapidez con la que son atendidos, y el tiempo que tardan en llegar a la mesa sus comandas.

Nuestro producto pretende reducir drásticamente el tiempo que transcurre entre que los clientes son atendidos por los camareros y que su comanda llegue a cocina. Por tanto, hemos planteado un sistema tecnológico que, mediante un servidor, transfiere instantáneamente el pedido de los clientes de las tablets de los camareros al terminal situado en cocina.

El software KIKE HOSTELERIA está pensado para negocios de hostelería de carácter medio. La capacidad adquisitiva de dicho negocio debe ser suficiente como para sufragar los gastos que conlleve la compra del hardware que necesita nuestra aplicación. Además, KIKE HOSTELERIA, mediante el Log In de los usuarios de la aplicación, diferencia in-

ternamente a cada uno de los siguientes 5 tipos diferentes de usuario, y ofrecerá diferentes posibilidades según su rango:

- Jefe.
- Maître / recepcionista.
- Camarero.
- Chef / cocinero.
- Encargado de limpieza.

2.2. Funciones principales

Dentro de las funciones que oferta nuestro software, cabe destacar la gestión de las bases de datos tanto de clientes como de empleados, sin las cuales la aplicación no podría funcionar. Esto incluye añadir, editar, dar de baja empleados o clientes, así como mostrar las fichas de cada uno de ellos.

Adicionalmente, KIKE HOSTELERIA ofrece servicios que facilitarán las tareas tanto a empleados del negocio como a clientes.

Cabe destacar el novedoso sistema de pedidos, que mediante el uso de tablets por parte de los empleados, podrá comunicar instantáneamente las comandas a cocina.

Entre los servicios ofertados a los clientes destaca la reserva desde la habitación, con la que en unos pocos clics podrá ordenar su comanda eligiendo entre la oferta de platos a cargo del restaurante.

2.3. Aspectos de rendimiento

La memoria que consume la aplicación es muy reducida, por lo que podrá combinarse con el sistema multitareas de las tablets de los camareros, y con otras aplicaciones abiertas en los ordenadores de escritorio.

La aplicación necesita conexión con el servidor para poder asegurar el correcto funcionamiento del software. En el propio servidor están almacenadas las bases de datos de clientes y empleados.

2.4. Restricciones y técnicas de gestión

Respecto a **restricciones económicas**, como ya hemos comentado, lo único necesario es poder costear el hardware requerido para el funcionamiento de la aplicación.

El **período de adaptación** entre el sistema anterior y el utilizado por KIKE HOSTELERIA es únicamente el necesario en conseguir el hardware y enseñar al personal a utilizarlo. No obstante lo último es sumamente sencillo debido a lo intuitivo de la aplicación.

El **Hardware** utilizado será Windows (Windows 7 para ordenadores de sobremesa y Windows RT para tablets) e iOS (iOS 6 para tablets). La aplicación está diseñada utilizando el lenguaje de programación Java.

3. Modelo de proceso

Para la realización de la aplicación KIKE HOSTELERIA hemos optado por utilizar el modelo del Proceso Unificado, que se caracteriza por ser un marco de desarrollo software dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. Este es el modelo que mejor se adecua a nuestro proyecto, pues los casos de uso, que forman su base, han sido definidos y trabajados desde el comienzo. Además, el Proceso Unificado nos permitirá la refinación de los documentos que presentemos en cada iteración.

Parte II

Estimaciones del proyecto

La estimación en un proyecto software es una de las partes indispensables dentro de la planificación, aunque es complicada y requiere experiencia. Por otro lado, es claro que nunca podrá ser definitiva y perfecta, pues el desarrollo de software sufre continuos cambios a lo largo de su vida.

De todos modos, una buena estimación resulta beneficiosa, pues ahorra bastante tiempo, que es esencial en el proyecto. Además, proporciona un marco de trabajo, para fijar fechas, costes y recursos, y cuándo estos se van a utilizar.

1. Datos históricos

No se dispone de ésta información, tratándose de un proyecto como el nuestro, académico, que se podría considerar de reingeniería.

