

Documento de PLANIFICACIÓN

Sagres

**Sistema de Administración y
Gestión de REStaurantes**



Gaspar Muñoz Soria
Carlos Moreno Muñoz
Samuel Guirado Navarro

v1.2

ÍNDICE DE CONTENIDO

| | |
|--|-----------|
| Apartado de control de versiones..... | 4 |
| 1. Introducción..... | 5 |
| 2. Declaración de alcance..... | 6 |
| <i>Objetivos.....</i> | <i>6</i> |
| Gestión de pedidos de los clientes (clientes del hotel)..... | 6 |
| Viabilidad..... | 7 |
| <i>Rendimiento.....</i> | <i>7</i> |
| <i>Interfaces Externas.....</i> | <i>7</i> |
| <i>Fiabilidad.....</i> | <i>8</i> |
| <i>Restricciones.....</i> | <i>8</i> |
| 3. Recursos disponibles..... | 9 |
| <i>Personal y Organización.....</i> | <i>9</i> |
| <i>Hardware y Software.....</i> | <i>11</i> |
| Infraestructura de manutención de código y documentos..... | 14 |
| 4. Estimación de costos..... | 15 |
| <i>Datos históricos utilizados.....</i> | <i>15</i> |
| 5. Planificación temporal..... | 16 |
| <i>Descripción de tareas.....</i> | <i>16</i> |
| Planificación..... | 16 |
| Modelado de requisitos..... | 16 |
| Análisis..... | 16 |
| Diseño..... | 17 |
| Implementación..... | 17 |
| <i>Diagrama de Gantt.....</i> | <i>18</i> |
| <i>Hitos.....</i> | <i>19</i> |
| Diagramas de Pert..... | 20 |
| <i>Planificación.....</i> | <i>20</i> |
| Descripción de las tareas..... | 20 |
| Información de la etapa..... | 21 |
| Holguras en actividades..... | 21 |
| <i>Modelado de Requisitos.....</i> | <i>21</i> |
| Descripción de las tareas..... | 21 |
| Información de la etapa..... | 22 |
| <i>Análisis.....</i> | <i>22</i> |
| Descripción de las tareas..... | 22 |
| Información de la etapa..... | 22 |
| Holguras en actividades..... | 22 |
| <i>Diseño.....</i> | <i>23</i> |
| Descripción de las tareas..... | 23 |
| Información de la etapa..... | 24 |
| Holguras en actividades..... | 24 |
| <i>Implementación.....</i> | <i>24</i> |
| Descripción de las tareas..... | 24 |
| Información de la etapa..... | 25 |
| Holguras en actividades..... | 25 |
| 6. Análisis de los riesgos..... | 26 |
| <i>Retrasos.....</i> | <i>26</i> |
| Modelado de Requisitos..... | 26 |
| 7. Formato de la documentación..... | 27 |
| <i>1. Estructura de carpetas.....</i> | <i>27</i> |

| | |
|---|-----------|
| 2. Documentación..... | 28 |
| Significado de las plantillas..... | 28 |
| Reglas de los documentos..... | 29 |
| Consejos prácticos de uso..... | 30 |
| 3. Estructura en los documentos de código..... | 31 |
| Reglas semánticas..... | 31 |
| Denominación de variables..... | 31 |
| Denominación de funciones..... | 32 |
| Denominación de clases..... | 32 |
| Documentación y comentarios..... | 33 |
| Llaves e indentación..... | 33 |
| Apéndice 1.0..... | 34 |
| Apéndice 1.1..... | 35 |
| Apéndice 1.2..... | 37 |



APARTADO DE CONTROL DE VERSIONES

Todas las versiones están especificadas a fondo en el apartado de “Apéndices”, al final de este documento, cada apéndice se corresponde en nombre con su número de versión. Por ejemplo, el “Apéndice 0.1” se corresponde con la versión v0.1. Para ver los cambios realizados sobre cada versión, hay que ir deshaciendo los cambios desde el final.

| Versión | Fecha | Descripción |
|---------|----------|---|
| V1.0 | 14/05/10 | Presentación inicial del documento. |
| V1.1 | 17/05/10 | Retrasos. Nuevo Diagrama Gantt, cambio en hitos. |
| V 1.2 | 23/05/10 | Añadidos Diagramas de Pert, cambios por retrasos, nueva numeración. |

Sagres

1. INTRODUCCIÓN

Este documento describe la organización y planificación del grupo TouchTeam para el proyecto de Ingeniería del Software 3, al cual hemos denominado **SAGRES** (Sistema de Administración y Gestión de **RE**staurantes).

Este proyecto se realizará durante el curso 2009/2010 para aprender el desarrollo del ciclo de vida del software. El proyecto se desarrolla durante tres iteraciones.



2. DECLARACIÓN DE ALCANCE

Objetivos

En la segunda iteración y tras algunos cambios con respecto a la planificación inicial de la asignatura se implantaron los subsistemas de gestión de pedidos de los clientes, y el subsistema de facturación, quedando el sistema final prácticamente acabado de cara al usuario del restaurante.

Para finalizar el ciclo de desarrollo en esta iteración se contempla:

- Gestión de Estadísticas.
- Gestión de los pedidos de los clientes (hotel).

Se contempla en principio la posibilidad del refinamiento del diseño y la implementación como consecuencia de los posibles errores o capacidades incompletas aparecidas en el software resultante de iteraciones anteriores. Esta decisión se ha tomado teniendo en cuenta que estos errores podrían comprometer el resultado de ésta última iteración, y la carga de trabajo de los componentes de los grupos de diseño e implementación (ahora en implementación y planificación respectivamente) actualmente. Si con el avance del proyecto se considera que es inviable la corrección de errores, porque se considere imposible el cumplir los plazos previstos se modificará la planificación y se informará puntualmente de todos los cambios..

Como la especificación a alto nivel de los subsistemas se realizó en la primera iteración, ahora recordaremos solo la especificación perteneciente a los subsistemas relativos a esta iteración.

Gestión de pedidos de los clientes (clientes del hotel)

Cuando se realizan los pedidos se dividen en dos partes: comidas, se pone en la cola de platos del cocinero y Bebidas, se ponen en la cola de bebidas del bar. Cada uno funciona de manera independiente y cuando esté preparado (platos o bebidas) se comunica al camarero para servirlo.

Hay dos tipos de clientes para el restaurante: clientes alojados en el hotel que realizan pedidos desde sus habitaciones y clientes que lleguen al restaurante.

Los clientes del hotel reciben sus pedidos completos mientras que los clientes en el restaurante se les puede servir las bebidas antes de la comida, el pago se puede realizar en efectivo o tarjeta de crédito en caso de los clientes en el restaurante o mediante tarjeta de crédito o tarjeta del hotel en caso de clientes alojados en el hotel.

Los clientes pueden modificar sus pedidos siempre y cuando el cocinero no empezó a prepararlo, en caso contrario solo pueden añadir o eliminar solamente bebidas a su pedido..

Gestión de estadísticas

El sistema generará algunas estadísticas útiles para el restaurante: estadísticas de facturación, platos mas vendidos o menos vendidos, bebidas más vendidas o menos vendidas.

Viabilidad

Como resultado de la segunda iteración realizaremos una estimación de la viabilidad de la

planificación correspondiente a esta tercera iteración. Durante la segunda iteración se produjo un importante en Análisis debido a varias causas, entre ellas, que el Diseño tenía pocos días planificados con respecto al trabajo que realmente requería y éste lógicamente se alargó, lo que llevó a que el equipo de implementación no tuviera apenas tiempo de realizar su labor para la fecha estimada.. El principal cambio con respecto a iteraciones anteriores será precisamente éste, el planificar un menor tiempo para las fases de modelado de requisitos y análisis y dar más días para que el diseño de los subsistemas correspondientes se haga adecuadamente y dejando al equipo de implementación un tiempo razonable para trabajar.

Durante esta iteración también podremos ahorrarnos el tiempo de aprendizaje de las herramientas de comunicación entre el grupo, lo cual debería corresponderse con una comunicación mas rápida y fluida entre los integrantes y entre los grupos de trabajo.

Para especificar el proceso y la duración de las etapas de desarrollo de la tercera iteración se ha utilizado al igual que en las anteriores un diagrama de Gantt, donde se especificarán los hitos de entrega para los distintos equipos, y que también se modificará según vayan los objetivos siendo alcanzados o no por los miembros del equipo.

Hay que tener en cuenta que la implementación se alargó una semana más en la segunda iteración por las causas anteriormente descritas, esto ha hecho que el equipo de planificación tenga trabajos de implementación y planificación solapados durante la primera semana. Esto no supone mayor inconveniente salvo el pequeño retraso en la entrega de este documento por una falta de entendimiento.

