



Planificacin

Fco. Javier Bohrquez Ogalla

ndice

1. Visin general	3
2. Metodologa de desarrollo	4
3. Planificacin	5
3.1. General	5
3.2. Expresiones lgicas	6
3.3. Sentencias de entrada/salida	7
3.4. Sistema de errores	8
3.5. Expresiones aritmticas	9
3.6. Smbolos variables	10
3.7. Sentencias de control I	11
3.8. Interfaz de usuario I	12
3.9. Expresiones cadenas de caracteres	13
3.10. Conversin de tipos	14
3.11. Funciones de cadenas	15
3.12. Expresiones array	16
3.13. Smbolos funciones	17
3.14. Expresiones regulares	18
3.15. Smbolos clases I	19
3.16. Expresiones condicionales	20
3.17. Funciones de depuracin	21

3.18. Optimizacin de memoria	22
3.19. Sentencias de control II	23
3.20. Paso de argumentos	24
3.21. Interfaz de usuario II	25
3.22. Procesos I	26
3.23. Fechas y tiempo	27
3.24. Ficheros	28
3.25. Extensiones	29
3.26. Extensin gettext	30
3.27. Procesos II	31
3.28. Smbolos clases II	32
3.29. Smbolos funciones II	33
3.30. Extensin mysql	34
4. Organizacin	34
5. Costes	35
6. Riesgos	36
6.1. Riesgos tecnolgicos	37
6.2. Riesgos personales	39
6.3. Riesgos organizativos	39
6.4. Riesgos de requisitos	40
6.5. Riesgos de soluciones	41
6.6. Riesgos de costes, tiempos y recursos	43

1. Visin general

En esta seccin se trata todos los aspectos relativos a la gestin del proyecto. Esta pretende dar alcance a la meta del proyecto y los objetivos del mismo dentro de las limitaciones dadas: alcance, tiempo, calidad y presupuesto.

La gestin del proyecto comprende la implantacin de una metodologa de desarrollo, vindose esta como la implantacin de una serie de pasos, tcnicas, procedimientos y dems recursos que ayuden a desarrollar el producto software dentro de un marco de trabajo.

Por otro lado tambin se ha de tratar la planificacin del proyecto, quedando este organizado en una serie de tareas y actividades derivadas de la metodologa implantada.

En la ejecucin de un proyecto es necesario la inversin de una serie de recursos, tales como herramientas, personal, equipos, herraminetas... Se ha de determinar los recursos asignados a la ejecucin del mismo, as como los roles de las personas asignadas y la relacin entre estas.

El desarrollo de un proyecto software tiene unos costes derivados de los recursos asociados al mismo. Estos recursos sern tanto materiales como humanos, y tendrn un coste fijo asociado que se utilizar para el clculo del coste total.

Otra de las tareas que se llevan a cabo durante la gestin de un proyecto software es el anlisis de riesgos. Todo proyecto esta sujeto a una probabilidad de que se den escenarios de riesgo en los que se pueda ver perjudicada la correcta realizacin del mismo. Se detectarn, listarn y analizarn los riesgos y sus consecuencias, as como la probabilidad de que estos sucedan y las prticas que se llevarn a cabo para mitigar los efectos derivados de estos.

Por ltimo se expondrn las prticas seguidas para asegurar la calidad del desarrollo y los productos obtenidos en cada paso de la metodologa. Para ello se incuyen estndares seguidos los estndares, prticas y normas aplicables durante el desarrollo. Adems se recogen los distintos tipos de revisiones, verificaciones y validaciones que se han llevado a cabo, as como los criterios para la aceptacin o rechazo de cada producto y los procedimientos para implantar acciones correctoras o preventivas.

La planificacin expuesta no recoge aspectos como la instalacin, el mantenimiento o el soporte. Esta se centra nicamente en el ciclo de desarrollo del proyecto y no en los procesos posteriores, los cuales, aunque tambin forman parte del ciclo de vida hbil del software, no se encuentran dentro de las etapas de desarrollo del mismo.

2. Metodologia de desarrollo

Para la realizacin del proyecto se ha seguido una metodologa iterativa e incremental. Ms concretamente se ha tomado como base el proceso unificado de desarrollo de software, el cual sigue un enfoque dirigido por casos de uso y centrado en la arquitectura.

El ciclo de vida sigue un enfoque en espiral, dividido en cuatro etapas: determinar objetivos, anlisis de riesgos, desarrollo y planificacin.

Determinar objetivos :

- Se fijan los productos a obtener: requisitos, especificacin, manuales ...
- Se fijan las restricciones a las que estar sujeta el proyecto
- Slo en la primera iteracin se lleva a cabo una planificacin inicial en esta etapa.

Anlisis de riesgos :

- Se estudia las posibles amenazas y eventos no deseados, as como los daos y consecuencias derivados de estos.
- Se evaluan las distintas alternativas que permitan minimizar los riesgos.

Desarrollo :

- Se lleva a cabo el desarrollo de lo fijado en las etapas anteriores.
- El desarrollo de cada iteracin se divide en cuatro etapas: anlisis, diseo, codificacin y pruebas.

Planificacin :

- Se analiza los productos obtenidos y el estado del proyecto.
- Se lleva a cabo una planificacin de la siguiente iteracin del ciclo de vida.

En un enfoque en espiral lo ms comn es que en la primera iteracin se ofrezca un prototipo del producto a desarrollar, no obstante en el proyecto abordado no ha sido as. En lugar de ello se ha planteado una primera iteracin que recoja el alcance del proyecto, as como los requisitos y anlisis de los riesgos globales, adems se realiza una planificacin de las iteraciones que seguirn. Las dems iteraciones contemplan un subconjunto de estos requisitos aadiendose as en cada iteracin caractersticas al software y afinando el anlisis global llevado en la primera y siguientes iteraciones.

Para la realizacin de los productos obtenidos en cada paso de la metodologa se a utilizado el lenguaje de modelado UML.

3. Planificacin

La planificacin se divide en una serie de iteraciones. Todas las iteraciones con excepcin de la primera tienen las mismas etapas en funcin de la metodologa seguida.

Las etapas y subetapas en las que se divide cada iteracin son:

- Objetivos
- Riesgos
- Desarrollo
 - Anlisis
 - Diseo
 - Codificacin
 - Pruebas
- Planificacin

La planificacin tiene como punto de partida el da 03/11/2014, da en el que se comenz el desarrollo del proyecto. Se ha tomado una jornada laboral de 8 horas, y una semana hbil de 5 das.

En cada etapa de la planificacin se hacen labores de documentacin para que toda la informacin relativa al proyecto quede reflejada en la memoria del mismo.

La primera etapa refleja un planteamiento general del proyecto, este se ha ido refinando con cada iteracin del ciclo de vida. Para ello se han modificado los documentos obtenidos de iteraciones anteriores.

3.1. General

Nombre	Trab...	Semana 45, 2014						
		3	4	5	6	7	8	9
General	3d	[Barra]						
Objetivos	1d							
Análisis	1d							
Planificaci3n	1d							