2. Técnicas de estimación

Existen varias, pero la que vamos a utilizar se encuadra dentro de las técnicas de descomposición basadas en el problema, y nace del estudio del tamaño del software en base a su funcionalidad (Puntos de Función o PF). No es baladí subrayar esto último: los PF miden la funcionalidad que el usuario solicita y recibe, no la complejidad. Existen otras métricas, como el recuento de líneas de código o LDC, pero no hay estándares (ISO etc.), y con nuestros conocimientos sería complicado obtener una buena estimación. Además, el número de líneas de código no es bueno como benchmark, pues varía notablemente en función del lenguaje de programación, el programador... Otro dato importante es que la métrica de PF es suficientemente sencilla como para no retrasar o perjudicar el proyecto, pero suficientemente potente como para ser de gran utilidad.

En total, se ha estimado que la aplicación tiene 267 puntos de función sin ajustar, con un factor de ajuste de 1.04 hace un total de 277.7 puntos de función ajustados.

Para más detalle, se dispone del documento adjunto **Estimación del proyecto Software**, donde se estudia en profundidad el número de puntos de función y su origen.

3. Estimaciones de esfuerzo, coste y duración

Como conclusión del apartado anterior y utilizando la herramienta COCOMO II.2000.4 (CONstructive COst MOdel), podemos obtener una estimación en esfuerzo, dinero y tiempo para el producto.

En cuanto al esfuerzo, se calculan 71.5 personas · mes. El coste se estima en 38.544,637 euros, repartidos en un total de 14.7 meses.

De nuevo, esto tan sólo es una visión global. Este contenido está ampliado en el documento anexo **Estimación del proyecto Software**.

Parte III

Estrategia de gestión del riesgo

Los detalles de esta parte se encuentran en el documento **Gestión de Riesgos**. En el documento se trata la gestión de los riesgos. Primero se llevó a cabo la identificación de los riesgos que podían afectar al proyecto. Seguidamente se analizaron estos riesgos, estudiando tanto la probabilidad que tenían de ocurrir como la consecuencia que podían tener en el proyecto. Posteriormente se priorizaron los riesgos, eligiendo aquellos que tenían mayor nivel de riesgo. Aplicando el Principio de Pareto, se eligieron los primeros 7 riesgos. Finalmente se gestionaron estos 7 riesgos.

Parte IV

Planificación temporal

1. Estructura de descomposición del trabajo, Planificación temporal

Para estructurar la descomposición del trabajo hemos optado por un grado de rigor medio. Ni un esquema excesivamente sofisticado, ni demasiados detalles superfluos. Para estructurar las tareas, hemos optado por un calendario dividido en dos cuatrimestres, que incluyen una hora de trabajo diaria y las horas de laboratorio. También se incluyen dos horas tanto los sábados como los domingos. Esto puede parecer excesivo, pero refleja nuestra intención de trabajo.

Calendario base el jue 28/02/13
KikeHosteleria.mpp

CALENDARIO BASE:	CalendarioPaiky
Día	Horas
lunes	22:00 - 23:00
martes	22:00 - 23:00
miércoles	22:00 - 23:00
jueves	22:00 - 23:00
viernes	9:00 - 11:00, 22:00 - 23:00
sábado	17:00 - 19:00
domingo	17:00 - 19:00

Figura IV.1: Calendario laboral

La EDT detallada de todo el proyecto, aunque es una mala aproximación debido a la falta de experiencia, nos va a permitir hacer un seguimiento del proyecto y ver cómo de certeras son las previsiones. Las tareas a las que ya se las ha asignado un tiempo y unas fechas pretender reflejar lo hecho hasta ahora y planificar las acciones futuras.