Rendimiento

Como comentaron en planificaciones previas la aplicación no cuenta con complejos procesos de cálculo, por tanto las limitaciones al rendimiento pueden venir en relación a la gestión de la Base de datos. La Base de Datos puede afectar al rendimiento en caso de que el volumen de datos sea superior a lo que la base de datos soporta. Esto podría afectar, a priori en nuestro caso a la velocidad de inicio del sistema, pero en la experiencia durante estas dos primeras iteraciones hemos observado que dicha velocidad no se ve damnificada significativamente.

También cabe decir que en un principio se prevé que dicha velocidad de inicio y la agilidad de la aplicación se vea resentida ligeramente en los clientes que usan el sistema desde sus habitaciones.

Interfaces Externas

Aunque en el caso de una aplicación para gestión de servicios de restauración pudiera ser conveniente la comunicación con sistemas externos de pago, para poder permitir el pago con tarjeta de crédito, esta eventualidad queda fuera del alcance del desarrollo de esta aplicación por tanto no sera considerado.

Fiabilidad

Se toma la fiabilidad de la aplicación como uno de los aspectos clave en el desarrollo de la

aplicación. Se toman distintas medidas y métodos en el desarrollo para conseguirlo:

- Control exhaustivo de los datos que contiene la base de datos, es decir, evitar rotundamente inconsistencias o repetición de datos, para tener calidad en los datos contenidos.
- Interfaz de usuario segura ante errores del usuario, pidiendo siempre confirmaciones antes de realizar operaciones que impliquen cambios en los datos almacenados.
- Consistencia de las operaciones y la base de datos debido a fallos imprevistos como caídas de luz, error del sistema..

Restricciones

Existe una restricción temporal, debiendo estar el sistema totalmente operativo y con todos los requisitos deseados por el cliente el día 4 de Junio. Los diferentes hitos referentes a cada proceso se especifican en el diagrama de Gantt correspondiente a esta iteración, también incluido en este documento en su primera versión.



3. RECURSOS DISPONIBLES

Personal y Organización

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

1. Dionisio Ruiz, Jose David (j2dr@correo.ugr.es)
2. García Sánchez, Ángel (pilli@correo.ugr.es)
3. Guerrero Martínez, Daniel (lein@correo.ugr.es)
4. Guirado Navarro, Samuel (sgn88@correo.ugr.es)
5. Moreno Muñoz, Carlos (camomu@correo.ugr.es)
6. Muñoz Soria, Gaspar (gmunozs@correo.ugr.es)
7. Pérez Lopera, Adrián Víctor (elpelu@correo.ugr.es)
8. Rodríguez Lumley, Sergio (lumley@correo.ugr.es)
9. Sabeg, Nabil (nsabeg@correo.ugr.es)
10. Salas Morales, Carlos (csalasm@correo.ugr.es)

Hemos dividido el grupo en tres pequeños subgrupos, (A, B, C). Estos son:

Equipo C:

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

Equipo A:

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

Equipo B:

- Dionisio Ruiz, Jose David
- García Sánchez, Ángel
- Guerrero Martínez, Daniel
- Salas Morales, Carlos

La división de tareas para las tres iteraciones es la siguiente:

- **Iteración 1:** Planificación (C), Análisis y Diseño (A), Implementación (B)
- **Iteración 2:** Planificación (B), Análisis y Diseño (C), Implementación (A)
- **Iteración 3:** Planificación (A), Análisis y Diseño (B), Implementación (C)

En esta etapa, como estaba previsto tenemos cuatro miembros para la fase de análisis y diseño para

que el equipo de implementación reciba un diseño adecuado y le facilite su trabajo. También se contempla la posibilidad de que algún miembro de este equipo se sume a la implementación en caso de que fuese necesario.



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.

| Nombre y apellido | Marca | procesador | Ram | Disco Duro | Sistema operativo |
|-----------------------------------|-----------------------|---------------------------------|------------------|---------------------------------|--|
| Carlos Salas Morales | Pórtatil ASUS G50V | Core 2 Duo P8600, 2.4 GHz | 4 GB DDR 800 MHz | 320 GB | Windows Vista Ultimate |
| Samuel Guirado Navarro | Sony VAIO VGN-FE28H | Intel Core Duo 1,6 GHz T2200 | 1 GB DDR | 160GB | Windows XP sp3 |
| Adrián Víctor Pérez Lopera | Asus Z92J | Intel Core Duo T2250 1.7ghz | 2 GB DDR2 | Hitachi Travelstar 4K120 100 GB | Windows Vista |
| Gaspar Muñoz Soria | Airis N1005 | Intel Core2Duo 1,66 Ghz | 1 GB RAM | 100 GB Hd | Windows Xp |
| Carlos Moreno Muñoz | HP | Intel Core 2 Duo T7250 2 GHz | 2 GB RAM | 500 GB | Windows Vista |
| Daniel Guerrero Martínez | HP DV3550es | Intel Core 2 Duo 2.66GHz | 4GB RAM | 320GB | Windows 7/Ubuntu 9.10 |
| Sergio Rodríguez Lumley | HP DV3550es | Intel Core 2 Duo 2.66GHz | 4GB RAM | 320GB | Windows 7/Ubuntu 9.10 |
| Sabeg Nabil | Compaq presario 740es | Intel Pentium dual_core 1,7 GHz | 2GB RAM | 160 GB | Windows 7 |
| Ángel Luis García Sánchez | Dell Studio XPS 13 | Intel Core 2 Duo P8700 | 4 GB RAM | 500 GB HDD | Archlinux 64 bits y Windows Professional 64 bits |
| Jose David Dionisio Ruiz | Sony vaio fw22e | Core 2 Duo P8400 2,26 Ghz | 4 GB RAM | 220 GB | Windows 7 64 bits |

El grupo dispone de un servidor remoto privado para realizar las pruebas necesarias de conexión a servidores y para construir el programa deseado. El Sistema de Gestión de Bases de Datos es MySQL 5.0.51. El driver será cualquiera compatible con la versión 5 de MySQL. El driver hay que

descargarlo de la web de MySQL. El servidor que corre el SGBD es un Intel Celeron 1.2, con 512 MB de RAM. El sistema operativo es Ubuntu Hardy Heron.

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, OpenProject, NetBeans6.8, Dia, Gmail, Awacate, Google Code, TortoiseSVN, Axure.

| Nombre | Versión | Descripción | Campo asociado | Licencia |
|------------------------------------|-------------|----------------------------------|--|-------------|
| <i>GIMP</i> | 2.6 | Programa de edición de imágenes. | Gráficos. | GNU |
| <i>OpenOffice</i> | 3.1.0/3.2.0 | Edición de texto. | Documentación. | GNU |
| <i>Enterprise Architect</i> | 7.5 | Editor de diagramas UML. | Modelado de requisitos, análisis y diseño. | Propietario |
| <i>OpenProject</i> | 1.4-2 | Editor de diagramas de Gantt. | Planificación. | GNU |
| <i>NetBeans</i> | 6.8 | IDE de Java y PHP. | Implementación. | CDDL |
| <i>Dia</i> | 0.97 | Editor de diagramas ligero. | Planificación. | GNU |
| <i>Gmail</i> | 2.0 | Gestor de correo. | Comunicación. | Propietario |
| <i>Awacate</i> | Beta | Gestor y comunicador de grupo. | Comunicación. | LGPL |
| <i>Google Code</i> | Beta | Servidor SVN. | Servidor. | LGPL |
| <i>TortoiseSVN</i> | 1.6.7 | Cliente SVN | Control de ficheros | LGPL |
| <i>Axure</i> | 5.6 | Herramienta de prototipado. | Diseño. | Propietario |

Tabla 1. Tabla de recursos de software.

A continuación vamos a mostrar los campos en los que generalmente asociaremos el software utilizado y la razón de su selección. Los campos que a continuación se mostrarán, no son los únicos que los utilizarán, sin embargo si será más común que aparezcan ahí:

1. Planificación

- a) **Diagramas de Gantt:** Utilizaremos OpenProject, al ser de código abierto, nos ahorra la tarea del pago de licencias, disminuyendo el costo asociado al proyecto. Es también multiplataforma, con lo que no hay restricción a la hora de elegir un sistema operativo. Además, es plenamente compatible con Microsoft Project 2000, herramienta de la que dispusimos inicialmente, con lo que la portación de proyectos es instantánea.
- b) **Comunicación con el grupo:** toda la comunicación se gestiona a través del sistema Awacate de la ugr (<http://awacate.ugr.es>). Con ella, todos los miembros del grupo nos enviamos mensajes, disponemos de un foro de discusión, un apartado para subir documentos en común y un sistema de asignación de tareas, a través del cual se puede llevar un seguimiento continuado de tareas realizadas (El porcentaje de acabado y las horas dedicadas a ellas). En el apartado de documentos se seguirá un estricto formato de árbol de carpetas, lo cual detallaremos en el apartado 7 de este mismo documento (“Formato de la Documentación”).
- c) **Documentación escrita:** esto será realizado con el programa OpenOffice3. Elegimos este editor de textos puesto que es libre, de código abierto y multiplataforma. Además es de la calidad suficiente para la generación y gestión de toda la documentación necesaria para el proyecto. Otra razón de la elección de este formato es que nuestro sistema de control de versiones queda muy simplificado gracias a que SVN (nuestro sistema de repositorios) puede leer este formato (.odt) y, por tanto, notificar las diferencias entre las versiones que maneja. Con este programa también es posible portar el documento a formato PDF. Para la lectura de este formato, el equipo dispone de diversas herramientas como son FoxitReader, Adobe Reader o Evince.

