

Documento de  
**PLANIFICACIÓN**



*Sistema de Administracion y Gestion de RESTaurantes*



v0.2

# ÍNDICE DE CONTENIDO

|   |    |
|---|----|
| Apartado de control de versiones.....   | 3  |
| 1. Declaración de alcance.....          | 4  |
| Objetivos.....                          | 4  |
| Especificación a alto nivel.....        | 4  |
| Viabilidad.....                         | 4  |
| 2. Recursos disponibles.....            | 5  |
| Personal y Organización.....            | 5  |
| Hardware y Software.....                | 5  |
| 3. Estimación de costos.....            | 7  |
| Datos historicos utilizados.....        | 7  |
| Datos para la estimación de costos..... | 7  |
| 4. Planificación temporal.....          | 9  |
| Descripción de tareas.....              | 9  |
| Modelado de requisitos.....             | 9  |
| Análisis.....                           | 9  |
| Diseño.....                             | 9  |
| Implementación.....                     | 10 |
| Prueba.....                             | 10 |
| Diagrama de GANT.....                   | 11 |
| 5. Análisis de los riesgos.....         | 12 |
| Apéndice O.1.....                       | 13 |

## APARTADO DE CONTROL DE VERSIONES

- V0.1: Presentación inicial del documento.
- V0.2: Cambio de plataforma de comunicación al Awacate.



## 1. DECLARACIÓN DE ALCANCE

### Objetivos

En este primer paso, vamos a tratar de completar toda la tarea de gestión de los ingredientes y de los productos ofertados en la carta. A continuación haremos una especificación a alto nivel de los elementos que esperamos llevar a cabo.

### Especificación a alto nivel

El cocinero elaborará la carta una vez cada seis meses, si lo desea, aunque podrá realizar modificaciones puntuales siempre y cuando el restaurante esté cerrado. Así se evitará el riesgo de que un cliente pida un plato que esté siendo modificado.

La carta está dividida en distintas secciones:

- Entradas
- Carnes
- Pescados
- Bebidas
- Postres

Estas secciones serán fijas y no se podrán eliminar de la carta.

A la hora de añadir un nuevo plato a la carta, el cocinero deberá especificar cada uno de los ingredientes que componen el plato, el precio, el gasto de elaboración (asociado a los ingredientes usados) y si es posible adjuntar una foto del mismo. Todos estos detalles del plato podrán ser modificados posteriormente por el cocinero, e incluso se podrán eliminar de la carta si fuese necesario.

Respecto a los ingredientes de cada plato, el cocinero se encargará de ir marcando en el sistema los ingredientes que se van utilizando al cocinar un plato. El sistema también le deberá permitir notificar si un ingrediente se encuentra en mal estado (estos casos se marcarán como pérdidas).

El cocinero y el camarero conjuntamente fijarán el mínimo y el máximo de cada producto. A partir de los productos consumidos y del mínimo que debe haber de existencias el sistema deberá elaborar una lista de necesidades con las cantidades requeridas de cada uno de los productos que se deban comprar. El sistema deberá permitir imprimir dicha lista.

Cuando se recibe el pedido del proveedor el cocinero deberá notificarlo en el sistema, actualizándose la base de datos de ingredientes de este.

### Viabilidad

Para la conclusión de las tareas anteriormente descritas disponemos de, aproximadamente, 30 días. A priori conocemos que tendremos las etapas de Modelado de requisitos, Análisis, Diseño e Implementación. En principio, suponiendo que llevara una media de una semana por cada etapa, dada la embergadura de las funcionalidades que esperamos completar, vemos que el proyecto por cuestiones de tiempo es viable, teniendo en cuenta los recursos de los que dispondremos.