Figura 1: General: 03/11/2014 - 05/11/2014

3.2. Expresiones lógicas

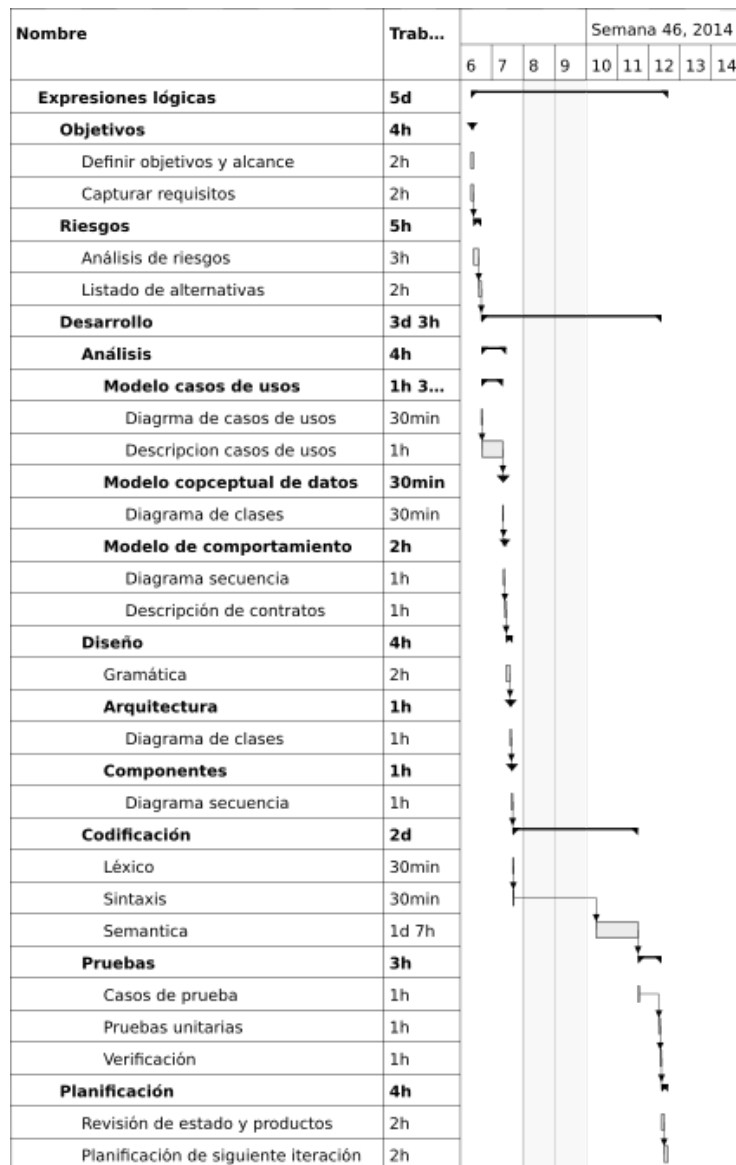


Figura 2: Exp. lógicas: 06/11/2014 - 12/11/2014

3.3. Sentencias de entrada/salida

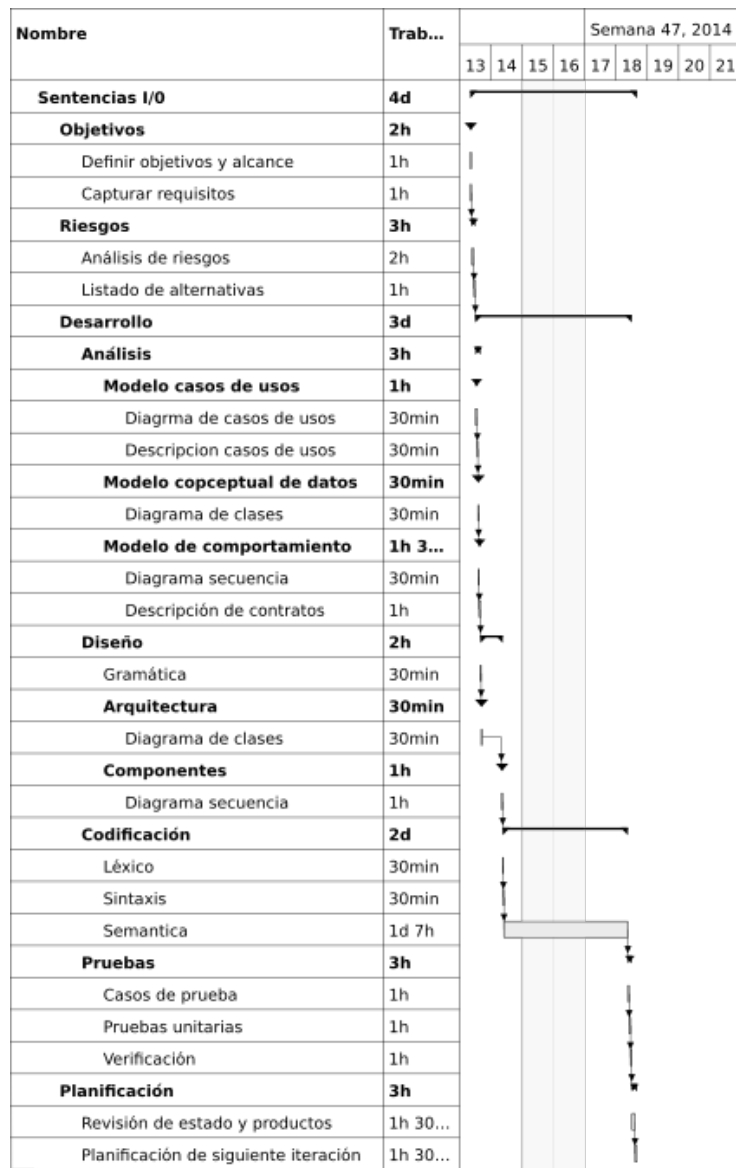


Figura 3: Sentencias I/O: 13/11/2014 - 18/11/2014

3.4. Sistema de errores

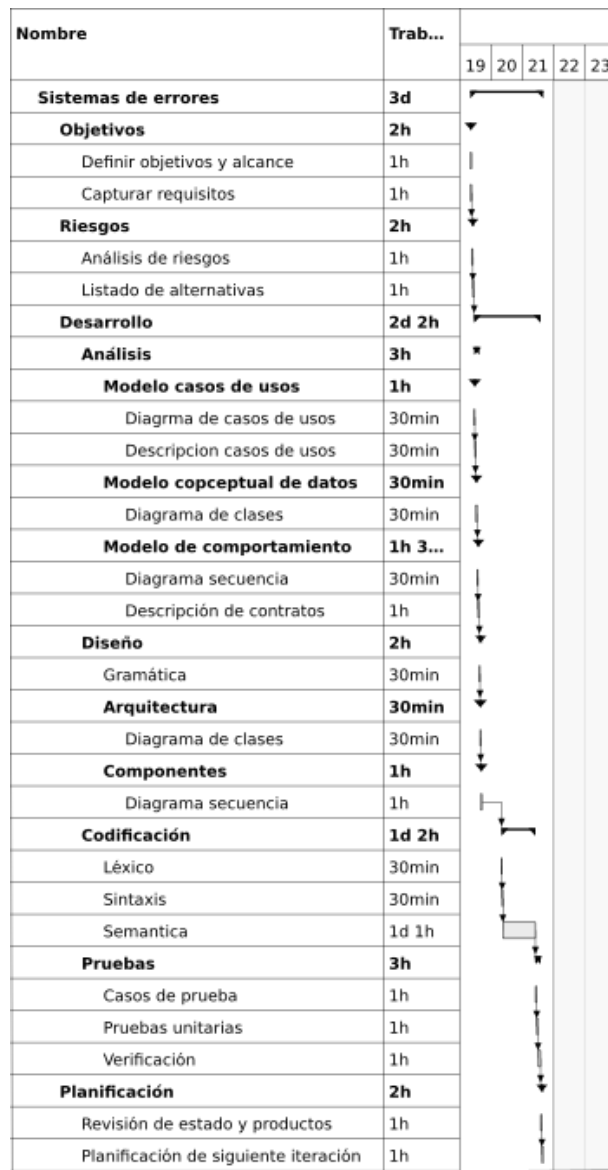


Figura 4: Sistema de errores: 19/11/2014 - 21/11/2014

3.5. Expresiones aritméticas

Nombre	Trab...	Semana 48, 2014						
		24	25	26	27	28	29	30
Expresiones aritméticas	5d							
Objetivos	4h							
Definir objetivos y alcance	2h							
Capturar requisitos	2h							
Riesgos	5h							
Análisis de riesgos	3h							
Listado de alternativas	2h							
Desarrollo	3d 3h							
Análisis	4h							
Modelo casos de usos	1h 3...							
Diagrama de casos de usos	30min							
Descripción casos de usos	1h							
Modelo conceptual de datos	30min							
Diagrama de clases	30min							
Modelo de comportamiento	2h							
Diagrama secuencia	1h							
Descripción de contratos	1h							
Diseño	4h							
Gramática	2h							
Arquitectura	1h							
Diagrama de clases	1h							
Componentes	1h							
Diagrama secuencia	1h							
Codificación	2d							
Léxico	30min							
Sintaxis	30min							
Semántica	1d 7h							
Pruebas	3h							
Casos de prueba	1h							
Pruebas unitarias	1h							
Verificación	1h							
Planificación	4h							
Revisión de estado y productos	2h							
Planificación de siguiente iteración	2h							