1. ESTRUCTURA DE DESCOMPOSICIÓN DEL TRABAJO, PLANIFICACIÓN TEMPORAL

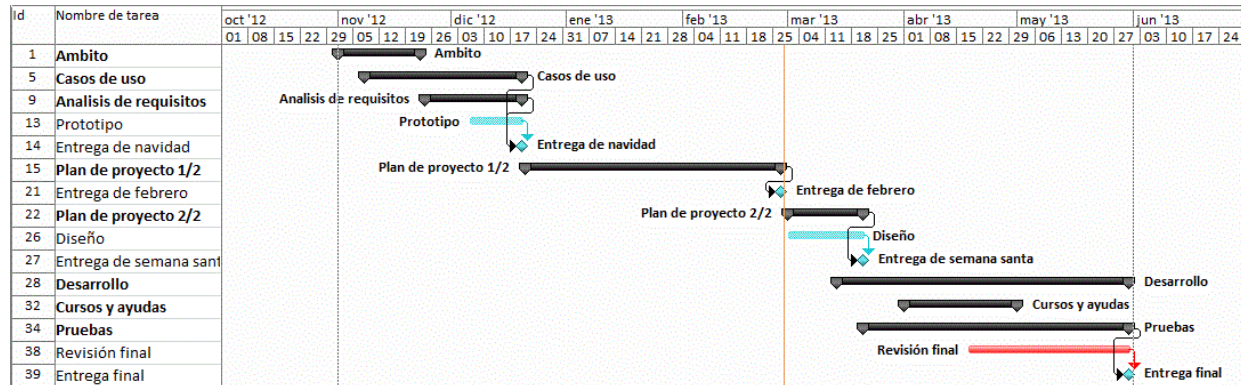
Plan de proyecto

Id	Nombre de tarea	Duración	Comienzo	Fin
1	Ambito	34 horas	jue 01/11/12	vie 23/11/12
2	Determinar el ambito	9 horas	jue 01/11/12	mié 07/11/12
3	Definir recursos preliminares	28 horas	lun 05/11/12	jue 22/11/12
4	Ambito terminado	0 horas	vie 23/11/12	vie 23/11/12
5	Casos de uso	64 horas	jue 08/11/12	mié 19/12/12
6	Listado inicial	12 horas	jue 08/11/12	jue 15/11/12
7	Desarrollo	58 horas	lun 12/11/12	mié 19/12/12
8	Análisis hecho	0 horas	mié 19/12/12	mié 19/12/12
9	Análisis de requisitos	39 horas	sáb 24/11/12	mié 19/12/12
10	Introducción	29 horas	sáb 24/11/12	dom 09/12/12
11	Funciones principales	20 horas	jue 06/12/12	mié 19/12/12
12	Requisitos terminados	0 horas	mié 19/12/12	mié 19/12/12
13	Prototipo	20 horas	jue 06/12/12	mié 19/12/12
14	Entrega de navidad	0 horas	jue 20/12/12	jue 20/12/12
15	Plan de proyecto 1/2	108 horas	vie 21/12/12	mié 27/02/13
16	Introducción	30 horas	vie 21/12/12	mié 09/01/13
17	Estimación	39 horas	jue 10/01/13	sáb 02/02/13
18	Gestión de riesgos	39 horas	jue 10/01/13	sáb 02/02/13
19	Planificación temporal	39 horas	jue 10/01/13	sáb 02/02/13
20	Primera parte terminada	0 horas	mié 27/02/13	mié 27/02/13
21	Entrega de febrero	0 horas	mié 27/02/13	mié 27/02/13
22	Plan de proyecto 2/2	30 horas	vie 01/03/13	mié 20/03/13
23	Plan de garantía de calidad	30 horas	vie 01/03/13	mié 20/03/13
24	Gestión de la configuración	30 horas	vie 01/03/13	mié 20/03/13
25	Plan terminado	0 horas	mié 20/03/13	mié 20/03/13
26	Diseño	30 horas	vie 01/03/13	mié 20/03/13
27	Entrega de semana santa	0 días	jue 21/03/13	jue 21/03/13
28	Desarrollo	122 horas	jue 14/03/13	jue 30/05/13
29	Desarrollar el código	122 horas	jue 14/03/13	jue 30/05/13
30	Depuración preliminar	23 horas	jue 21/03/13	jue 04/04/13
31	Desarrollo terminado	0 horas	jue 30/05/13	jue 30/05/13
32	Cursos y ayudas	46 horas	lun 01/04/13	mar 30/04/13
33	Desarrollar cursos y ayudas	46 horas	lun 01/04/13	mar 30/04/13
34	Pruebas	111 horas	jue 21/03/13	jue 30/05/13
35	Desarrollar plan de pruebas	23 horas	jue 21/03/13	jue 04/04/13
36	Realizar pruebas	89 horas	jue 04/04/13	jue 30/05/13
37	Fin de las pruebas	0 horas	jue 30/05/13	jue 30/05/13
38	Revisión final	67 horas	jue 18/04/13	jue 30/05/13
39	Entrega final	0 horas	vie 31/05/13	vie 31/05/13