## 2. RECURSOS DISPONIBLES

### **Personal y Organización**

El grupo consta de nueve miembros, los cuales listamos a continuación:

1. Dionisio Ruiz, Jose David
2. Guerrero Martinez, Daniel
3. Guirado Navarro, Samuel
4. Moreno Muñoz, Carlos
5. Muñoz Soria, Gaspar
6. Pérez Lopera, Adrián Victor
7. Rodríguez Lumley, Sergio
8. Sabeg, Nabil
9. Salas Morales, Carlos

Hemos dividido el grupo en tres pequeños subgrupos. Para esta iteración, los grupos son:

#### **Equipo de gestión:**

- Pérez Lopera, Adrián Victor
- Rodríguez Lumley, Sergio
- Sabeg, Nabil

#### **Equipo de Análisis:**

- Guirado Navarro, Samuel
- Moreno Muñoz, Carlos
- Muñoz Soria, Gaspar

#### **Equipo de Implementación:**

- Dionisio Ruiz, Jose David
- Guerrero Martinez, Daniel
- Salas Morales, Carlos

### **Hardware y Software**

Todos los miembros del equipo disponen de un PC portátil, con lo que las reuniones entre los subgrupos se podrán realizar en cualquier parte y cualquier fecha.

Como grupo de desarrollo de bajo costo, nos interesa utilizar software libre, con licencias abiertas y con el respectivo ahorro de recursos. Esto, sin embargo, no es posible en todos los casos.

El software del que dispondremos es: GIMP 2.6, OpenOffice3.1.0, Enterprise Architect 7.5, Microsoft Project 2000, NetBeans6.8, Dia, Gmail, Google Groups, Google Code.

| Nombre                             | Versión | Descripción                      | Campo asociado                             | Licencia    |
|------------------------------------|---------|----------------------------------|--|-------------|
| <b><i>GIMP</i></b>                 | 2.6     | Programa de edición de imágenes. | Gráficos.                                  | GNU         |
| <b><i>OpenOffice</i></b>           | 3.1.0   | Edición de texto.                | Documentación.                             | GNU         |
| <b><i>Enterprise Architect</i></b> | 7.5     | Editor de diagramas UML.         | Modelado de requisitos, análisis y diseño. | Propietario |
| <b><i>Microsoft Project</i></b>    | 2000    | Editor de diagramas de GANT.     | Planificación.                             | Propietario |
| <b><i>NetBeans</i></b>             | 6.8     | IDE de Java y PHP.               | Implementación.                            | CDDL        |
| <b><i>Dia</i></b>                  | 0.97    | Editor de diagramas ligero.      | Planificación.                             | GNU         |
| <b><i>Gmail</i></b>                | 2.0     | Gestor de correo.                | Comunicación.                              | Propietario |
| <b><i>Awacate</i></b>              | Beta    | Gestor y comunicador de grupo.   | Comunicación.                              | LGPL        |
| <b><i>Google Code</i></b>          | Beta    | Servidor SVN.                    | Servidor.                                  | LGPL        |

Sagres



### **3. ESTIMACIÓN DE COSTOS**

#### ***Datos historicos utilizados***

Como referencia en la realización de diagramas y especificaciones en las etapas de Modelado de Requisitos, Análisis y Diseño, tenemos todos los miembros del equipo la experiencia de la asignatura Ingeniería del Software 2 (2009-2010). Con lo que podemos estimar un tiempo necesario para la realización de cada una de las tareas individuales de cada etapa, así como el tiempo de la etapa en sí.

Como experiencia en implementación, algunos miembros del equipo han desarrollado asignaturas de cometido final similar al sistema pedido por el cliente. A pesar de ser experiencias basadas en proyectos de menor tamaño, si nos sirven como estimación del tiempo que emplearemos en el desarrollo y utilización de herramientas.

#### ***Datos para la estimación de costos***

Para valorar el esfuerzo empleado sobre los subsistemas seleccionados, hemos realizado un boceto de un diagrama de casos de uso, con lo que se puede ver la funcionalidad esperada del sistema al completo y la proporción de lo que debemos realizar en esta iteración.

Como podemos comprobar en la figura 1, las funcionalidades que completaremos en esta iteración (Marcadas en Azul celeste, arriba a la derecha, y en Violeta (Arriba)) son el subsistema de gestion de carta y de ingredientes. Todo esto se ha elaborado para hacer una repartición óptima de tareas y una estimación de costos más acertada, sin embargo todo esto es susceptible a cambios.