2. Análisis y Diseño

- a) **Prototipado de diagramas:** utilizaremos la herramienta Dia, por su facilidad a la hora de representar diagramas UML. Además es libre, de código abierto (Licencia GNU) y multiplataforma, lo que significa que todos podremos compartir los diagramas y los bocetos de forma rápida y sencilla.
- b) **Diagramas UML:** para la realización de los diagramas UML finales, utilizaremos la herramienta Enterprise Architect 7.5, ya que cumple con todos los requisitos y la calidad que esperamos de un software de este tipo. Además tiene la posibilidad de generar código a partir de sus diagramas, lo que puede reducir el trabajo a los miembros del grupo de implementación.
- c) **Diseño de la interfaz de usuario:** para la realización de bocetos y acabado de diseño, utilizaremos GIMP 2.6, editor de dibujo pixelado libre, de código abierto (GNU) y multiplataforma. Presenta plena compatibilidad para todos los formatos estándar de imágenes y con los proyectos de otros editores (como Photoshop).

3. Implementación

- a) **Edición de prototipos:** utilizaremos en principio el programa Axure 5.6, propietario, con licencia de estudiante para la UGR. Esta herramienta nos permite realizar en apenas unas horas un prototipo visual con una apariencia visual muy similar al resultado final y con una aparente funcionalidad (que no será real). Esto nos permitirá avanzar rápidamente por el diseño de la interfaz de usuario.

- b) **Generación de código:** nuestra elección, tanto para Java como PHP como bien cualquier otro lenguaje soportado, es NetBeans6.8. Este IDE libre, de código abierto y multiplataforma es líder en su sector por la calidad de su entorno y su estricta regla de seguir con los estándares abiertos tanto en la generación de código como en la generación de documentación y bibliotecas utilizadas. Otra posible elección de alto nivel podría ser Eclipse Galileo 3.5.1, sin embargo este IDE no es tan conocido por los integrantes del grupo, con lo que habría que tener en cuenta un tiempo de aprendizaje y acomodación al mismo.
- c) **Compartición de código:** utilizamos una herramienta cliente SVN, TortoiseSVN. Ésta es libre, de código abierto y multiplataforma, con lo que solo será necesario aprender a utilizar una herramienta. Su sencilla interfaz y sus múltiples funcionalidades convierten esta herramienta como clara elección a la hora de compartir código desde Windows. Además es plenamente integrable en NetBeans6.8. Para linux tenemos también una versión de TortoiseSVN, o bien de otros como KDEsvn.

Infraestructura de manutención de código y documentos

Para compartir y mantener el código seguro, utilizamos un sistema de repositorio basado en SVN. El servidor SVN se encuentra en Google Code, una solución libre y de código abierto, puesta para los desarrolladores de proyectos de código abierto. Esto significa que no tenemos ningún costo asociado a este servicio, sin embargo disponemos de un servicio de calidad profesional, puesto que Google pone a nuestra disposición suficientes servidores como para asegurar una efectiva redundancia, con la velocidad de actualización y conexión que sólo una empresa del tamaño de Google pueden ofrecer. El acceso a dicho servidor es libre y abierto, basta con conectar con el servidor:

<http://touchteam.googlecode.com/svn/trunk>

No son necesarios ni usuario ni contraseña. Esto permite que cualquier persona pueda llevar cuenta de cómo avanza el proyecto, sin embargo para tener privilegios para realizar cambios y subirlos al servidor, sí que es necesario pertenecer al grupo.

A pesar de este control, es posible que algún día (seguramente en el que más necesario sea) el servidor se encuentre con dificultades técnicas. Como regla general, el equipo mantiene una copia en disco, con lo que se puede compartir rápidamente a través del sistema de Awacate o bien por e-mail convencional. Los documentos tienen una doble redundancia, ya que cada versión final se sube al sistema de documentos de Awacate, conservando todos los antiguos.

4. ESTIMACIÓN DE COSTOS

Datos históricos 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.

Ademas se incluye la experiencia obtenida en la dos primeras iteraciones de este proyecto.



5. PLANIFICACIÓN TEMPORAL

Descripción de tareas

Para cada una de las tareas se ha asociado el tiempo estimado de dedicación en horas por los miembros del grupo. Esta estimación se ha hecho en relación a los datos y la experiencia obtenidos en la 1ª Iteración y la 2ª Iteración. Cabe destacar que la jornada laboral, es decir, un día de trabajo se ha valorado en aproximadamente 8 horas, ya que debemos tener en cuenta el tiempo dedicado a otras actividades como asistir a clases de teoría, otras prácticas, tiempo de ocio, etc...

Planificación

Tarea 2: Elaboración del diagrama de Gantt (4 horas)

Tarea 3: Redacción del documento de planificación (4 horas)

Tarea 4: Diagramas de Pert. (6 horas)

Tarea 5: Diseño de Pruebas (5 horas)

Tarea 6: Revisión Modelado Requisitos (2 horas)

Tarea 7: Revisión Análisis. (3 horas)

Tarea 8: Revisión Diseño. (5 horas)

Tarea 9: Revisión de la implementación y ejecución de pruebas (9 horas)

Tarea 10: Elaboración de manuales de instalación (10 horas)

Tarea 11: Revisión del documento de planificación (Toda la etapa) (15 horas)

Modelado de requisitos

Tarea 13: Identificación de los casos de uso (1 horas)

Tarea 14: Elaboración del diagrama de casos de uso (1 horas)

Tarea 15: Realizar la especificación detallada de los casos de uso. (1 horas)

Tarea 16: Identificación de los subsistemas funcionales (1 hora)

Tarea 17: Elaboración del diagrama de paquetes para subsistemas funcionales (1 horas)

Tarea 18: Identificación y descripción de los requisitos no funcionales del sistema (1 horas)

Tarea 19: Identificación y descripción de los requisitos no funcionales de los casos de uso (1 horas)

Tarea 20: Elaboración de los diagramas de secuencia del sistema (3 horas)

Tarea 21: Identificación de las operaciones del sistema (0,5 horas)

Tarea 22: Redacción del documento de Modelado de Requisitos (3 horas)

Análisis

Tarea 24: Descripción de clases conceptuales, relaciones entre ellas y atributos asociados (2 horas)

Tarea 25: Elaboración del diagrama de clases del análisis (3 horas)

Tarea 26: Descripción de contratos del sistema (4 horas)

Tarea 27: Elaboración de diagramas de colaboración para cada contrato (6 horas)

Tarea 28: Redacción del documento de Análisis (5 horas)

Diseño

Tarea 30: Descripción de la arquitectura software (2 horas)

Tarea 31: Descomposición del sistema en subsistemas (1 horas)

Tarea 32: Elaboración del diagrama de clases del diseño (3 horas)

Tarea 33: Elaboración del diagrama de paquetes estructurales (6 horas)

Tarea 34: Elaboración del diagrama de componentes (2 horas)

Tarea 35: Elaboración del diagrama de despliegue (3 horas)

Tarea 36: Elaboración de los diagramas de colaboración del diseño (8 horas)

Tarea 37: Diseño de la interfaz de usuario (6 horas)

Tarea 38: Diseño de la base de datos (1 horas)

Tarea 39: Redacción del documento de Diseño (10 horas)

Implementación

Tarea 41: Aprendizaje Java (7 horas)