Figura 5: Exp. aritméticas: 24/11/2014 - 28/11/2014

3.6. Símbolos variables



Figura 6: Símbolos variables: 01/12/2014 - 14/12/2014

3.7. Sentencias de control I

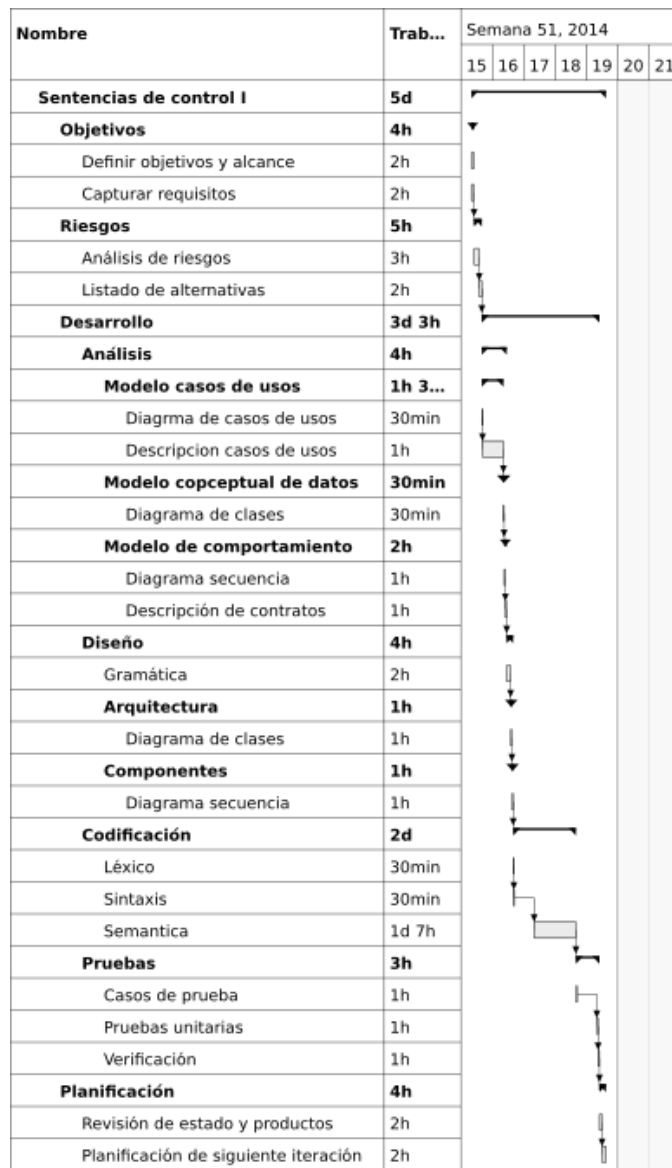


Figura 7: Sentencias control I: 15/12/2014 - 19/12/2014

3.8. Interfaz de usuario I

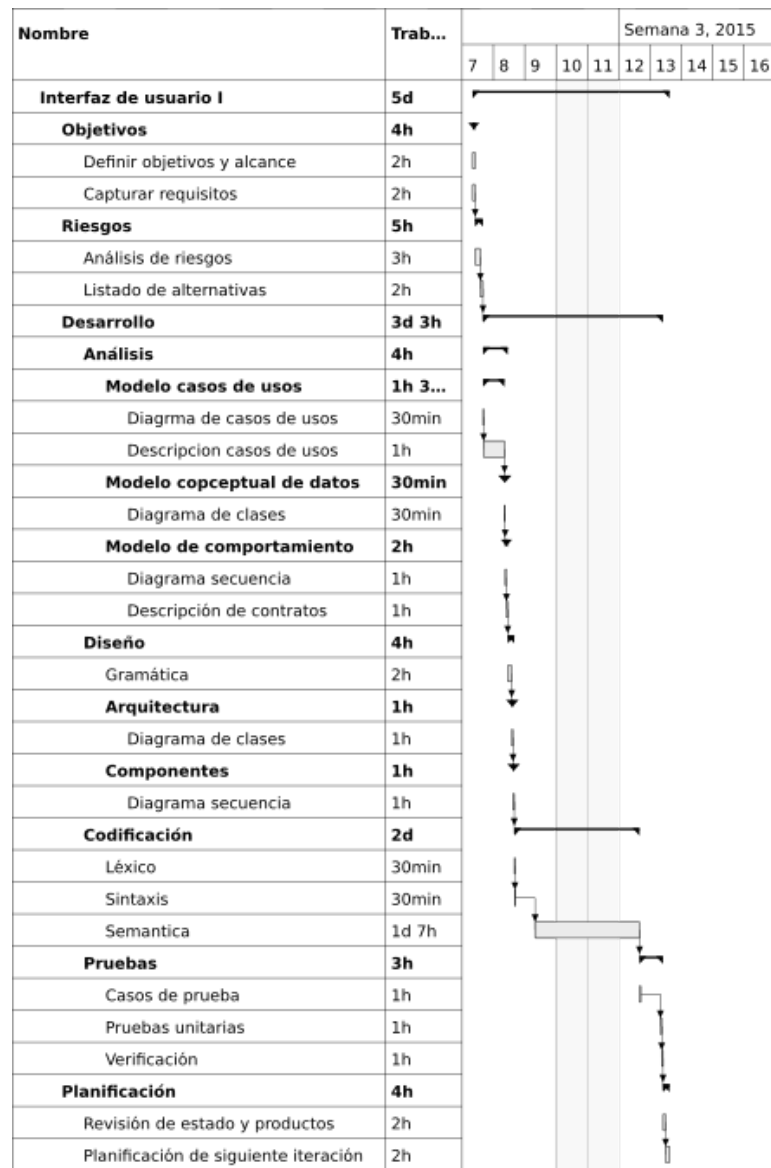


Figura 8: Interfaz usuario I: 07/01/2015 - 13/01/2015

3.9. Expresiones cadenas de caracteres

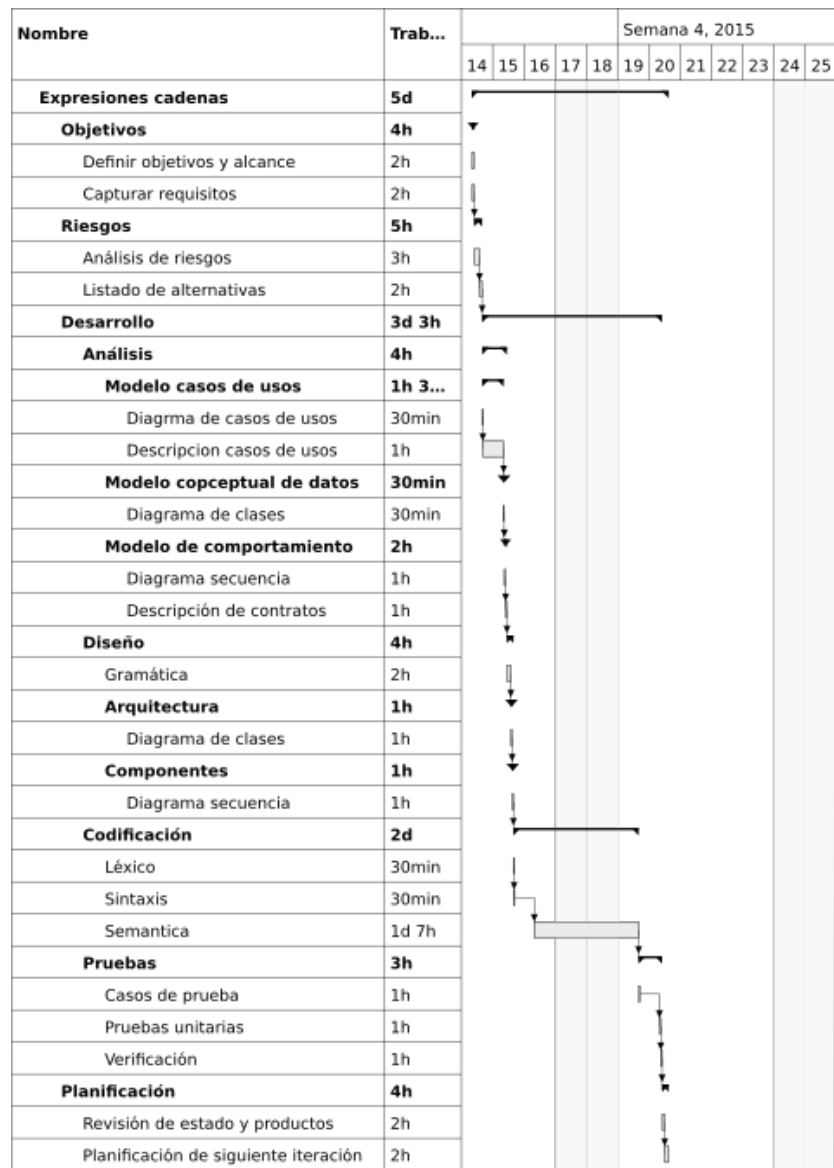


Figura 9: Interfaz usuario I: 14/01/2015 - 20/01/2015

3.10. Conversin de tipos

Nombre	Trab...					
		21	22	23	24	25
Conversión de tipos	3d					
Objetivos	2h					
Definir objetivos y alcances	1h					
Capturar requisitos	1h					
Riesgos	2h					
Análisis de riesgos	1h					
Listado de alternativas	1h					
Desarrollo	2d 2h					
Análisis	3h					
Modelo casos de usos	1h					
Diagrama de casos de usos	30min					
Descripción casos de usos	30min					
Modelo conceptual de datos	30min					
Diagrama de clases	30min					
Modelo de comportamiento	1h 3...					
Diagrama secuencia	30min					
Descripción de contratos	1h					
Diseño	2h					
Gramática	30min					
Arquitectura	30min					
Diagrama de clases	30min					
Componentes	1h					
Diagrama secuencia	1h					
Codificación	1d 2h					
Léxico	30min					
Sintaxis	30min					
Semántica	1d 1h					
Pruebas	3h					
Casos de prueba	1h					
Pruebas unitarias	1h					
Verificación	1h					
Planificación	2h					
Revisión de estado y productos	1h					
Planificación de siguiente iteración	1h					