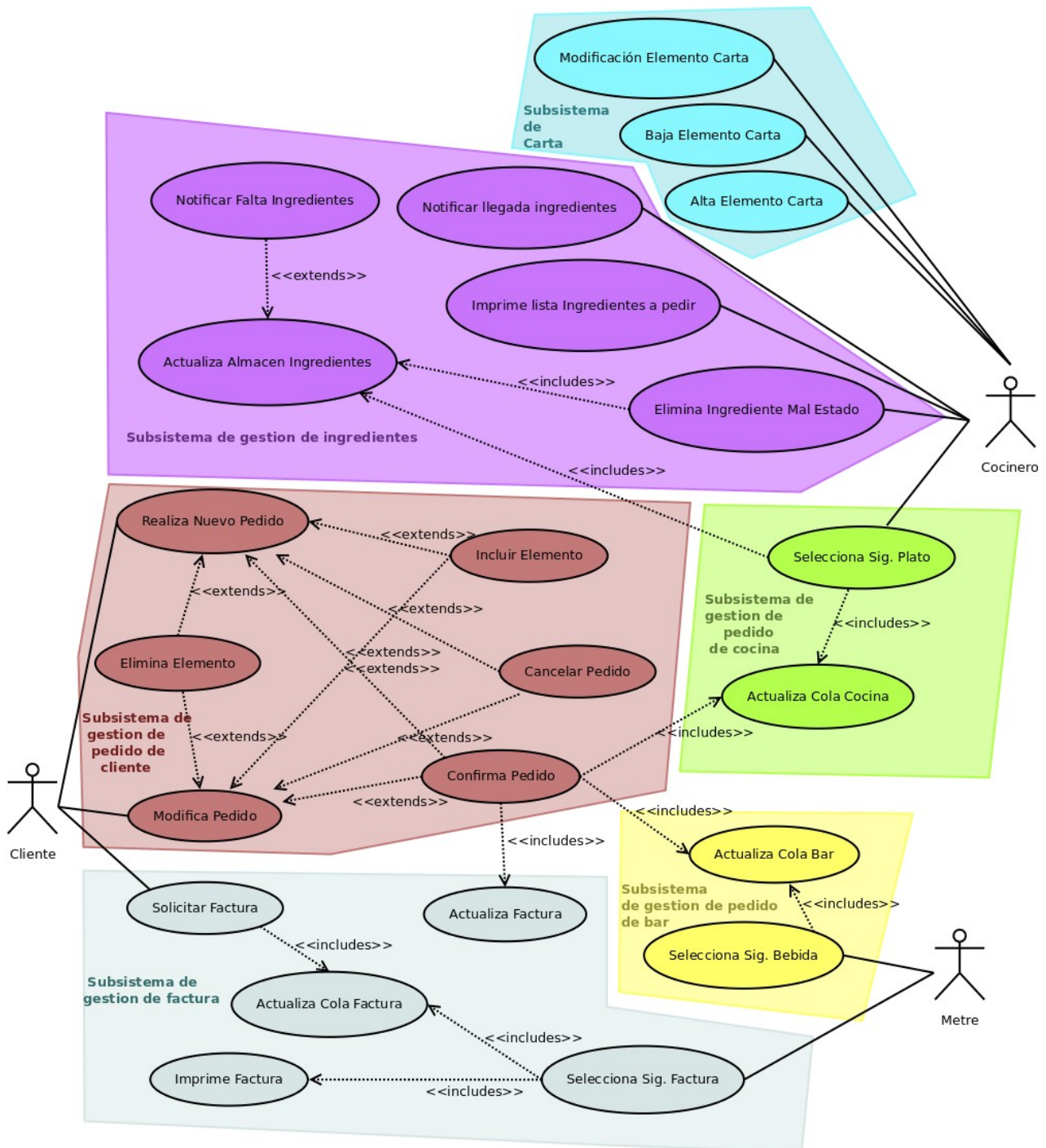


Figura 1. Boceto de diagrama de Casos de Uso de sistema.



## 4. PLANIFICACIÓN TEMPORAL

### *Descripción de tareas*

#### *Modelado de requisitos*

**Tarea 1:** Realizar el modelado de requisitos funcionales, mediante el diagrama de casos de uso del sistema.

**Tarea 2:** Realizar la especificación detallada de los casos de uso, intentando identificar en cada uno de ellos requisitos no funcionales del problema.

**Tarea 3:** Realizar descomposición del sistema en subsistemas funcionales, mediante el diagrama de paquetes funcionales.