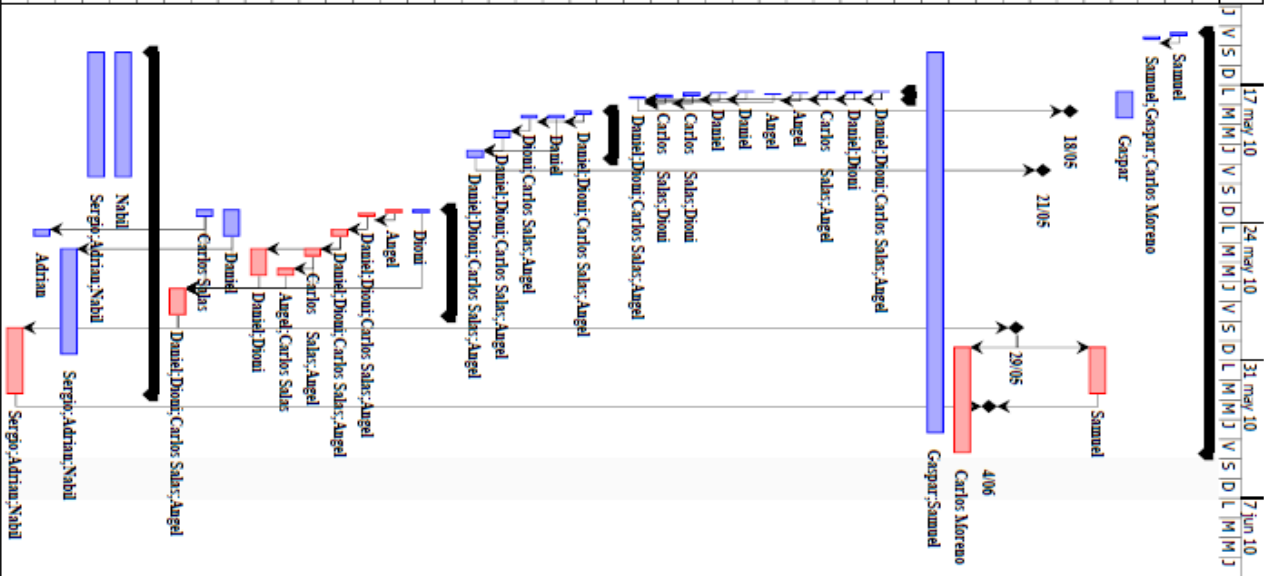
Tarea 42: Aprendizaje de HTML, CSS, JavaScript, MySQL y PHP (7 horas)

Tarea 43: Implementación de las interfaces (web y de escritorio) (20 horas)

Tarea 44: Implementación de la base de datos (1 hora)

Tarea 45: Implementación completa de los subsistemas (30 horas)

| | Nombre | Duración | Inicio | Terminado | Predicadores |
|----|---|-------------|----------------|----------------|-------------------|
| 1 | Planificación | 22 days? | 14/05/10 8:00 | 4/06/10 17:00 | |
| 2 | Elaboración del diagrama de Gantt | 0.5 days? | 14/05/10 8:00 | 14/05/10 13:00 | |
| 3 | Redacción del documento de planificación (3ª Iteración) | 0.5 days? | 14/05/10 13:00 | 14/05/10 17:00 | 2 |
| 4 | Elaboración de los diagramas de Pert | 2 days? | 17/05/10 8:00 | 18/05/10 17:00 | |
| 5 | Diseño de pruebas | 3 days? | 30/05/10 8:00 | 1/06/10 17:00 | 8 |
| 6 | Revisión y entrega del documento de Modelado de Requisitos | 1 day? | 18/05/10 8:00 | 18/05/10 17:00 | 22 |
| 7 | Revisión y entrega del documento de Análisis | 1 day? | 21/05/10 8:00 | 21/05/10 17:00 | 28 |
| 8 | Revisión y entrega del documento de Diseño | 1 day? | 29/05/10 8:00 | 29/05/10 17:00 | 39 |
| 9 | Revisión ejecución de pruebas y entrega de la implementación | 3 days? | 2/06/10 8:00 | 4/06/10 17:00 | 5-45 |
| 10 | Elaboración de manuales de instalación | 6 days? | 30/05/10 8:00 | 4/06/10 17:00 | 8 |
| 11 | Revisión del documento de planificación (3ª Iteración) | 20 days? | 15/05/10 8:00 | 3/06/10 17:00 | |
| 12 | Modelado de Requisitos | 1 day? | 17/05/10 8:00 | 17/05/10 17:00 | |
| 13 | Identificación de los casos de uso | 0.125 days? | 17/05/10 8:00 | 17/05/10 9:00 | |
| 14 | Elaboración del diagrama de casos de uso | 0.125 days? | 17/05/10 9:00 | 17/05/10 10:00 | 13 |
| 15 | Descripción de los casos de uso | 0.125 days? | 17/05/10 9:00 | 17/05/10 10:00 | 13 |
| 16 | Identificación de los subsistemas funcionales | 0.125 days? | 17/05/10 10:00 | 17/05/10 11:00 | 14 |
| 17 | Elaboración del diagrama de paquetes para subsistemas funcionales | 0.125 days? | 17/05/10 11:00 | 17/05/10 13:00 | 16 |
| 18 | Identificación y descripción de los requisitos no funcionales del sistema | 0.125 days? | 17/05/10 8:00 | 17/05/10 9:00 | |
| 19 | Identificación y descripción de los requisitos no funcionales específicos de los casos de uso | 0.375 days? | 17/05/10 10:00 | 17/05/10 11:00 | 15 |
| 20 | Elaboración de los diagramas de secuencia del sistema | 0.375 days? | 17/05/10 10:00 | 17/05/10 14:00 | 15 |
| 21 | Identificación de las operaciones del sistema | 0 days? | 17/05/10 14:00 | 17/05/10 14:00 | 20 |
| 22 | Redacción del documento de Modelado de Requisitos | 0.375 days? | 17/05/10 14:00 | 17/05/10 17:00 | 17:18:19:21 |
| 23 | Análisis | 3 days? | 18/05/10 8:00 | 20/05/10 17:00 | |
| 24 | Descripción de clases conceptuales, relaciones entre ellas y atributos asociados | 0.5 days? | 18/05/10 8:00 | 18/05/10 13:00 | |
| 25 | Elaboración del diagrama de clases del análisis | 0.5 days? | 18/05/10 13:00 | 18/05/10 17:00 | 24 |
| 26 | Descripción de contornos de las operaciones del sistema | 0.5 days? | 18/05/10 13:00 | 18/05/10 17:00 | 24 |
| 27 | Elaboración de los diagramas de colaboración para cada contorno | 1 day? | 19/05/10 8:00 | 19/05/10 17:00 | 26 |
| 28 | Redacción del documento de Análisis | 1 day | 20/05/10 8:00 | 20/05/10 17:00 | 25:27 |
| 29 | Diseño | 6 days? | 23/05/10 8:00 | 28/05/10 17:00 | |
| 30 | Descripción de la arquitectura software | 0.5 days? | 23/05/10 8:00 | 23/05/10 13:00 | |
| 31 | Descomposición del sistema en subsistemas | 0.5 days? | 23/05/10 8:00 | 23/05/10 13:00 | |
| 32 | Elaboración del diagrama de clases del diseño | 0.5 days? | 23/05/10 13:00 | 23/05/10 17:00 | 31 |
| 33 | Elaboración del diagrama de paquetes estructurales | 1 day | 24/05/10 8:00 | 24/05/10 17:00 | 32 |
| 34 | Elaboración del diagrama de componentes | 1 day? | 25/05/10 8:00 | 25/05/10 17:00 | 33 |
| 35 | Elaboración del diagrama de despliegue | 1 day? | 26/05/10 8:00 | 26/05/10 17:00 | 34 |
| 36 | Elaboración de los diagramas de colaboración del diseño | 2 days? | 23/05/10 8:00 | 24/05/10 17:00 | 33 |
| 37 | Diseño de la interfaz de usuario | 2 days? | 23/05/10 8:00 | 24/05/10 17:00 | |
| 38 | Diseño de la base de datos | 1 day? | 23/05/10 8:00 | 23/05/10 17:00 | |
| 39 | Redacción del documento de Diseño | 2 days? | 27/05/10 8:00 | 28/05/10 17:00 | |
| 40 | Implementación | 18 days? | 15/05/10 8:00 | 1/06/10 17:00 | 30:34:35:36:37... |
| 41 | Aprendizaje Java | 7 days? | 15/05/10 8:00 | 21/05/10 17:00 | |
| 42 | Aprendizaje de HTML, CSS, JavaScript, MySQL y PHP | 7 days? | 15/05/10 8:00 | 21/05/10 17:00 | |
| 43 | Implementación de las interfaces (web) y de escritorio | 6 days? | 25/05/10 8:00 | 30/05/10 17:00 | 37 |
| 44 | Implementación de la base de datos | 1 day? | 24/05/10 8:00 | 24/05/10 17:00 | 38 |
| 45 | Implementación completa de los subsistemas | 4 days? | 29/05/10 8:00 | 1/06/10 17:00 | 39 |