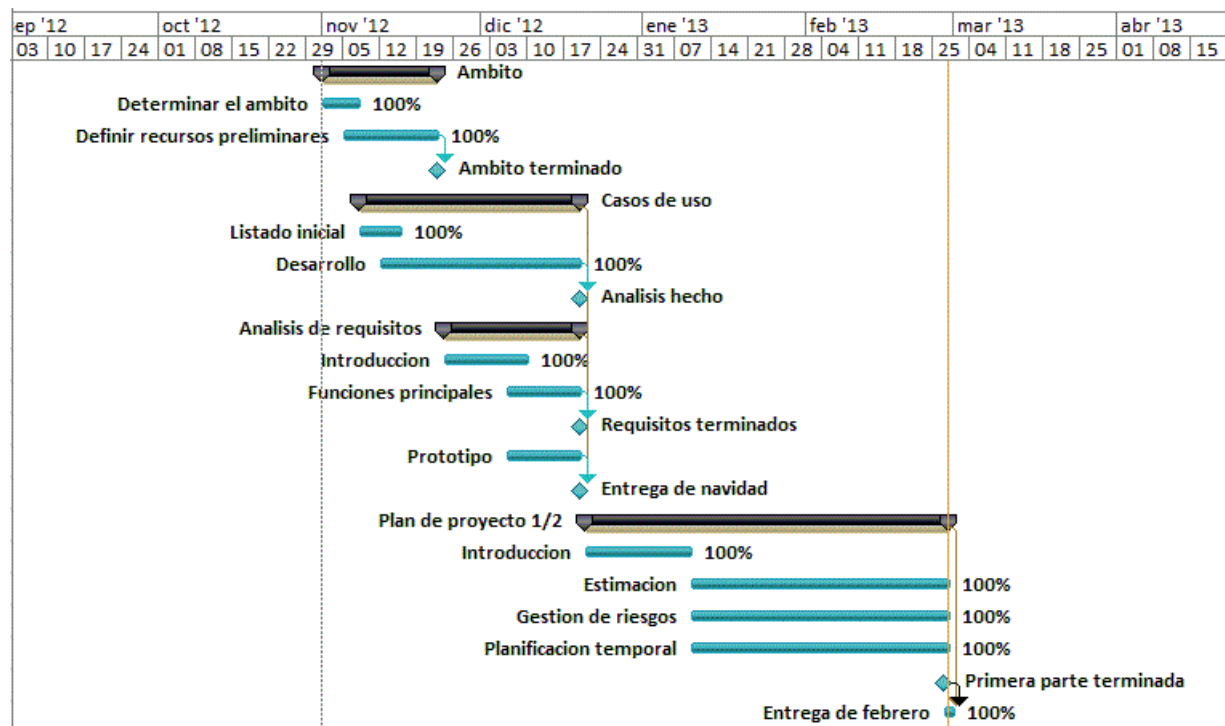
Figura IV.2: Hoja de las tareas

2. Gráfico de Gantt

Este gráfico de Gantt muestra la distribución principal del proyecto, sin mucho detalle:



Y el siguiente detalla lo hecho hasta la entrega de febrero:



Puede observarse que está contabilizado como un 100 hecho. Esto no debe hacernos pensar que estén terminados, pues aun deben ser modificados durante el proyecto.