Figura 10: Conversin de tipos: 21/01/2015 - 23/01/2015

3.11. Funciones de cadenas

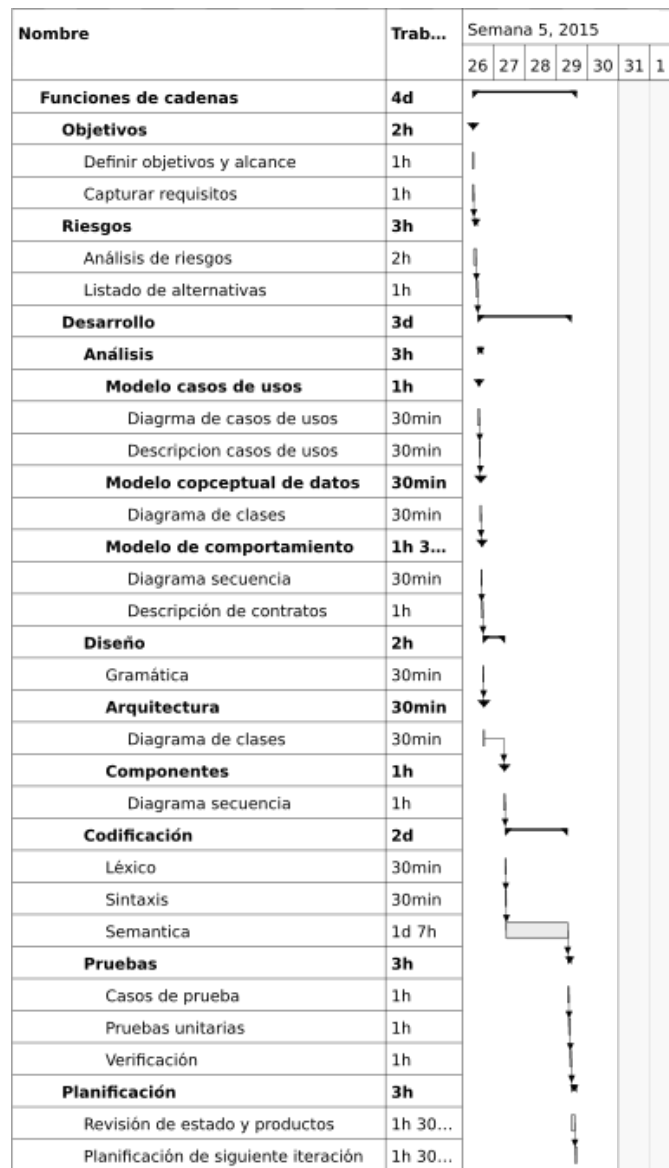


Figura 11: Funciones de cadenas: 26/01/2015 - 29/01/2015

3.12. Expresiones array

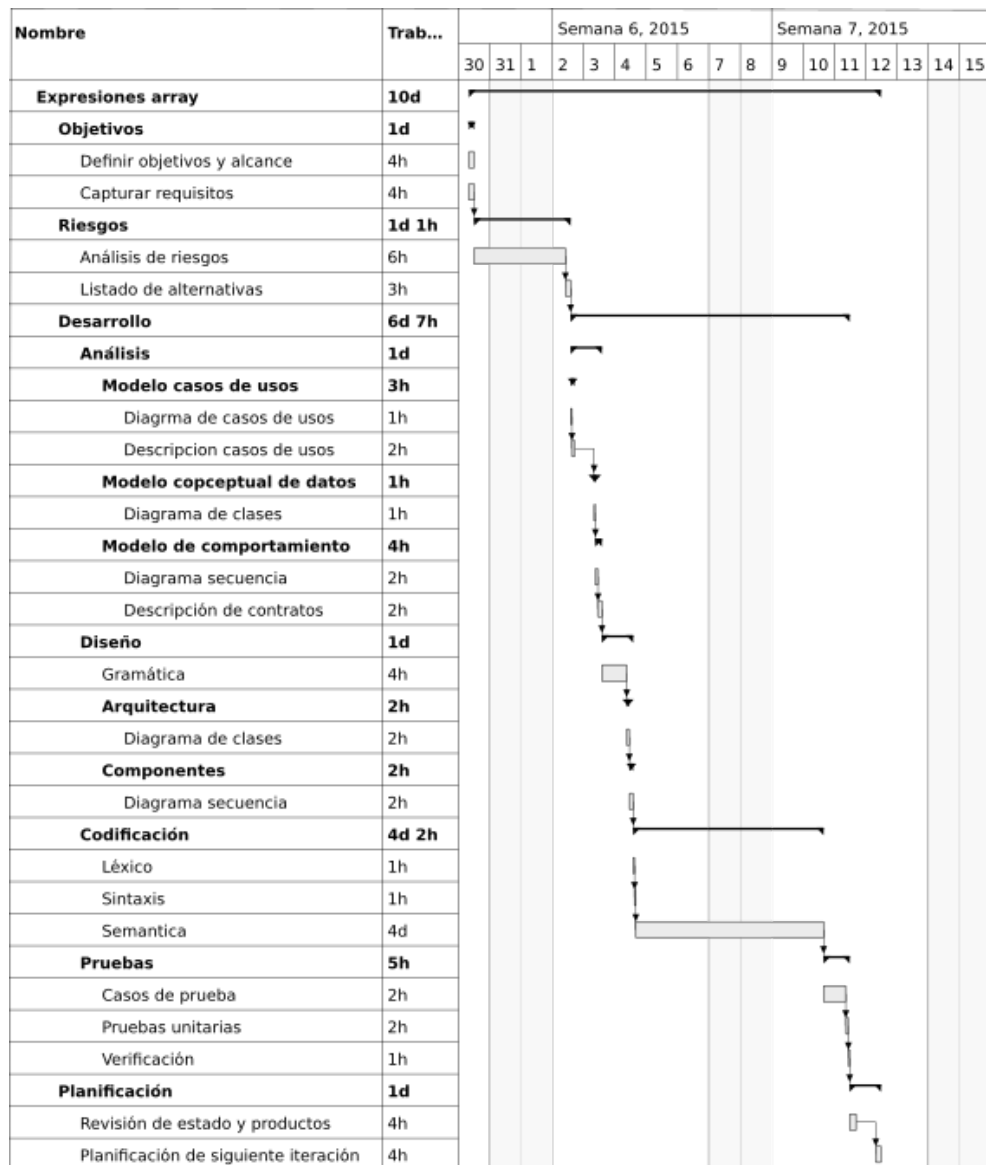


Figura 12: Expresiones array: 30/01/2015 - 12/02/2015

3.13. Smbolos funciones

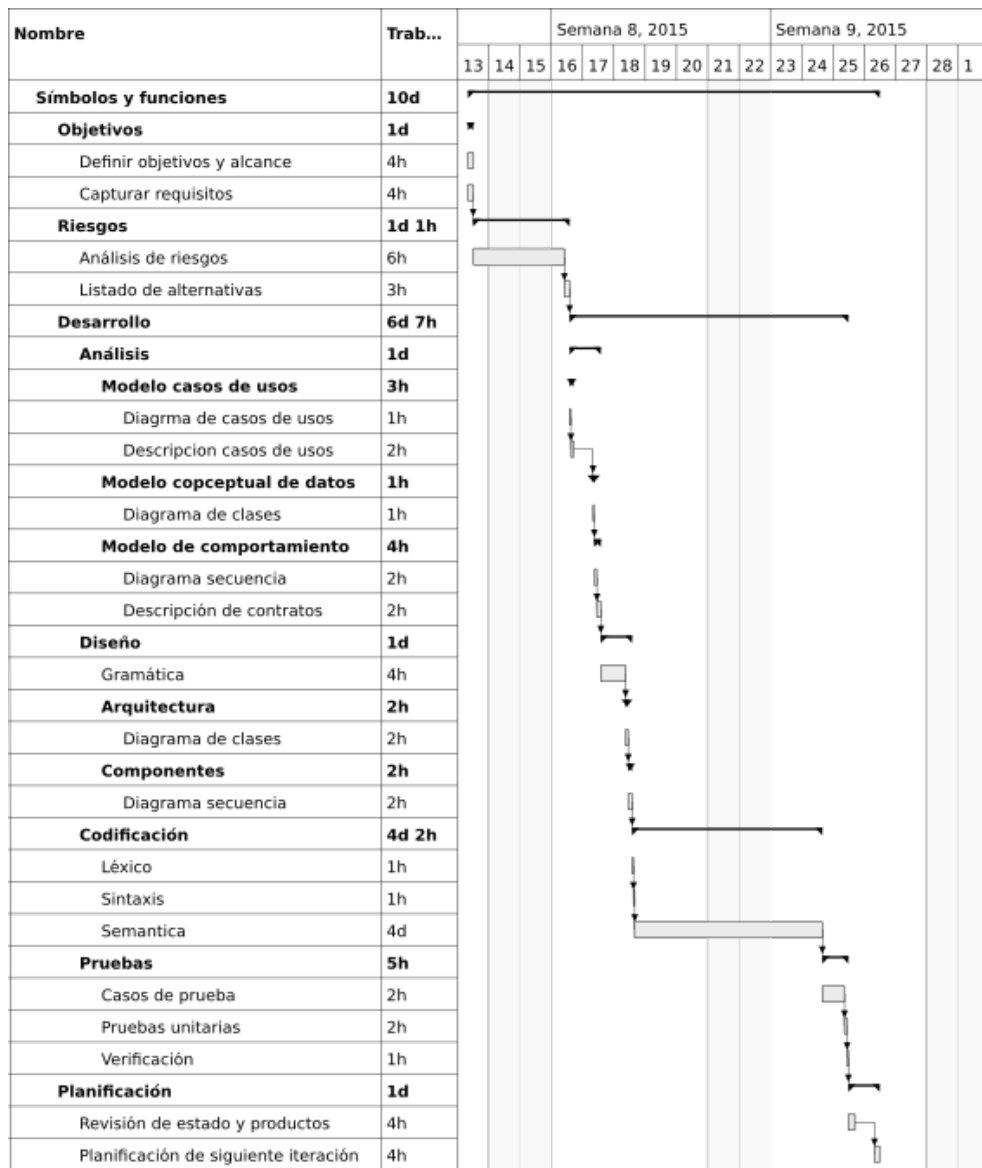


Figura 13: Smbolos funciones: 13/02/2015 - 26/02/2015

3.14. Expresiones regulares

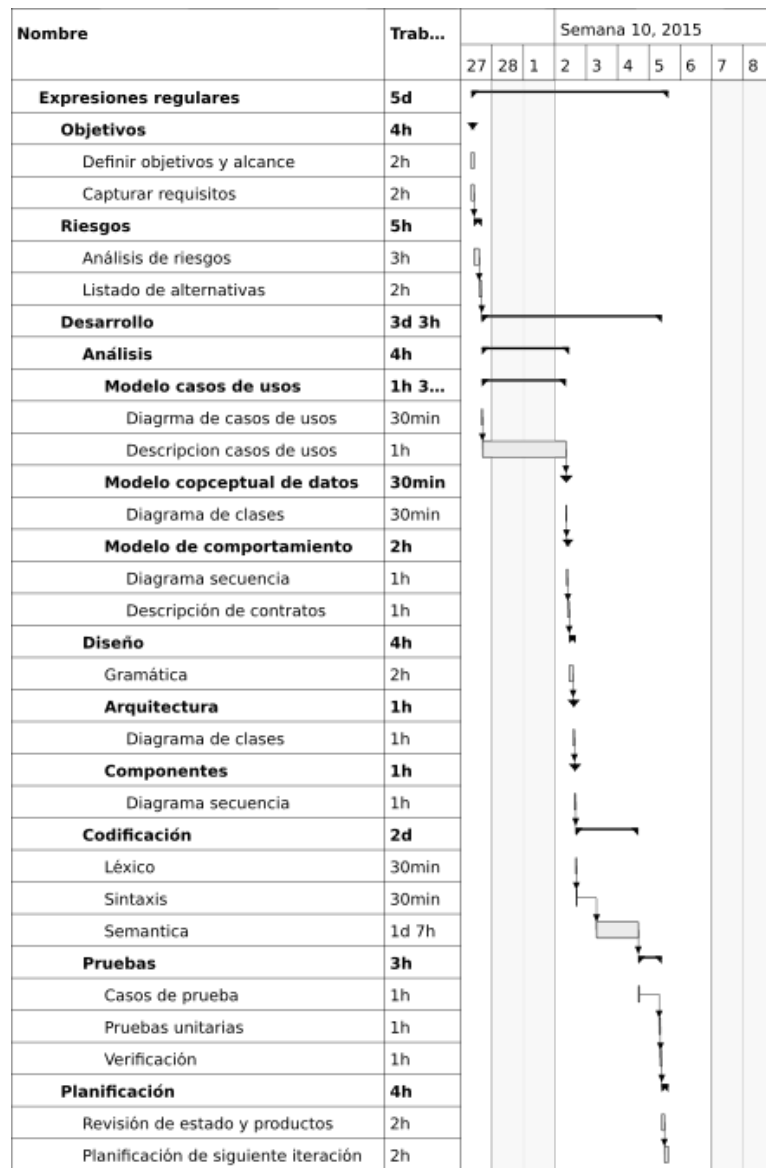


Figura 14: Expresiones regulares: 27/02/2015 - 05/03/2015

3.15. Símbolos clases I

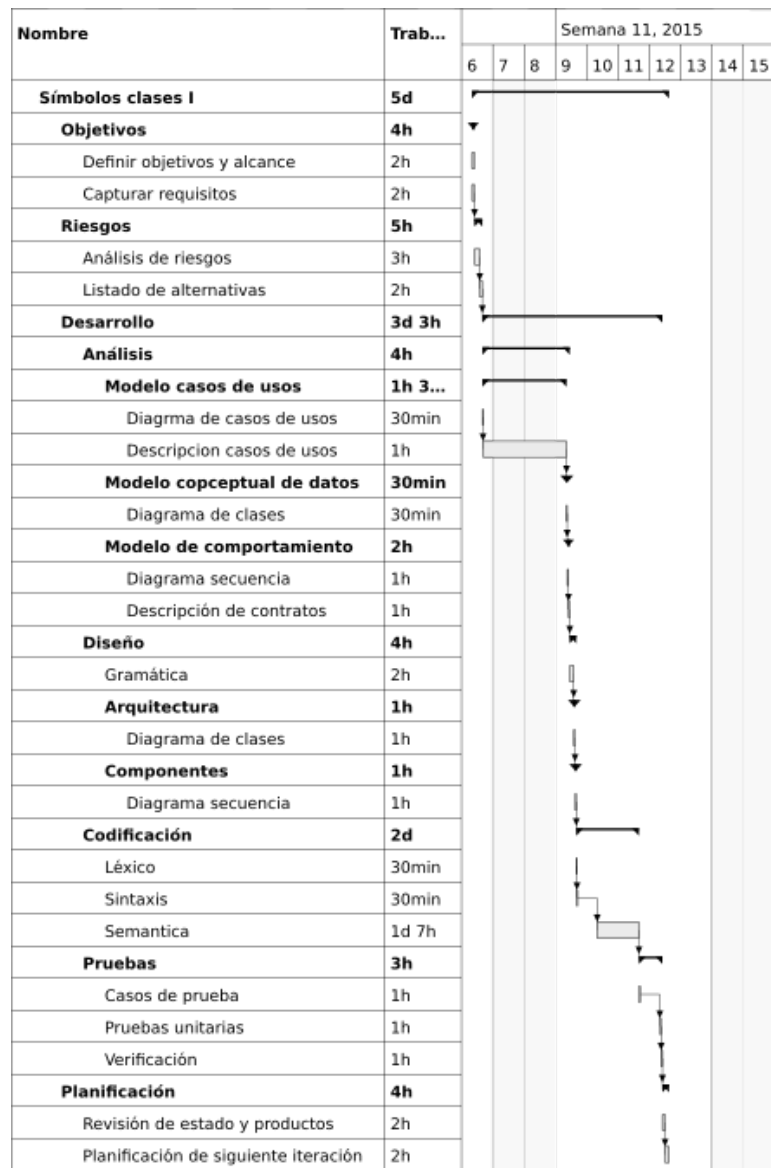


Figura 15: Símbolos clases I: 06/03/2015 - 12/03/2015

3.16. Expresiones condicionales

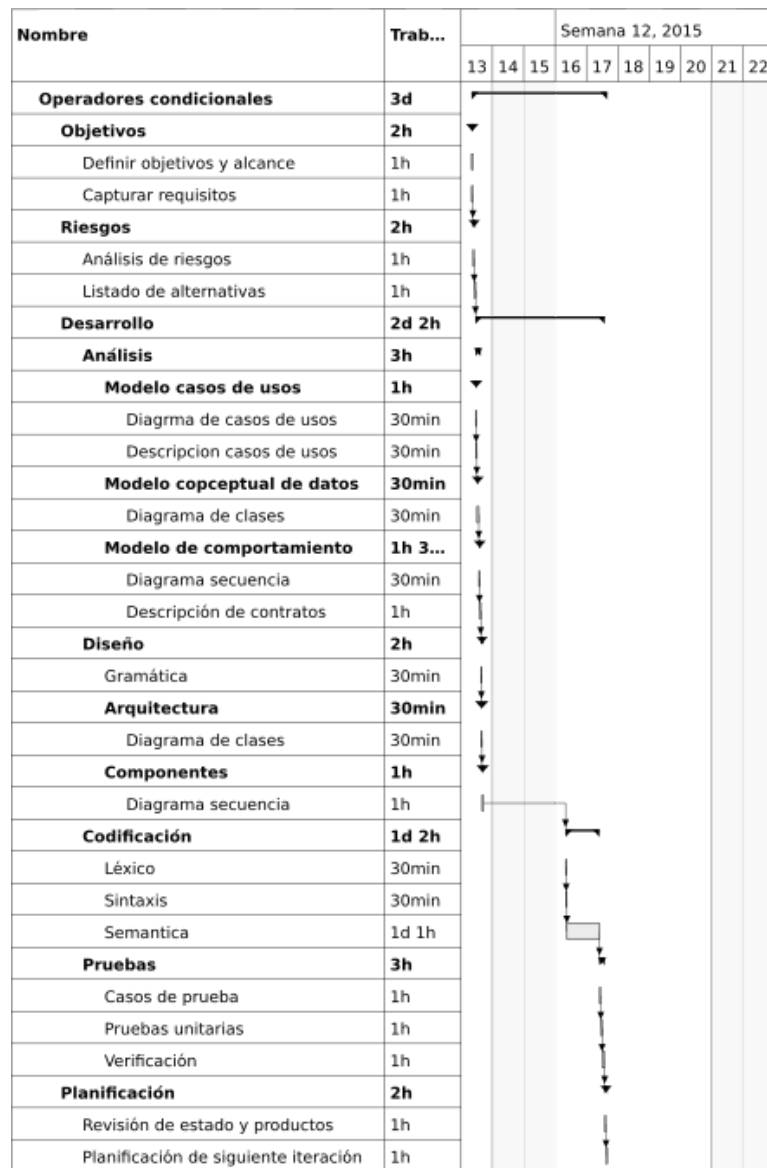


Figura 16: Expresiones condicionales: 13/03/2015 - 17/03/2015

3.17. Funciones de depuracin

Nombre	Trab...					
		18	19	20	21	22
Funciones depuración	3d					
Objetivos	2h					
Definir objetivos y alcance	1h					
Capturar requisitos	1h					
Riesgos	2h					
Análisis de riesgos	1h					
Listado de alternativas	1h					
Desarrollo	2d 2h					
Análisis	3h					
Modelo casos de usos	1h					
Diagrama de casos de usos	30min					
Descripción casos de usos	30min					
Modelo conceptual de datos	30min					
Diagrama de clases	30min					
Modelo de comportamiento	1h 3...					
Diagrama secuencia	30min					
Descripción de contratos	1h					
Diseño	2h					
Gramática	30min					
Arquitectura	30min					
Diagrama de clases	30min					
Componentes	1h					
Diagrama secuencia	1h					
Codificación	1d 2h					
Léxico	30min					
Sintaxis	30min					
Semántica	1d 1h					
Pruebas	3h					
Casos de prueba	1h					
Pruebas unitarias	1h					
Verificación	1h					
Planificación	2h					
Revisión de estado y productos	1h					
Planificación de siguiente iteración	1h					