Hitos

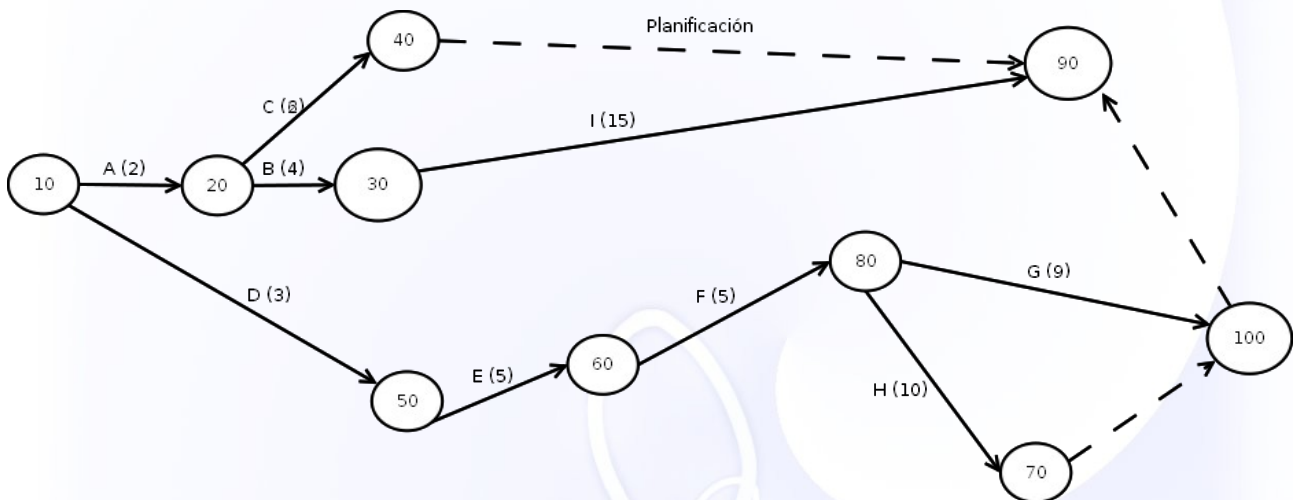
| Fecha | Descripción |
|----------|---|
| 14/05/10 | Presentación del documento de Planificación |
| 18/05/10 | Modelado de Requisitos |
| 21/05/10 | Análisis |
| 29/05/10 | Diseño |
| 04/06/10 | Implementación |

The logo for Sagres features a stylized, flowing script. Above the word 'Sagres' is a large, elegant loop that forms a partial circle, resembling a stylized 'S' or a decorative flourish. The word 'Sagres' itself is written in a cursive, handwritten style.

DIAGRAMAS DE PERT

El diagrama PERT es una representación gráfica de las relaciones entre las tareas del proyecto que permite calcular los tiempos del proyecto de forma sencilla. A continuación veremos los diagramas correspondientes a nuestras etapas dentro de la iteración en la planificación inicial. Nuestra unidad de tiempo serán horas.

Planificación



Descripción de las tareas

| Tarea | Tiempo | Descripción |
|-------|--------|--|
| A | 2 | Diagrama de Gantt |
| B | 4 | Elaboración del documento de planificación. |
| C | 6 | Elaboración de los diagramas de Pert |
| D | 2 | Revisión del modelado de requisitos |
| E | 3 | Revisión del análisis |
| F | 5 | Revisión del diseño |
| G | 5 | Diseño de pruebas |
| H | 9 | Revisión de la implementación y ejecución de pruebas |
| I | 15 | Revisiones del documento de planificación |
| J | 10 | Elaboración de manuales de instalación y usuario. |

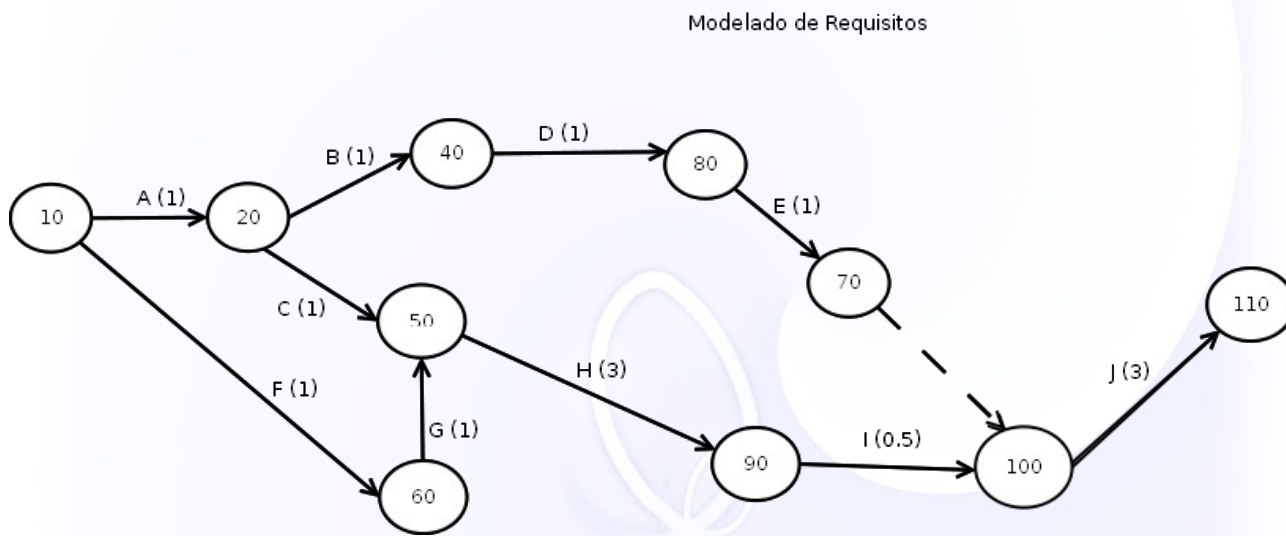
Información de la etapa

- Duración de la etapa: 22 días

Holguras en actividades

- Para realizar las actividad C hay una holgura de 17 unid. tiempo.

Modelado de Requisitos



Descripción de las tareas

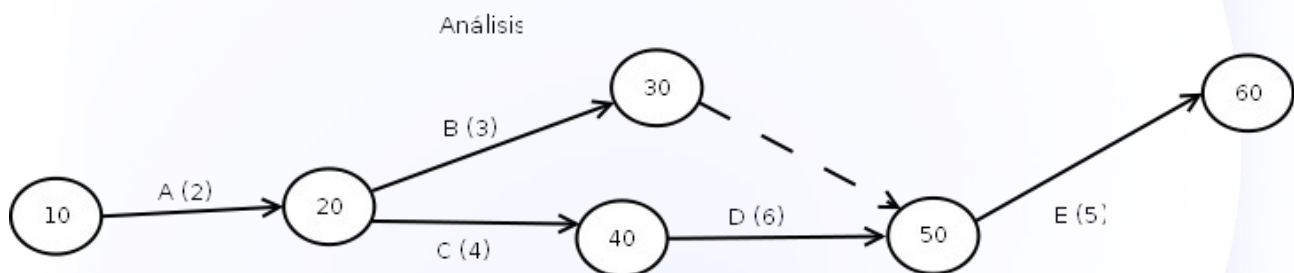
| Tarea | Tiempo | Descripción |
|-------|--------|--|
| A | 1 | Identificación de Casos de Uso. |
| B | 1 | Elaboración del diagrama de casos de uso. |
| C | 1 | Descripción de los casos de uso |
| D | 1 | Identificación de subsistemas funcionales |
| E | 1 | Diag. Paquetes para subsistemas funcionales. |
| F | 1 | Identificación y descripción de los requisitos no funcionales. |
| G | 1 | Identificación y descripción de los requisitos no funcionales de los casos de uso. |
| H | 3 | Elaboración de los diagramas de secuencia del sistema. |
| I | 0,5 | Identificación de las operaciones del sistema. |

| | | |
|---|---|--|
| J | 3 | Redacción del documento de Modelado de Requisitos. |
|---|---|--|

Información de la etapa

- Duración de la etapa: 1 día

Análisis



Descripción de las tareas

| Tarea | Tiempo | Descripción |
|-------|--------|--|
| A | 2 | Identificar clases, atributos y relaciones |
| B | 3 | Diagrama de clases |
| C | 4 | Contratos |
| D | 6 | Diagramas de colaboración |
| E | 5 | Documento de análisis |