**Tarea 4:** Llevar a cabo la identificación de requisitos no funcionales del problema.

**Tarea 5:** Realizar los diagramas de secuencia del sistema. Se realiza un diagrama por cada caso de uso.

**Tarea 6:** Realizar una lista con las operaciones del sistema.

**Tarea 7:** Revisión del modelado de requisitos.

**Tarea 8:** Realizar el documento de modelado de requisitos.

#### *Análisis*

**Tarea 9:** Identificar en el problema las posibles clases, las relaciones que hay entre ellas y sus atributos.

**Tarea 10:** Realizar el diagrama de clases del análisis.

**Tarea 11:** Realizar los contratos de todas las operaciones, obtenidas en los diagramas de secuencia de sistema.

**Tarea 12:** Definir un diagrama de colaboración por cada contrato, teniendo en cuenta el diagrama de clases definido.

**Tarea 13:** Revisión del análisis.

**Tarea 14:** Realizar el documento de análisis.

#### *Diseño*

**Tarea 15:** Decidir el estilo arquitectónico que se adecua mejor a la solución.

**Tarea 16:** Definir el diagrama de clases del diseño inicial, obteniéndolo a partir del diagrama de clases del análisis.

**Tarea 17:** Realizar diagrama de paquetes estructurales. Se descompone el diagrama de clases en subsistemas estructurales, utilizando el diagrama de paquetes obtenido en el modelado de requisitos.

**Tarea 18:** Eliminar dependencias cíclicas del diagrama de paquetes estructurales.

**Tarea 19:** Para cada paquete, revisar su diagrama de clases interno, identificando los servicios requeridos y los ofrecidos (interfaces).

**Tarea 20:** Una vez definidas las interfaces de los paquetes, transformar el diagrama de paquetes en un diagrama de componentes.

**Tarea 21:** Realizar el diagrama de despliegue, identificando la ubicación física de cada subsistema estructural.

**Tarea 22:** Revisión del diseño.

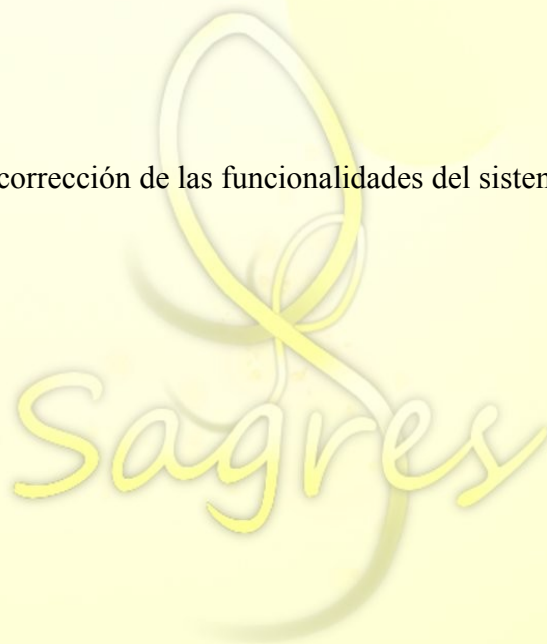
**Tarea 23:** Realizar documento de diseño.