Figura 17: Funciones de depuración: 18/03/2015 - 20/03/2015

3.18. Optimizacin de memoria

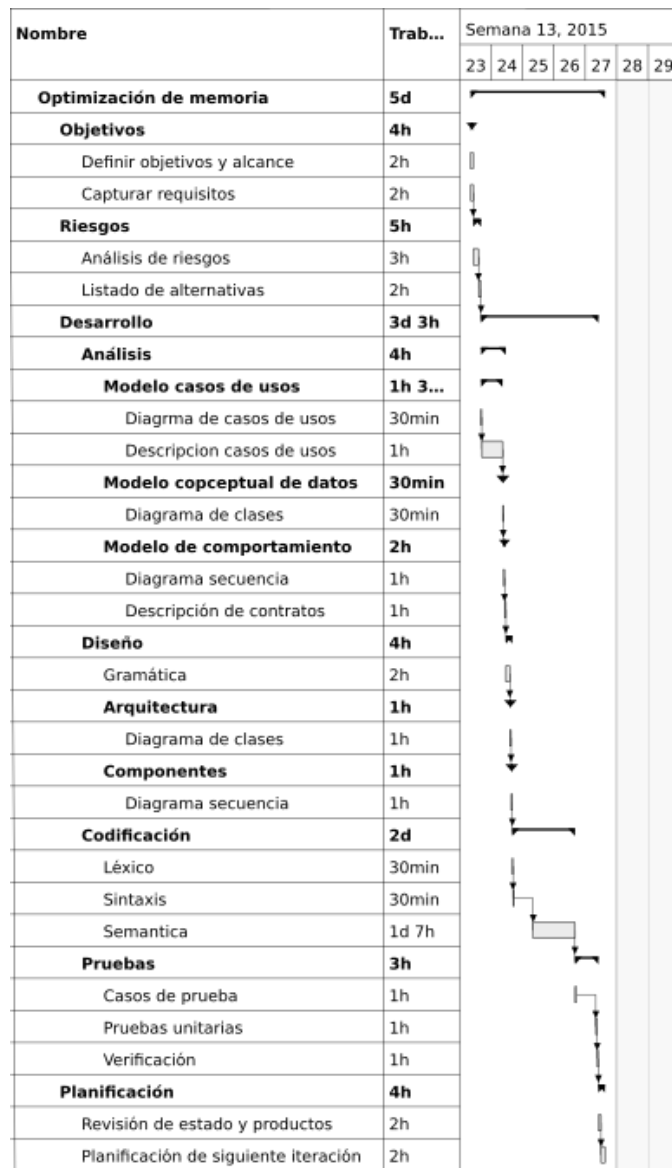


Figura 18: Optimizacin de memoria: 23/03/2015 - 27/03/2015

3.19. Sentencias de control II

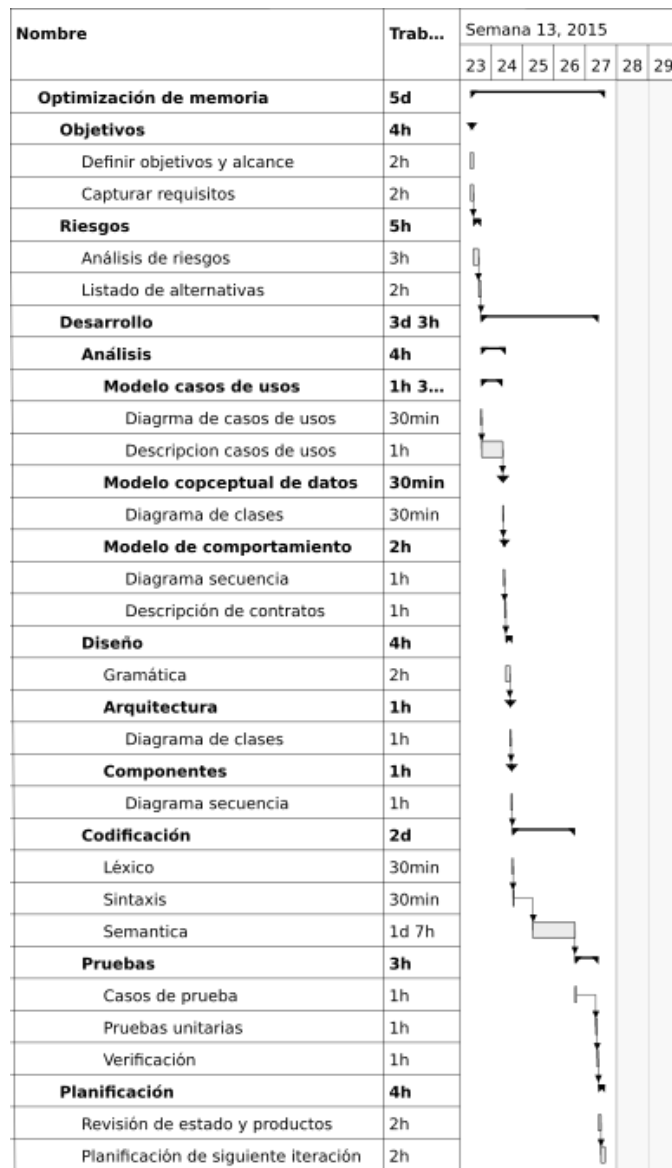


Figura 19: Optimización de memoria: 23/03/2015 - 27/03/2015

3.20. Paso de argumentos

Nombre	Trab...	Semana 15, 2015						
		6	7	8	9	10	11	12
Paso de argumento y entorno	3d							
Objetivos	2h	▼						
Definir objetivos y alcance	1h							
Capturar requisitos	1h	↓						
Riesgos	2h	↓						
Análisis de riesgos	1h	↓						
Listado de alternativas	1h	↓						
Desarrollo	2d 2h							
Análisis	3h	×						
Modelo casos de usos	1h	▼						
Diagrama de casos de usos	30min	↓						
Descripción casos de usos	30min	↓						
Modelo conceptual de datos	30min	↓						
Diagrama de clases	30min	↓						
Modelo de comportamiento	1h 3...	↓						
Diagrama secuencia	30min	↓						
Descripción de contratos	1h	↓						
Diseño	2h	↓						
Gramática	30min	↓						
Arquitectura	30min	↓						
Diagrama de clases	30min	↓						
Componentes	1h	↓						
Diagrama secuencia	1h	↓						
Codificación	1d 2h							
Léxico	30min	↓						
Sintaxis	30min	↓						
Semántica	1d 1h	↓						
Pruebas	3h	↓						
Casos de prueba	1h	↓						
Pruebas unitarias	1h	↓						
Verificación	1h	↓						
Planificación	2h	↓						
Revisión de estado y productos	1h	↓						
Planificación de siguiente iteración	1h	↓						

Figura 20: Paso de argumentos: 06/04/2015 - 08/04/2015

3.21. Interfaz de usuario II



Figura 21: Interfaz de usuario II: 09/04/2015 - 15/04/2015

3.22. Procesos I

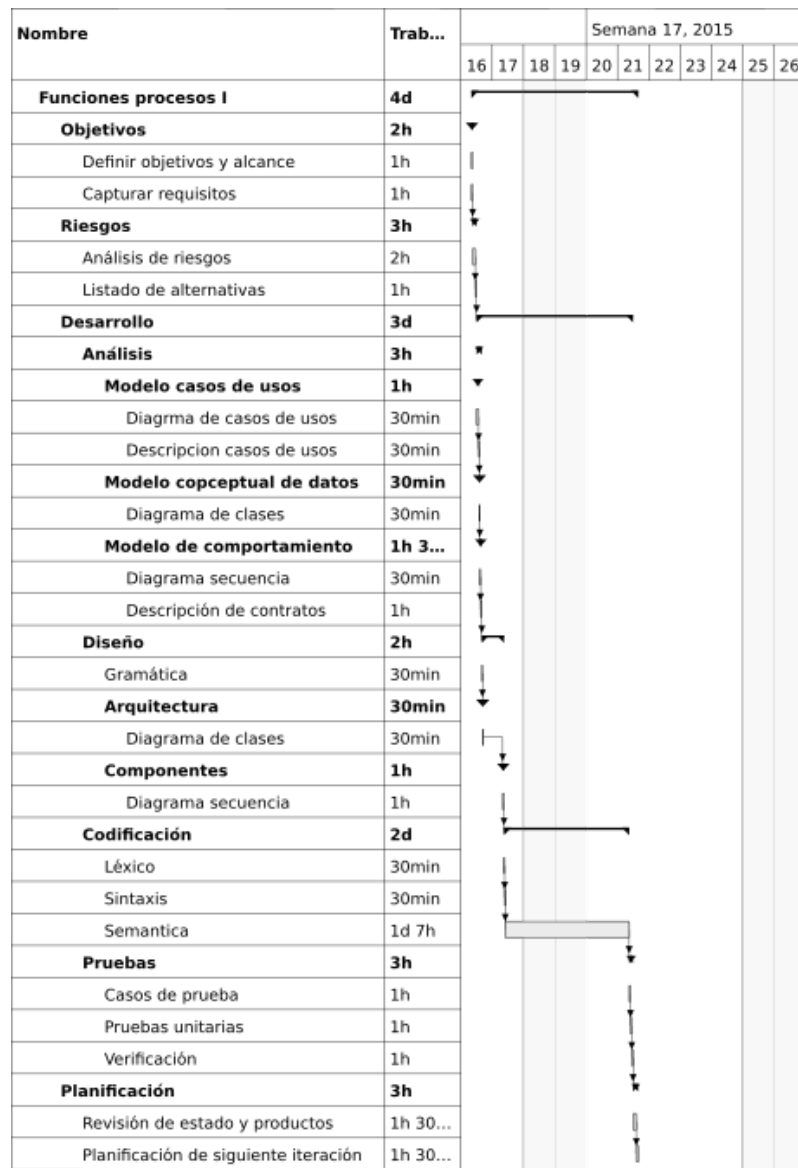


Figura 22: Procesos I: 16/04/2015 - 21/04/2015

3.23. Fechas y tiempo



Figura 23: Fechas y tiempo: 22/04/2015 - 24/04/2015

3.24. Ficheros

Nombre	Trab...	Semana 18, 2015						
		27	28	29	30	1	2	3
Funciones ficheros	3d							
Objetivos	2h	▼						
Definir objetivos y alcance	1h							
Capturar requisitos	1h	↓						
Riesgos	2h	↓						
Análisis de riesgos	1h	↓						
Listado de alternativas	1h	↓						
Desarrollo	2d 2h							
Análisis	3h	*						
Modelo casos de usos	1h	▼						
Diagrama de casos de usos	30min	↓						
Descripción casos de usos	30min	↓						
Modelo conceptual de datos	30min	↓						
Diagrama de clases	30min	↓						
Modelo de comportamiento	1h 3...	↓						
Diagrama secuencia	30min	↓						
Descripción de contratos	1h	↓						
Diseño	2h	↓						
Gramática	30min	↓						
Arquitectura	30min	↓						
Diagrama de clases	30min	↓						
Componentes	1h	↓						
Diagrama secuencia	1h	↓						
Codificación	1d 2h							
Léxico	30min	↓						
Sintaxis	30min	↓						
Semántica	1d 1h							
Pruebas	3h	↓						
Casos de prueba	1h	↓						
Pruebas unitarias	1h	↓						
Verificación	1h	↓						
Planificación	2h	↓						
Revisión de estado y productos	1h	↓						
Planificación de siguiente iteración	1h	↓						