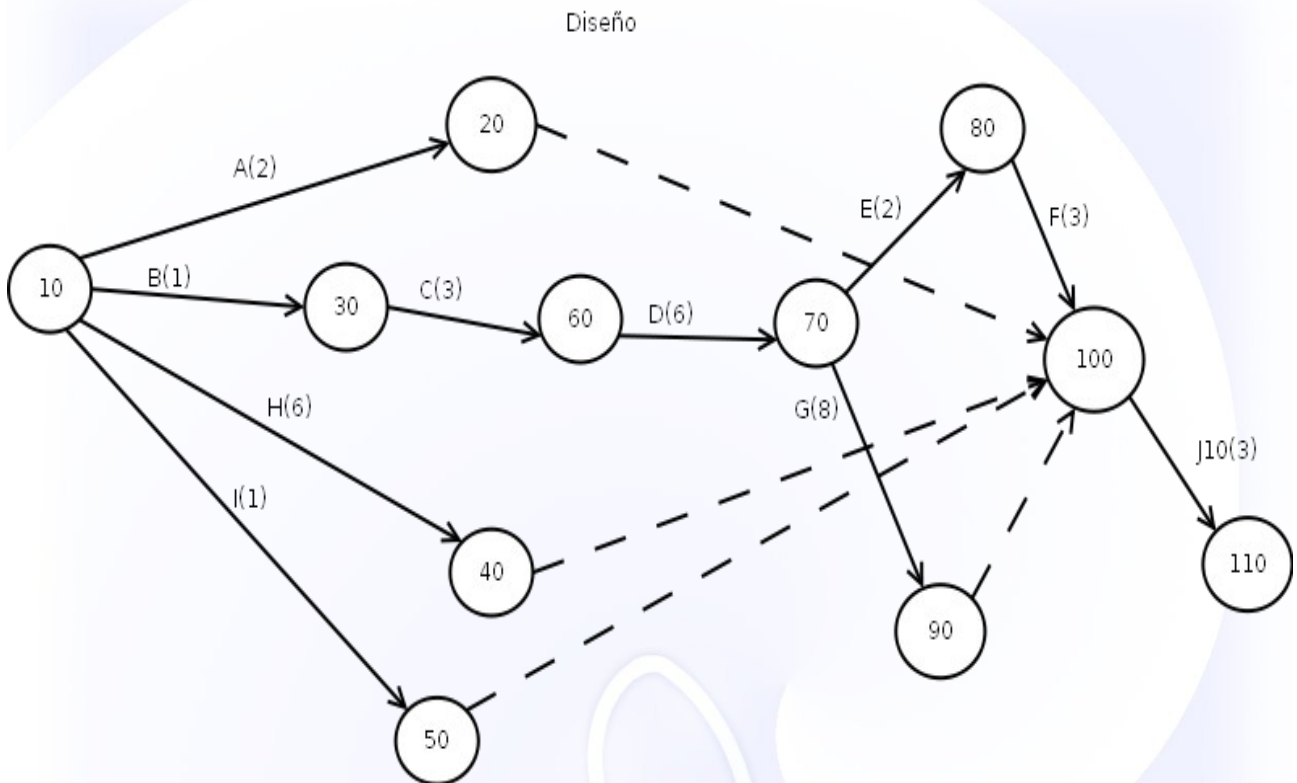
Información de la etapa

- Duración de la etapa: 3 días
- Camino crítico formado por las actividades: A, C, D, E

Holguras en actividades

- La actividad B tiene una holgura de 7.

Diseño



Descripción de las tareas

| Tarea | Tiempo | Descripción |
|-------|--------|---|
| A | 2 | Descripción de la arquitectura software. |
| B | 1 | Descomposición en subsistemas. |
| C | 3 | Elaboración del diagrama de clases de diseño. |
| D | 6 | Elaboración del diagrama de paquetes estructurales. |
| E | 2 | Elaboración del diagrama de componentes |
| F | 3 | Elaboración del diagrama de despliegue. |
| G | 8 | Elaboración de los diagramas de colaboración del diseño |
| H | 6 | Diseño de la interfaz de usuario. |
| I | 1 | Diseño de la base de datos. |
| J | 10 | Redacción del documento de diseño. |

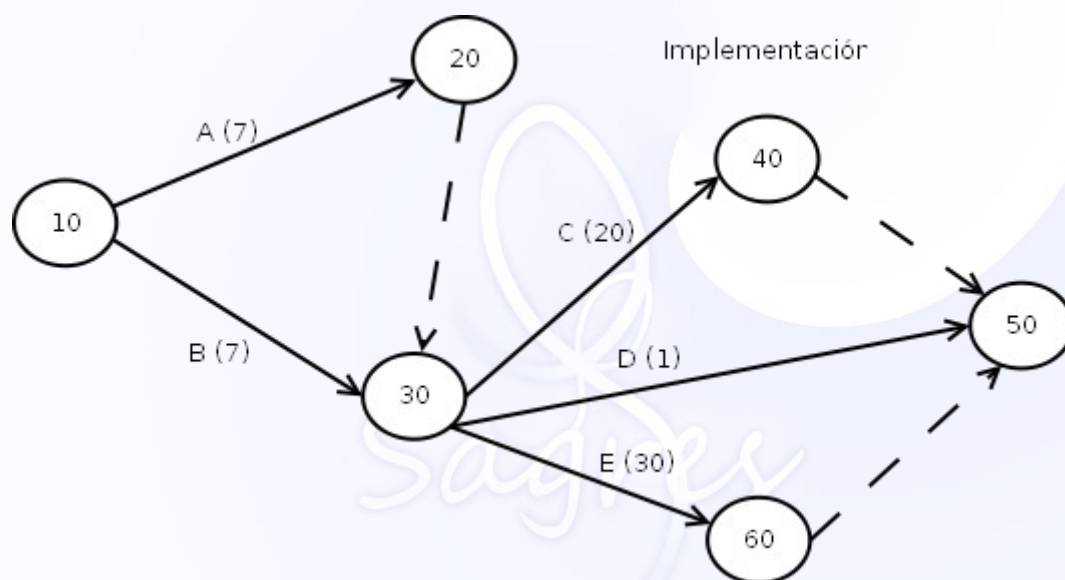
Información de la etapa

- Duración de la etapa: 7 días
- Camino Crítico: B, C, D, G, J

Holguras en actividades

- La actividad A tiene una holgura de 16 unidades de tiempo.
- La actividad H tiene una holgura de 12 unidades de tiempo.
- La actividad I tiene una holgura de 17 unidades de tiempo.
- La actividad E y F tienen una holgura de 3 unidades de tiempo.

Implementación



Descripción de las tareas

| Tarea | Tiempo | Descripción |
|-------|--------|---|
| A | 7 | Aprendizaje Java |
| B | 7 | Aprendizaje HTML, Javascript, CSS, MySql, PHP |
| C | 20 | Implementación de las interfaces (web y escritorio) |
| D | 1 | Implementación de la base de datos. |
| E | 30 | Implementación completa de los subsistemas. |

Información de la etapa

- Duración de la etapa: 18 días
- Camino Crítico: A, E o B, E

Holguras en actividades

- La actividad C tiene una holgura de 10 unidades de tiempo.
- La actividad D tiene una holgura de 29 unidades de tiempo.



6. ANÁLISIS DE LOS RIESGOS

Un posible riesgo es que un miembro del grupo falle. Entiéndase 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, colaborar de la forma necesitada.

Un riesgo que se ha presentado durante la dos fases de desarrollo es la caída del servidor SVN, en el que se encuentran alojados todos nuestros datos. Como solución a esto subimos continuamente versiones de los datos que manejamos a AWACATE, que es un servidor menos propenso a fallos según nuestra experiencia y también manejamos copias locales que en un posible fallo de AWACATE podríamos mandarnos por correo electrónico.

Un posible riesgo es la lesión o imposibilidad de un miembro de algún equipo de realizar su trabajo. En este supuesto la solución sería agregar un miembro que tenga menos carga de trabajo a ese equipo o en el caso de que no sea posible, repartir la carga asignada al miembro entre los restantes miembros del equipo, esto, como ya he dicho, como última opción.

Retrasos

En este apartado iremos comentando los retrasos que vamos sufriendo.

Modelado de Requisitos

| Nombre | Tarea | Diagrama de Gantt | Fecha | Motivo | Medida tomada |
|------------|-----------------|-------------------|----------|---|--|
| Jose David | 13,14 | V1.1 | 15/05/10 | No ha podido realizar nada porque se encuentra en las jornadas interinformativas | Se ha repartido su trabajo entre el resto de los compañeros |
| Angel | 13,15 | V1.1 | 15/05/10 | No ha podido realizar nada por cuestiones familiares | Se ha repartido su trabajo entre el resto de los compañeros |
| Carlos | 13,15 | V1.1 | 15/05/10 | Ha pasado todo el día fuera | El único integrante disponible del grupo es Daniel. Él se encargará de las tareas 13,14 y 15 |
| Daniel | 13,14 y 15 | V1.1 | 15/05/10 | Ha estado muy liado con las practicas de empresa y con prácticas de otras asignaturas y no ha podido hacer nada | El día 15 de Mayo el equipo de Análisis no ha podido hacer nada por lo que todo el trabajo de este día se pasa al día 16 |
| Jose David | 13, 14, 20 y 21 | V1.1 | 16/05/10 | No ha podido realizar nada porque se encuentra en las jornadas interinformativas | No hay ningún otro miembro del grupo disponible luego se pasan todas las tareas al 17 de Mayo. |
| Ángel | 13, 15, 16 y 17 | V1.1 | 16/05/10 | No ha podido realizar nada por cuestiones familiares | No hay ningún otro miembro del grupo disponible luego se pasan todas las tareas al 17 de Mayo. |
| Carlos | 13,15, 20 y 21 | V1.1 | 16/05/10 | Ha pasado todo el día fuera | No hay ningún otro miembro del grupo disponible luego se pasan todas las tareas al 17 de Mayo. |
| Daniel | 13,14,18 y 19 | V1.1 | 16/05/10 | Ha pasado todo el día en la graduación de su novia | No hay ningún otro miembro del grupo disponible luego se pasan todas las tareas al 17 de Mayo. |
| Todos | 28 | V1.2 | 20/05/10 | No está justificado | Se ha modificado la planificación de la etapa de Diseño. Se comienza el día 23 de Mayo |