3. Red de tareas

Las tareas han estado condicionadas principalmente por las entregas. La entrega de navidad obligó a tener la especificación de requisitos, los casos de uso y el prototipo preparados. Y la de febrero a tener la primera parte del plan de proyecto terminada. Esta es la red de tareas principal hasta la entrega de febrero:

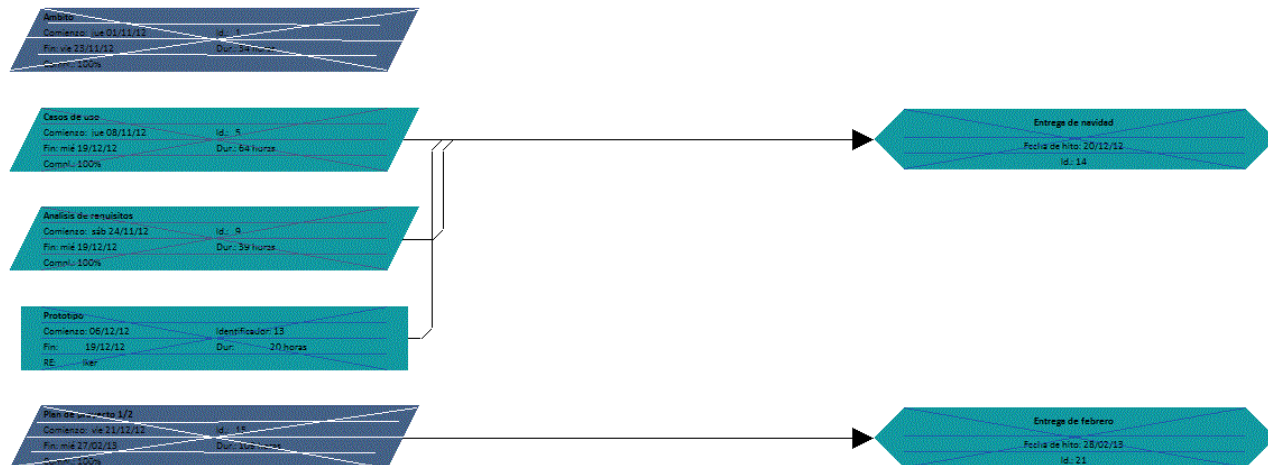


Figura IV.3: Red de tareas

Si bien posteriormente la SRS y los casos de uso han sido revisados, puede verse cómo para ser posible la entrega fue necesario tener todo a punto para ese momento. Además el prototipo ha sido desechado, y tanto los casos de uso como la SRS están en continua revisión y cambio.

4. Tabla de uso de recursos

Las siguientes tablas muestran cómo se ha distribuido el trabajo entre los recursos y como piensa distribuirse. Por ahora es una aproximación y no acaban de ser datos fiables. Sin embargo sí que permiten hacerse una idea de qué tareas requieren más esfuerzo

Id	Nombre del recurso	Trabajo
1	Jaime	610,5 horas
	<i>Determinar el ambito</i>	9 horas
	<i>Definir recursos preliminares</i>	28 horas
	<i>Listado inicial</i>	12 horas
	<i>Desarrollo</i>	29 horas
	<i>Introduccion</i>	14,5 horas
	<i>Funciones principales</i>	10 horas
	<i>Gestion de riesgos</i>	154 horas
	<i>Diseño</i>	30 horas
	<i>Desarrollar el código</i>	122 horas
	<i>Desarrollar cursos y ayudas</i>	46 horas
	<i>Realizar pruebas</i>	89 horas
	<i>Revisión final</i>	67 horas
2	Alejandro	617,5 horas
	<i>Determinar el ambito</i>	9 horas
	<i>Definir recursos preliminares</i>	28 horas
	<i>Listado inicial</i>	12 horas
	<i>Desarrollo</i>	29 horas
	<i>Introduccion</i>	14,5 horas
	<i>Funciones principales</i>	10 horas
	<i>Gestion de riesgos</i>	154 horas
	<i>Gestion de la configuracion</i>	30 horas
	<i>Diseño</i>	30 horas
	<i>Desarrollar el código</i>	122 horas
	<i>Depuración preliminar</i>	23 horas
	<i>Realizar pruebas</i>	89 horas
	<i>Revisión final</i>	67 horas

