

Plan de proyecto

Ingeniería del Software

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

28 de Febrero 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	3
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	8
1. Análisis del riesgo	8
2. Estudio de los riesgos	8
3. Plan de gestión del riesgo	8
IV. Planificación temporal	10
1. Estructura de descomposición del trabajo, Planificación temporal .	10
2. Gráfico de Gantt	10
3. Red de tareas	11
4. Tabla de uso de recursos	11
V. Recursos del proyecto	13
1. Personal	13
2. Hardware y software	13
3. Lista de recursos	13

VI.Mecanismos de seguimiento y control	15
1. Garantía de calidad y control	15
2. Gestión y control de cambios	15

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 hosteleria 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 internamente 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.

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, podemos obtener una estimación en esfuerzo, dinero y tiempo para el producto.

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

1. Análisis del riesgo
2. Estudio de los riesgos
3. Plan de gestión del riesgo

Parte IV

Planificación temporal

1. Estructura de descomposición del trabajo, Planificación temporal
2. Gráfico de Gantt

Aquí vemos primero un gráfico de Gantt general de todo el proyecto:

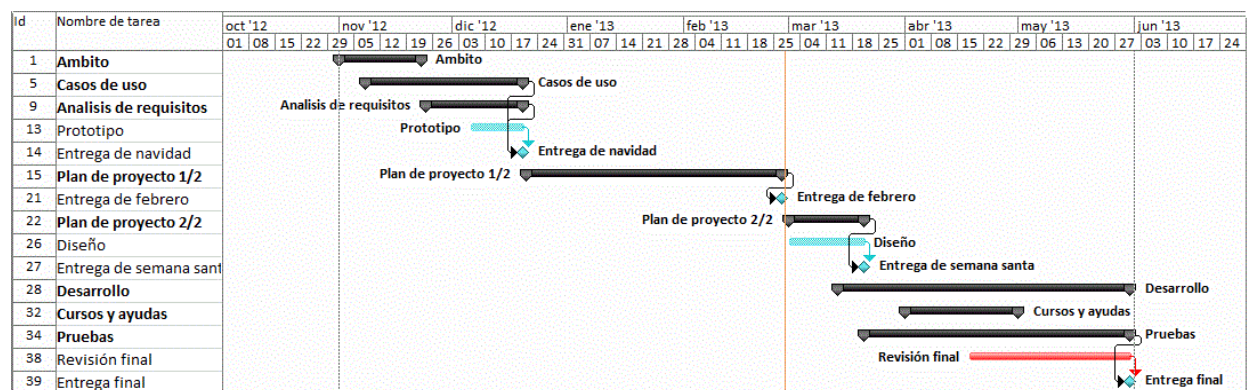


Figura IV.1: Diagrama de Gantt

Aquí tenemos las tareas desglosadas con las fechas:

Y aquí un gráfico de Gantt desglosado con lo hecho hasta la entrega del proyecto:

3. Red de tareas

4. Tabla de uso de recursos

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. Todos los miembros del equipo tienen el mismo papel dentro del mismo.

2. Hardware y software

Para elaborar el producto se requieren numerosas herramientas de hardware y software.

En la fase de captura de requisitos se usaron diversos programas. 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. 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 testeado, se necesitarán estos dispositivos. No se requerirán herramientas de hardware adicionales.

3. Lista de recursos

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 HOSTELERIA. 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