7. FORMATO DE LA DOCUMENTACIÓN

1. Estructura de carpetas

Puesto que utilizamos para compartir carpetas, tanto el repositorio SVN como el sistema de documentos del Awacate, llevamos a cabo una estricta estructura de carpetas. Esta queda como sigue:

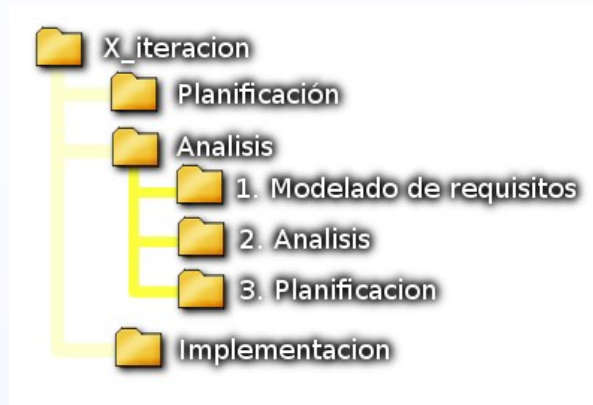


Figura 3. Estructura de carpetas.

Con X el número de la iteración. Esta estructuración permite clonar los documentos en su interior y presentar una plena homogeneidad en el proyecto, ya que un documento dentro de “1_iteración → Planificación” se da por hecho que es un documento única y exclusivamente perteneciente a la planificación de la primera iteración. Además da cierto grado de libertad a cada subgrupo puesto que cada grupo tiene únicamente control sobre su carpeta, haciendo que los cambios en una no estorben al resto del equipo.

2. Documentación

Toda la documentación es realizada en OpenOffice3, luego definiremos cómo se estructura con dicho programa de edición de textos. Para asegurar una documentación que siga acorde con los puntos que detallaremos a continuación, se han creado plantillas de documentación (formato .ott), con lo que se facilita el cumplir con ellos. Todos los miembros del grupo cuentan con una copia online o local de ellas.

Significado de las plantillas

- **Plantilla de Documento:** para crear un documento sobre una etapa, es la documentación que se entregará en cada hito. Tiene anexo un fichero de cabeceras para el título disponible en cada documento, acorde con el diseño seguido en dicho formato. El diseño ha sido escogido con tonos azules suaves. La razón de coger azules es porque es un color que da tranquilidad y sensación de bienestar, para asegurar que la lectura resulte lo más placentera posible. En detalle, punto por punto y en orden, se define:
 - *Título al completo:* prediseñado, con logo incluido, será necesario cambiar la versión marcada por la actual, al igual que la iteración.
 - *Índice de contenido:* es autogenerado, sólo **es necesario** actualizarlo tras cada cambio. El formato de cada nivel es el mismo que el correspondiente con las cabeceras. Detallaremos éstas a continuación.
 - *Apartado de control de versiones:* presenta una introducción de explicación sobre dicho apartado, además de una tabla con un diseño acorde al del documento, la tabla tiene por formato una cabecera (la primera fila) Times New Roman, tamaño 12 en negrita, centrado. El resto de filas serán en Times New Roman, tamaño 12 sin negrita. Las primeras dos columnas están alineadas a la derecha, mientras que la tercera a la izquierda. Esto resulta en una mayor comodidad a la hora de leer, ya que la versión y la fecha se encuentran más cercanas a su descripción, ayudando a los ojos del lector a tener que desplazarse menos sobre el texto para encontrar su concordancia. La letra elegida es por su tamaño estándar, puesto que en el mundo empresarial, la gente está acostumbrada a leer en Times New Roman, es un formato fácilmente reconocible que no requiere a penas esfuerzo de interpretación.
 - *Cabeceras:* tienen todas en común que utilizan el estilo de letra Segoe Print, esta tipografía es libre y se encuentra disponible en nuestro repositorio y en el apartado de documentos del Awacate, para asegurar que todos los miembros puedan disponer de ella. La elección de esta tipografía es para romper con la homogeneidad, lo que destaca más a los ojos del lector, diferenciando fácilmente que este texto es de mayor importancia.
 - Cabecera de nivel 1: La letra es grande 16.1, con un color azul-grisáceo oscuro en consonancia con el tema del documento, la letra es de capitalización automática, en negrita y con subrayado completo. Esto simboliza un corte sobre el tema del documento, significa que a partir de ahora cambiamos completamente del tema de lectura. Ayuda a diferenciar los apartados estructurales del documento.
 - Cabecera de nivel 2: La letra es de tamaño 14, en negrita. El color es azul-grisáceo, con sombreado (para ayudar a reconocer los contornos y no confundir el texto con el fondo). Una cabecera de segundo nivel indica que se va a tratar un punto referente al tema tratado en la cabecera de primer nivel.

- Cabecera de nivel 3: Letra de tamaño 12, sin negrita. El color es azul-grisáceo oscuro. Una cabecera de tercer nivel sencillamente indica que se entra en detalle sobre un apartado tratado en la cabecera de segundo nivel.
- *Apartado de apéndices:* Al final de cada documento, debe haber un apartado de apéndices. Estos tienen el nombre de la versión a la que hacen referencia (Por ejemplo, el “Apéndice 0.1” hace referencia a los cambios realizados en la versión v0.1).
- **Plantilla de Especificación de casos de uso:** es una tabla que especifica todos los detalles a enumerar, el formato de las enumeraciones en los cursos de los eventos, tanto normales como alternativos. Se ha utilizado un bordeado azul-grisáceo suave dispuesto de tal forma que de la sensación de profundidad. Los colores son acorde al tema utilizado en el documento. La cabecera de la tabla es la primera columna, la letra elegida es Segoe Prints, tamaño 11, en negrita. El cuerpo de la tabla tiene letra Times New Roman tamaño 12, por las razones anteriormente detalladas. Esta diferenciación permite al lector reconocer rápidamente cual es la cabecera y cual el contenido.
- **Plantilla de Contratos:** es una tabla idéntica a la anterior, con la cabecera de tabla necesaria para detallar un contrato correcto.

Reglas de los documentos

- Las cabeceras no pueden terminar en dos puntos, o en punto (Por ejemplo, “Estructuras:”), porque rompe la visualización correcta en el índice y, por tanto, en los accesos del documento.
- **No se permite** insertar ninguna indentación (o sangría) que no sea implícita del texto, tal como las enumeraciones. Esto hará que se aproveche en todo momento el ancho del documento al completo, evitando tener que cambiar de línea de lectura demasiado a menudo.
- Los documentos siguen todos el nombre de lo que son. Aquí encontramos dos apartados:
 - El documento está en formato .odt (u otro modificable).
 - Estos documentos no necesitan terminar con el nombre de la versión, ya que las herramientas de control de versiones (SVN) llevan control de ellas.
 - Estas no son versiones finales, debido a su carácter modificable, por lo que no se deberán consultar como tales.
 - El documento está en formato .pdf, significa que es una versión acabada (Salvo en el caso de que sea precedido por el nombre de su creador).
 - Estos documentos **deben** terminar con el nombre de su versión. Se le añade al nombre “v[version]”.
 - En cuanto a la versión, el número delante del punto indicará la iteración en que ha sido realizado, y el que hay a continuación la versión dentro de esa etapa (véase ejemplo abajo). Esto se hará en todos los documentos excepto el de planificación que se entiende que no es necesario por ser una guía para el equipo en la realización del proyecto.

Documento de Modelado de Requisitos v3.2.pdf.

- En el ejemplo anterior, tendremos que es la segunda versión de la tercera iteración.
- A partir de esta tercera iteración, los documentos incluirán al sistema completo, es decir, los documentos que empiezan por v3.x corresponden a documentos finales englobando toda la información del sistema en esa fase del desarrollo del software.
- **NUNCA** se borran, son acumulativos.
- Se permite que los documentos comiencen con el nombre de su creador entre corchetes. Estos aunque estén en formato .pdf (Con lo que se siguen las reglas anteriores) no se considerarán como versión final. Esto es para ayudar a la organización de los documentos.

[Sergio] Documento de Planificación v0.2.pdf.