Figura 24: Ficheros: 27/04/2015 - 29/04/2015

3.25. Extensiones

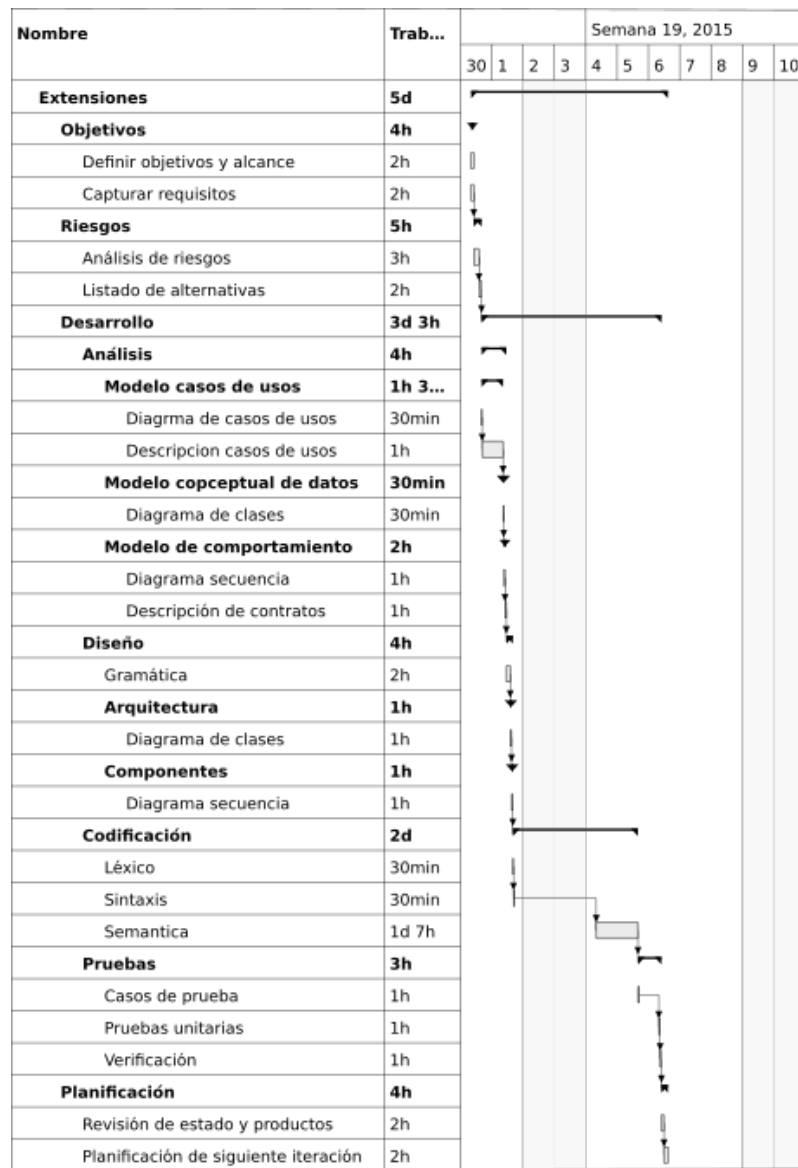


Figura 25: Extensiones: 30/04/2015 - 06/05/2015

3.26. Extensin gettext

Nombre	Trab...				
		7	8	9	10
Extensión gettext	2d				
Objetivos	1h 3...				
Definir objetivos y alcance	1h				
Capturar requisitos	30min				
Riesgos	1h 3...				
Análisis de riesgos	1h				
Listado de alternativas	30min				
Desarrollo	1d 4h				
Análisis	2h				
Modelo casos de usos	45min				
Diagrama de casos de usos	15min				
Descripción casos de usos	30min				
Modelo conceptual de datos	30min				
Diagrama de clases	30min				
Modelo de comportamiento	45min				
Diagrama secuencia	15min				
Descripción de contratos	30min				
Diseño	1h 1...				
Gramática	15min				
Arquitectura	30min				
Diagrama de clases	30min				
Componentes	30min				
Diagrama secuencia	30min				
Codificación	7h 3...				
Léxico	15min				
Sintaxis	15min				
Semántica	7h				
Pruebas	2h				
Casos de prueba	1h				
Pruebas unitarias	30min				
Verificación	30min				
Planificación	1h				
Revisión de estado y productos	30min				
Planificación de siguiente iteración	30min				

Figura 26: Extensiones gettext: 07/05/2015 - 08/05/2015

3.27. Procesos II

Nombre	Trab...	Semana 20, 2015						
		11	12	13	14	15	16	17
Funciones procesos II	2d							
Objetivos	1h 3...	▼						
Definir objetivos y alcance	1h							
Capturar requisitos	30min	↓						
Riesgos	1h 3...	▼						
Análisis de riesgos	1h	↓						
Listado de alternativas	30min	↓						
Desarrollo	1d 4h	└─						
Análisis	2h	▼						
Modelo casos de usos	45min	▼						
Diagrama de casos de usos	15min	↓						
Descripción casos de usos	30min	↓						
Modelo conceptual de datos	30min	▼						
Diagrama de clases	30min	↓						
Modelo de comportamiento	45min	▼						
Diagrama secuencia	15min	↓						
Descripción de contratos	30min	↓						
Diseño	1h 1...	▼						
Gramática	15min	↓						
Arquitectura	30min	▼						
Diagrama de clases	30min	↓						
Componentes	30min	▼						
Diagrama secuencia	30min	↓						
Codificación	7h 3...	└─						
Léxico	15min	↓						
Sintaxis	15min	↓						
Semántica	7h	▬						
Pruebas	2h	▼						
Casos de prueba	1h	↓						
Pruebas unitarias	30min	↓						
Verificación	30min	↓						
Planificación	1h	▼						
Revisión de estado y productos	30min	↓						
Planificación de siguiente iteración	30min	↓						

Figura 27: Procesos II: 11/05/2015 - 12/05/2015

3.28. Símbolos clases II

Nombre	Trab...	Semana 21, 2015											
		13	14	15	16	17	18	19	20	21	22	23	24
Símbolos clases II	5d												
Objetivos	4h												
Definir objetivos y alcance	2h												
Capturar requisitos	2h												
Riesgos	5h												
Análisis de riesgos	3h												
Listado de alternativas	2h												
Desarrollo	3d 3h												
Análisis	4h												
Modelo casos de usos	1h 3...												
Diagrama de casos de usos	30min												
Descripción casos de usos	1h												
Modelo conceptual de datos	30min												
Diagrama de clases	30min												
Modelo de comportamiento	2h												
Diagrama secuencia	1h												
Descripción de contratos	1h												
Diseño	4h												
Gramática	2h												
Arquitectura	1h												
Diagrama de clases	1h												
Componentes	1h												
Diagrama secuencia	1h												
Codificación	2d												
Léxico	30min												
Sintaxis	30min												
Semántica	1d 7h												
Pruebas	3h												
Casos de prueba	1h												
Pruebas unitarias	1h												
Verificación	1h												
Planificación	4h												
Revisión de estado y productos	2h												
Planificación de siguiente iteración	2h												

Figura 28: Símbolos clases II: 13/05/2015 - 19/05/2015

3.29. Símbolos funciones II

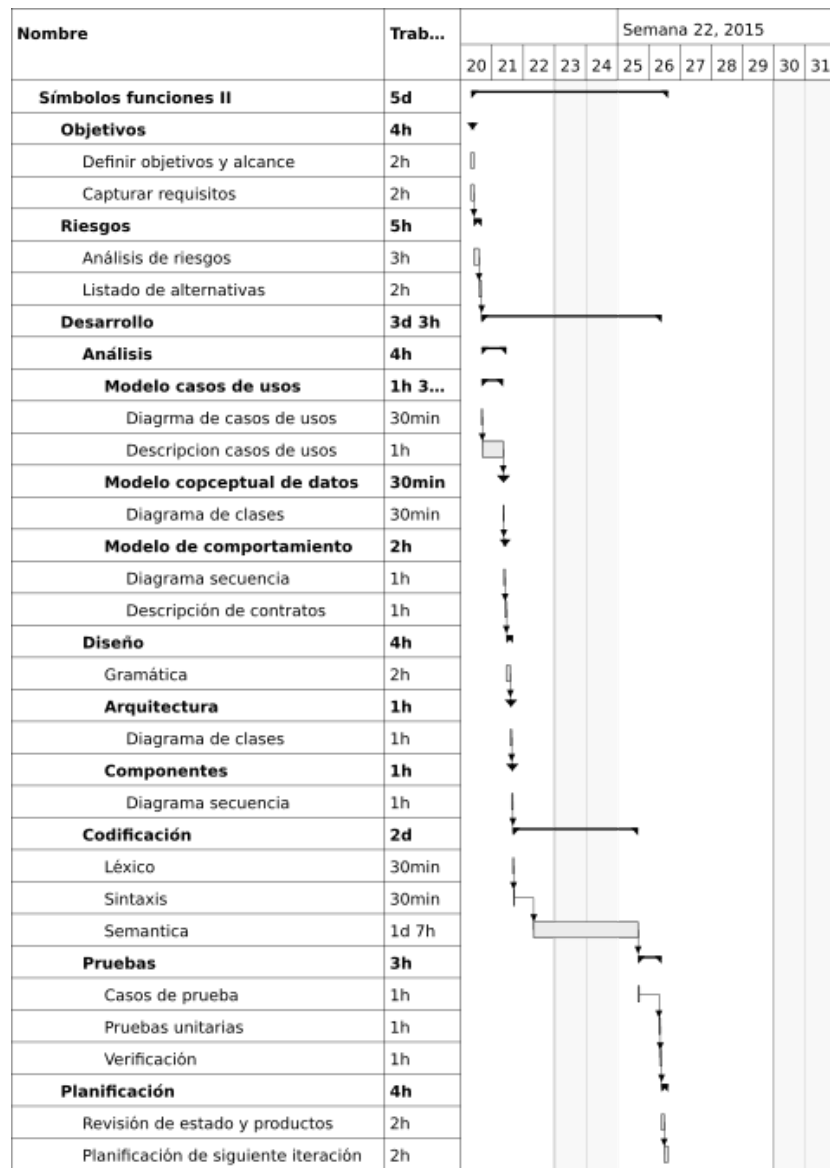


Figura 29: Símbolos funciones II: 20/05/2015 - 26/05/2015

3.30. Extensin mysql

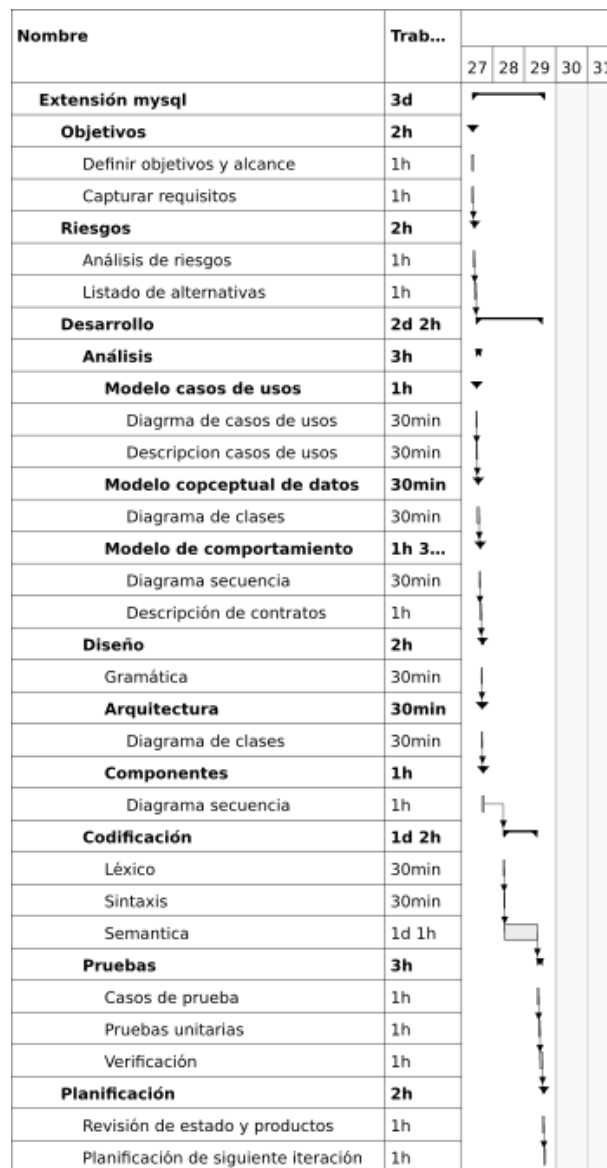


Figura 30: Extensin mysql: 27/05/2015 - 29/05/2015

4. Organizacin

Para la realizacin del proyecto se ha utilizado los siguientes recursos humanos :

Empleado 1: Director de proyecto, control de calidad.

Empleado 2: Analista, diseñador de arquitectura, diseñador de sistemas, desarrollador, tester.