Figura IV.4: Uso de los recursos1

3	Iker	631,5 horas
	<i>Determinar el ambito</i>	9 horas
	<i>Definir recursos preliminares</i>	28 horas
	<i>Listado inicial</i>	12 horas
	<i>Introduccion</i>	14,5 horas
	<i>Funciones principales</i>	10 horas
	<i>Prototipo</i>	20 horas
	<i>Estimacion</i>	77 horas
	<i>Gestion de riesgos</i>	100 horas
	<i>Gestion de la configuracion</i>	30 horas
	<i>Diseño</i>	30 horas
	<i>Desarrollar el código</i>	122 horas
	<i>Desarrollar plan de pruebas</i>	23 horas
	<i>Realizar pruebas</i>	89 horas
	<i>Revisión final</i>	67 horas
4	Jesus	602 horas
	<i>Determinar el ambito</i>	9 horas
	<i>Definir recursos preliminares</i>	28 horas
	<i>Listado inicial</i>	12 horas
	<i>Desarrollo</i>	29 horas
	<i>Introduccion</i>	30 horas
	<i>Funciones principales</i>	50 horas
	<i>Introduccion</i>	30 horas
	<i>Plan de garantia de calidad</i>	60 horas
	<i>Diseño</i>	30 horas
	<i>Desarrollar el código</i>	122 horas
	<i>Desarrollar cursos y ayudas</i>	46 horas
	<i>Realizar pruebas</i>	89 horas
	<i>Revisión final</i>	67 horas
5	Kike	540,5 horas
	<i>Determinar el ambito</i>	9 horas
	<i>Definir recursos preliminares</i>	28 horas
	<i>Listado inicial</i>	12 horas
	<i>Desarrollo</i>	29 horas
	<i>Introduccion</i>	14,5 horas
	<i>Funciones principales</i>	10 horas
	<i>Planificacion temporal</i>	77 horas
	<i>Plan de garantia de calidad</i>	30 horas
	<i>Diseño</i>	30 horas
	<i>Desarrollar el código</i>	122 horas
	<i>Desarrollar plan de pruebas</i>	23 horas
	<i>Realizar pruebas</i>	89 horas
	<i>Revisión final</i>	67 horas

Figura IV.5: Uso de los recursos2

Parte V

Recursos del proyecto

1. Personal

Para el desarrollo del producto se cuenta con un equipo de 5 personas. Este equipo se encargará de seguir todos los pasos para conseguir el producto, es decir, que se encargarán del total desarrollo del mismo. Esto engloba desde la captura de requisitos hasta la puesta en funcionamiento y el mantenimiento del producto.

Al ser un equipo pequeño, no hay un reparto bien definido de tareas, ya que todos los miembros del equipo participan en todas las actividades. El equipo está formado por:

- Jesús Aguirre Pemán (Reichführer)
- Enrique Ballesteros Horcajo
- Jaime Dan Porras Rhee
- Ignacio Iker Prado Rujas (Führer)
- Alejandro Villarín Prieto

El Führer es el encargado de organizar y repartir las tareas y el Reichführer supe al Führer en caso de baja. Todos los miembros del grupo asesoran al Führer, lo que se puede cumplir fácilmente al ser un grupo reducido.

En los diagramas de Gantt vienen reflejadas las tareas que realizará el equipo y las fechas en las que se realizan.

2. Hardware y software

2.1. Componentes reutilizables de software

Aún no se ha pensado en reutilizar componentes de software ya desarrollados, aunque no se descarta su uso en la fase de desarrollo.

2.2. Recursos de entorno

Para elaborar el producto se requieren numerosas herramientas de hardware y software. Estas herramientas son requeridas en todas las fases del proyecto. A continuación se describe cada uno de estos recursos.

En la fase de captura de requisitos se usaron diversos programas. Los documentos se crearon usando \LaTeX , en concreto se usó el programa **TeXWorks** para editar los documentos. También se usó el entorno de programación Eclipse, utilizando el plug-in para \LaTeX **TeXlipse**. Para la realización de presentaciones se ha utilizado el programa **Microsoft Power Point**.

Para el diseño de Diagramas de casos de uso, se utilizó el programa **ArgoUML**.