Consejos prácticos de uso

- Modificar directamente los archivos editables (por ejemplo, para la documentación, los .odt) desde el repositorio, o bien guardarlos ahí tras cada modificación, aunque no se saque una versión final, esto ayuda a que todos los miembros accedan al documento actualizado y puedan añadir sus modificaciones de forma actualizada.
- Actualizar el repositorio cada vez que se va a explorar, por la misma razón que el punto anterior.
- La portada debe mostrar la última versión en la que se encuentra el documento, es decir, debe coincidir con la última fila del “Historial de versiones” y con el número del último apéndice.
- Comprobar que la fecha tanto del historial de versiones como de los apéndices concuerda.
- Englobar los temas relacionados con encabezados, por ejemplo, si vamos a tener varios encabezados superiores que traten los mismos temas, será más correcto englobarlos en una única cabecera y utilizar cabeceras de menor grado. Por ejemplo:

[Encabezado 1] Revisión del documento de modelado de requisitos

[Encabezado 1] Revisión del documento análisis

[Encabezado 1] Revisión del documento diseño

[Encabezado 1] Revisiones

[Encabezado 2] Documento de modelado de requisitos

[Encabezado 2] Documento análisis

[Encabezado 2] Documento diseño

- Regenerar el índice antes de imprimir el documento en formato PDF.

3. Estructura en los documentos de código

Reglas semánticas

Las siguientes reglas se aplican para el código de Java, de PHP y SQL.

Todos los nombres de variables, funciones, comentarios, clases, etc. están en *español. Los nombres seguirán la estructura “*Camel Case*” donde todas las letras son minúsculas excepto la primera letra de cada palabra, la primera letra de la primera palabra es también una minúscula. Tienen que tener un nombre de valor informativo, acorde con su función o valor. Por ejemplo:

```
obtenerEdad(), edad, edadPadre, etc.
```

Por razones de compatibilidad los nombres de variables y métodos no podrán tener caracteres especiales españoles, como 'ñ' o acentos.

**Hay dos tipos de funciones que, por facilidad de escritura no estarán en español, estas son las funciones de asignación y de devolución de valor.*

Denominación de variables

Vamos a utilizar generalmente sustantivos para nombrar a las variables, ya que suelen significar instancias. El nombre de las variables globales de las clases empezarán con “g” (por “variable global”), por compatibilidad con la internacionalización de código (global es igual en inglés, idioma extensamente utilizado en el código). Por ejemplo:

```
gEdad, gEdadPadre
```

Las variables utilizadas para las iteraciones en los bucles no tienen porqué necesariamente tener un nombre con significado español, por su corta vida y su continuada referencia en bucles, lo más común será utilizar una única letra (tal como 'i', 'j', 'k', etc).

Las constantes no seguirán la regla de “*Camel Case*” ya que estarán totalmente en mayúsculas. Para solventar el problema de tener nombres compuestos por distintas palabras, la separación la haremos mediante un guión bajo:

```
EDAD, EDAD_PADRE
```


Denominación de funciones

Para la denominación de funciones tendremos en cuenta cuatro partes, la denominación de funciones genéricas, funciones booleanas, funciones de obtención de valor y funciones de asignación de valor:

1. Funciones genéricas: Los nombres de funciones están , si es aplicable, en forma verbal imperativa, aunque podrán contener un sustantivo. Por ejemplo:

```
pinta(...), conecta(...), pintaVentana(...)
```

2. Funciones de retorno booleano (o condicionales): estas funciones se utilizan para devolver estados (verdadero o falso). El nombre de este tipo de funciones será “está” seguido de un adjetivo o un participio.

```
estaLleno(...), estaConectado(...)
```

3. Funciones de asignación de valor: asignan un valor a algún dato (generalmente perteneciente a una clase), como excepción anteriormente mencionada, estas comenzarán por “set” seguido del nombre (de valor informativo) de variable. Por ejemplo, si vamos a asignar la altura a una ventana:

```
ventana.setAltura(x);
```

4. Funciones de obtención de valor: estas funciones devuelven algún valor, generalmente de alguna instancia de una clase. Estas funciones **nunca tendrán parámetros**. Al igual que la anterior, esta comienza en inglés con la palabra “get” (obtener) seguido del atributo que se espera obtener. Por ejemplo, obtener la altura de una ventana:

```
ventana.getAltura();
```

Denominación de clases

La definición de las clases serán sustantivos en singular, con la primera letra en mayúscula, para distinguirlos de la denominación de variables. Algunos ejemplos:

```
Ventana, Fecha, Persona
```


Documentación y comentarios

Cualquier comentario dentro de línea será aceptado siempre y cuando explique la funcionalidad con claridad, es decir, no se aceptarán comentarios del estilo “Aquí”, sino comentarios que aclaren la función, como por ejemplo:

```
Integer numero=0;
while(numero<5){
    imprimir objeto.getNombre(numero++);    // Aumentamos el valor de numero tras obtener
                                           // el nombre del numero deseado
}
```

Para documentar el código, se utiliza la estructura de JavaDoc, esta estructura queda como sigue:

Antes de cada función:

```
/**
 * Una descripción muy corta de lo que hace la función.
 * @param <parámetro> - descripción del parámetro <parámetro> (uno por cada parámetro)
 * @return            - si la función no es de tipo “void” se pondrá una descripción del valor de retorno
 * @throw             - si esta función puede lanzar una excepción se pondrá una descripción ella
 */
```

Antes de cada clase:

```
/**
 * Una descripción muy corta sobre el cometido de la clase
 * @author           - el nombre de persona que ha implementado esta clase
 */
```

Llaves e indentación

La llave de la izquierda '{' estará en la línea de declaración de clase/función/bucle etc. Después de cada '{' las siguientes líneas están indentadas con una tabulación. Puesto que todos los miembros están instados a utilizar el IDE NetBeans6.8, todos tendremos un código igual indentado, con formateado automático por parte del IDE. Después de cada '}' la indentación tendrá una tabulación menos.

Después de “{” o antes de “}” no habrá ninguna línea en blanco, tan sólo código o un comentario explicativo. Por ejemplo:

```
while(...) {
    if (...) {
        // código
        // código
        // código
    }
}
```

APÉNDICE 1.0

| | |
|---------------------------------|---|
| Fecha | 14/05/10 |
| Descripción del problema | - |
| Impacto del problema | - |
| Soluciones adoptadas | <ul style="list-style-type: none">• Se ha generado el documento de planificación inicial basado en el documento de planificación v1.5 de la segunda iteración.• Se rehace la declaración de alcance con respecto a la nueva iteración• Se especifica una nueva planificación temporal.• Se agregan nuevos riesgos a tener en cuenta en el análisis de riesgos. |
| Anexos a la versión | <ul style="list-style-type: none">• Agregado anexo “Diagrama de Gantt.png” y fichero de OpenProject “Planificación 3ª Iteración v1.0.pod” |

Sagres

APÉNDICE 1.1

| | |
|---------------------------------|--|
| Fecha | 17/05/10 |
| Descripción del problema | <ul style="list-style-type: none">- Se han producido algunos retrasos.- Información de dichos retrasos.- Cambio de hitos y reestructuración de tareas. |
| Impacto del problema | <ul style="list-style-type: none">- El equipo de análisis y diseño ha sufrido algunos retrasos debido a diversos inconvenientes que han surgido a sus miembros.- El equipo de planificación ha sido informado de todos estos problemas. |
| Soluciones adoptadas | <ul style="list-style-type: none">• Se ha creado un nuevo diagrama de Gantt con una reestructuración de tareas.• Se ha creado un informe de retrasos que irá siendo actualizado a lo largo de la iteración.• Se han cambiado las fechas de entrega de los Documentos de modelado, análisis y diseño. |
| Anexos a la versión | <ul style="list-style-type: none">• Agregado anexo “Diagrama de Gantt v1.2.png” y fichero de OpenProject “Planificacion 3ª Iteracion v1.2.pod” |

Sagres

APÉNDICE 1.2

| | |
|---------------------------------|--|
| Fecha | 23/05/10 |
| Descripción del problema | <ul style="list-style-type: none">- Añadidos diagramas de Pert.- La numeración de los documentos de la fase de análisis no es representativa.- Retrasos en la fase de análisis. |
| Impacto del problema | <ul style="list-style-type: none">- Planificación de la entrega de los diagramas de Pert.- Los documentos de análisis de esta tercera iteración, incluyen toda la información de la primera y la segunda. La numeración que usamos hasta ahora v1.x ... no refleja que contiene todo el desarrollo del sistema en sus tres iteraciones.- Se ha producido un retraso en la etapa de análisis que ha evitado entregar su documento en su fecha correcta. |
| Soluciones adoptadas | <ul style="list-style-type: none">• Se han añadido al documento los diagramas de Pert, con sus tablas de tareas correspondientes y sus horas.• Se han modificado las numeraciones de los documentos de análisis a Vx.y donde x corresponde a la iteración e y a la versión del documento. Se ha especificado en su correspondiente apartado.• Se han reestructurado las tareas, ampliando la etapa de análisis y recortando la de diseño. Esto se refleja en el nuevo diagrama de Gantt y la actualización de la tabla de retrasos |
| Anexos a la versión | <ul style="list-style-type: none">• Agregados en su correspondiente carpeta las imagenes .png y los ficheros .día de cada diagrama de Pert.• Agregado Diagrama de Gantt v1.3.png |