### **Implementación**

**Tarea 24:** Implementación de todas las clases de los subsistemas funcionales.

### **Prueba**

**Tarea 25:** Realizar pruebas de corrección de las funcionalidades del sistema desarrollado.



## Diagrama de GANT

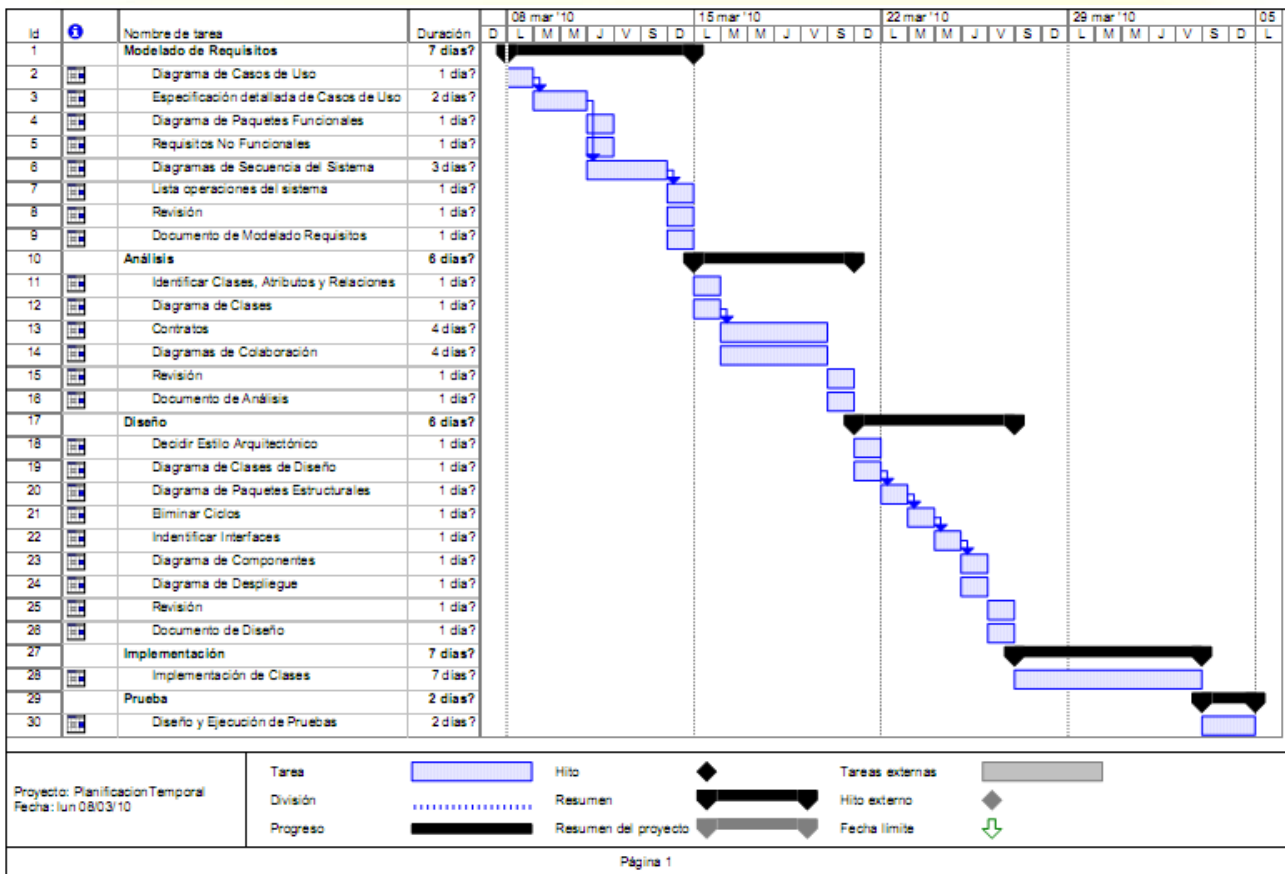


Figura 2. Diagrama de GANT.

## 5. ANÁLISIS DE LOS RIESGOS

Un posible riesgo es que un miembro del grupo falle. Entiendase por fallo que abandone el grupo, que no termine las tareas a tiempo o bien que quede incomunicado con el resto del equipo. En este caso, procederemos a repartir las tareas sobre los miembros restantes del subgrupo en el que se ha producido la falta. Si las tareas faltantes fuesen demasiado voluminosas como para ser cumplidas dentro del plazo estimado, se pedirá ayuda a algún miembro de otro subgrupo, que podría en cualquier caso, prestar la colaboración necesitada.

No consideraremos los riesgos por fallo del servidor SVN, el grupo de contacto o de conexión a internet, puesto que todos los miembros disponemos de una red suficientemente fiable en la Universidad de Granada y los servidores de datos son gestionados por una empresa que oferta soluciones software profesionales sin costo, con suficiente redundancia de datos como para mantener todos los datos seguros.



## APÉNDICE 0.1

- Se ha generado el documento de planificación inicial.





## APÉNDICE 0.2

- Se ha modificado en la página 6 la tabla “TablaDeRecursosSoftware”, se ha cambiado el sistema Google Groups por el sistema Awacate como gestor de comunicaciones del grupo.