En la fase de planificación y estimación se han utilizado nuevos programas. Para la edición de documentos \LaTeX se utilizaron los programas mencionados anteriormente. Para la planificación se ha utilizado el programa **Microsoft Project**, y para la estimación se ha utilizado el programa **COCOMO II.2000.4**.

El software se desarrollará sobre el entorno de programación Eclipse, utilizando el lenguaje de programación Java. También se usarán otros programas informáticos para diseñar los elementos de la interfaz de usuario. En cuanto al Hardware, el programa está pensado para que funcione sobre dispositivos como tablets o smartphones, tanto los de la plataforma Android como los de la plataforma Apple. Por lo tanto, para que el programa pueda ser probado y testado, se necesitarán estos dispositivos. Y para utilizar los programas mencionados anteriormente se necesitarán ordenadores personales sobre los que puedan funcionar estos programas. No se requerirán herramientas de hardware adicionales.

3. Lista de recursos

Recursos de personal

Son los encargados de hacer el producto, desde las primeras fases hasta el desarrollo y mantenimiento.

- Jesús Aguirre Pemán (Reichführer)
- Enrique Ballesteros Horcajo
- Jaime Dan Porras Rhee
- Ignacio Iker Prado Rujas (Führer)
- Alejandro Villarín Prieto

El equipo estará disponible en todo momento.

Recursos de entorno

■ TeXWorks y TeXlipse

- **Descripción del recurso:** Programa utilizado para editar los documentos L^AT_EX.
- **Informe de disponibilidad:** El programa es de licencia gratuita y estará disponible en todo momento.
- **Fecha cronológica en la que se requiere el recurso:** El programa será requerido en todo momento para generar la documentación.
- **Tiempo durante el que será aplicado el recurso:** En todo momento.

■ ArgoUML

- **Descripción del recurso:** Programa utilizado para diseñar los diagramas de casos de uso.
- **Informe de disponibilidad:** El programa es de licencia gratuita y estará disponible en todo momento.
- **Fecha cronológica en la que se requiere el recurso:** El programa será requerido al principio para poner los diagramas en el documento de casos de uso.
- **Tiempo durante el que será aplicado el recurso:** Desde el comienzo hasta la primera entrega.

■ Microsoft Power Point

- **Descripción del recurso:** Programa utilizado para realizar presentaciones.
- **Informe de disponibilidad:** El programa está disponible en los laboratorios de la Facultad de Informática.
- **Fecha cronológica en la que se requiere el recurso:** El programa será requerido en cada entrega, para realizar la presentación.
- **Tiempo durante el que será aplicado el recurso:** Desde la primera entrega hasta la última.

■ Microsoft Project

- **Descripción del recurso:** Programa utilizado para hacer la planificación del proyecto.
- **Informe de disponibilidad:** El programa está disponible en los laboratorios de la Facultad de Informática.
- **Fecha cronológica en la que se requiere el recurso:** El programa será requerido para hacer la planificación, es decir, en la segunda entrega.
- **Tiempo durante el que será aplicado el recurso:** Desde la primera entrega hasta la segunda.

■ **COCOMO II.2000.4**

- **Descripción del recurso:** Programa utilizado para hacer la estimación del proyecto.
- **Informe de disponibilidad:** Disponible a través de Softonic. La licencia se consiguió a través de la UCM. También disponible gratis en la página de la USC, web :
http://csse.usc.edu/csse/research/COCOMOII/cocomo_downloads.htm
- **Fecha cronológica en la que se requiere el recurso:** El programa será requerido en la segunda entrega, para realizar la estimación.
- **Tiempo durante el que será aplicado el recurso:** Desde la primera entrega hasta la segunda.

Parte VI

Mecanismos de seguimiento y control

La garantía de calidad y el control de cambios serán profundamente analizados en la próxima entrega de documentación.

1. Garantía de calidad y control

En este apartado comentaremos las revisiones técnicas sobre el producto KIKE-Hostelería ®. Las revisiones formales comenzarán con la fase de codificación.

2. Gestión y control de cambios

Aquí identificaremos, controlaremos y garantizaremos la corrección de la implementación de los cambios, informando de ellos al personal que lo necesite. El control de versiones nos permitirá identificar y gestionar los diversos documentos y versiones del sistema que tengamos