Adems se han utilizado los siguientes recursos materiales:

Equipo de trabajo 1: Computadora para las tareas de gestin y administracin.

Equipo de trabajo 2: Computadora para las tareas de desarrollo y documentacin

Las herraminetas utilizadas son las siguientes:

Sistema operativo: GNU/Linux.

Entorno integrado de desarrollo: Geany.

Generador lxico: Flex.

Generador sintctico: Bison.

Compilador: GCC.

Depurador: GDB.

Herramientas para la construccin automtica: Autoconf, automake, make.

Desarrollo de diagramas: Dia, railroad diagram generator.

Control de versiones: Subversion.

Creacin de documentacin: Latex.

Planificacin: Planner.

Bibliotecas de desarrollo: readline, boost.

Comunicacin: Servicio gratuito de correo electrnico.

5. Costes

A continuacin se presenta el coste relativo a los recursos humanos invertidos en el desarrollo del proyecto. Para ello se ha tomado como referencia el documento BOE publicado el sbado 30 de noviembre de 2013 por el ministerio de empleo y seguridad social. En este documento se recoge la tabla salarial segn el convenio colectivo de la empresa Trevenque Sistemas de Informacin SL.

Empleado	Grupo	Salario anual bruto	Salario mensual bruto
Empleado 1	Directivo tecnico	26.652,92	2.221,07
Empleado 2	Programado senior	18.120,00	1.510,00

El tiempo de desarrollo asciende a 8 meses. El coste de los recursos humanos es:

Empleado 1	17.768,56
Empleado 2	12.080,00
Total	29.848,56

Para el desarrollo del proyecto se ha necesitado de dos equipos con una potencia y prestaciones medias. El coste de los recursos materiales que suponen estas computadores es:

Computador i5-4460/ 4GB/ 1TB	2x	409,00
Total		818,00

Las herramientas utilizadas en el desarrollo del proyecto tiene licencia libre. El uso de estas herramientas no suponen coste adicional para el desarrollo del proyecto.

6. Riesgos

En este punto se muestran listan los riesgos identificados. Estos pueden originar un efecto negativo en el desarrollo del proyecto. Adems se muestra la probabilidad de que estos se den y el impacto que tendran. Una vez identificados los riesgos se definen los planes seguidos para reducir los efectos derivados estos o disminuir la probabilidad de que ocurran.

El analisis de riesgos se ha llevado a cabo en cada iteracin del ciclo de desarrollo. Cada iteracin a completado la informacin mostrada. Muchos de los riesgos indicados son comunes a todas las iteraciones realizadas.

Los riesgos analizados segn el impacto que pueden ocasionar en el proyecto quedan divididos en:

Insignificantes: No merecen ser tenido en cuenta.

Tolerables: Estn dentro de un marge de aceptacin, por tanto no comprometen ni el proyecto, ni el producto ni la organizacin.

Graves: Compromete gravemente el proyecto, el producto o la organizacin.

Crticos: Amenaza la supervivencia del proyecto o el producto o la organizacin.

Para dar claridad al análisis, la probabilidad de que se dé un determinado escenario de riesgo se presenta de forma relativa de la siguiente forma:

Muy baja: $< 10\%$.

Baja: del 10 al 25 %.

Moderada: del 25 al 50 %.

Alta: del 50 al 75 %.

Muy alta: $> 75\%$.

Los riesgos quedan organizados en distintas categorías para un mejor análisis.

6.1. Riesgos tecnológicos

Son los riesgos derivados del software y hardware necesarios para el desarrollo del proyecto.

Riesgo	Probabilidad	Impacto
Los recursos no están disponibles a tiempo	Baja	Tolerable
Fallos en el sistema operativo u otros software de sistema	Baja	Tolerable
Fallos en el hardware de desarrollo	Baja	Grave
Errores de configuración	Baja	Tolerable
Actualización del sistema no realizada correctamente	Baja	Tolerable
Integridad y privacidad de los datos	Baja	Grave
Manipulación deliberada de los programas	Moderada	Grave
Las herramientas de comunicación no se encuentran disponibles	Baja	Insignificante
El repositorio de código no se encuentra disponible	Baja	Tolerable

En el caso de que los recursos materiales, hardware de desarrollo, no estén disponibles a tiempo se puede comenzar con tareas de definición y análisis. Si la demora se hace demasiado extensa se cancela el pedido del hardware y se realiza a otro proveedor. Esto puede ocasionar un retraso de días que puede ser mitigado dedicando este tiempo a tareas en las que no se precise de hardware.

Si se produce algún fallo en el software de sistema que da soporte al desarrollo este puede derivar en pérdidas de datos. La solución será gestionar la incidencia o reinstalar el sistema. La pérdida de datos se puede mitigar si se realizan copias de seguridad periódicas. Como

medida preventiva se pueden guardar puntos de restauración del sistema para prevenir o mitigar la demora de tiempo que esto podrá ocasionar.

Es posible que se produzcan fallos en el hardware sobre el cual se desarrolla el proyecto, esto puede ocasionar retrasos de entregas o la pérdida de los datos. Para prevenir y mitigar este efecto negativo se puede realizar copias periódicas de los datos y mantener los equipos en un estado de funcionamiento óptimo, bien refrigerados y sin exposición a agentes externos.

Si se produce algún error en la configuración del sistema es posible que algunas características de este dejen de funcionar, en muchos casos esto puede llegar a ser perjudicial para el proyecto. Para prevenir y mitigar este efecto se deben realizar copias de seguridad, además se debe minimizar la cantidad de software instalado en los equipos. En el caso de producirse un fallo de configuración del sistema que afecte directamente a la ejecución del proyecto se debe invertir tiempo para corregir las causas y administrar el sistema.

En algunos casos una actualización del sistema que da soporte a la ejecución del proyecto puede ocasionar fallos en el mismo o en la configuración. Para ello además de las soluciones y prácticas comentadas en los puntos anteriores, se puede minimizar el número de actualizaciones que no sean críticas o que no sean indispensables para el desarrollo del proyecto. Como medida de actuación el sistema debe ser restablecido mediante un administrador cualificado.

La integridad y privacidad de los datos se puede ver comprometida por algún uso indebido de los mismos o por agentes ajenos al proyecto. Para prevenir y mitigar los efectos de este escenario se realizarán copias de seguridad periódicas y se hará uso de un software de control de versiones. Además se protegerá el acceso a los datos mediante técnicas como claves de acceso, configuraciones de red poco permisivas (por ejemplo mediante el uso de firewalls), configuraciones óptimas del sistema (en cuanto servicios y programas), cifrado, no permitir la conexión de dispositivos extraíbles, etc. Es muy importante en este punto controlar además los permisos de publicación, transferencia y portabilidad que tienen los empleados sobre los datos, quedando registrada y validada toda operación de esta naturaleza. Como medida de actuación ante un escenario de este tipo los datos quedarán en cuarentena y se llevará a cabo un análisis forense que determine los agentes implicados y los motivos.

Es posible que en algunos casos los programas que conforman el sistema sobre el que se desarrolla sean manipulados de forma consciente o inconsciente por una persona integrante del equipo o ajena al mismo. Para prevenir este escenario se acotará el uso del equipo al propósito que este tiene, no se permitirá la instalación de software adicional, se protegerá el acceso al sistema mediante claves robustas y se usará software antivirus. Como medida de actuación se bloqueará el acceso al proyecto a cualquier software que se tenga constancia de que se encuentre manipulado o infectado por alguna clase de malware, y el equipo implicado se podrá poner en cuarentena para evaluar el impacto de lo ocurrido.

Las herramientas de comunicación son un elemento clave en el desarrollo de un proyecto. Depender de terceros para dar soporte al proyecto en este aspecto puede llegar a ser

perjudicial. Para prevenir prdidas en el servicio se puede usar clientes estables, fiables y consolidados en el mercado. Para mitigar la perdida de informacin que supondra una cada del servicio se puede hacer copias de toda la informacin enviada por estos medios. Como medida de actuacin, siempre y cuando sea necesario una comunicacin, se utilizarn otros medios de comunicacin como el telefono.

El repositorio en el cual se aloja el cdigo fuente puede sufrir cadas en el servicio. Para mitigar y prevenir este tipo de incidencias se alojar el sistema de control de versiones en alguna mquina de la infraestructura local al proyecto. Esto bloquea o dificulta el acceso remoto al cdigo, lo cual tiene sus ventajas ya que disminuye el grado de exposicin, pero hace que el acceso al cdigo est condicionado por un equipo local, y por la disponibilidad y visibilidad de este.

6.2. Riesgos personales

En esta seccin quedan registrados los riesgos asociados a las personas involucradas en el proyecto.

Riesgo	Probabilidad	Impacto
Cambios en el personal directivo	Baja	Crtico
Cambios en el personal ejecutivo	Baja	Grave

Los cambios en el equipo directivo de un proyecto pueden amenazar drsticamente la supervivencia y desarrollo de este. El nuevo personal no tiene porque compartir los mismos criterios que el anterior, y lo normal es que no conozca aspectos internos del mismo. Un cambio directivo en el proyecto puede ocasionar replanteamientos de cuestiones que ya se encontraban cerradas. Este tipo de riesgos es difcil de preveer por otros medios que no sean contratos laborales, y an as no se podra asegurar que el equipo se mantendr. Para mitigar el efecto que un cambio de este tipo podra ocasionar toda decisin directiva deber quedar analizada, argumentada y registrada.

El personal ejecutivo es esencial para el desarrollo de un proyecto, por lo que un cambio de este tipo en el equipo puede suponer un alto riesgo para el desarrollo. Al igual que en el punto anterior los contratos laborales pueden prevenir en cierto grado el riesgo. Para mitigar el efecto se deber documentar todo los procesos y productos obtenidos. Como medida de actuacin, siempre que sea posible, el anterior personal podra instruir al nuevo hasta que tome la destreza y conocimientos suficientes.

6.3. Riesgos organizativos

En esta seccin se presentan y analizan los riesgos relativos a la organizacin del proyecto.

Riesgo	Probabilidad	Impacto
Las tareas no quedan bien definidas y/o acotadas	Moderada	Tolerable
Las tareas no quedan bien distribuidas	Moderada	Tolerable
No se atribuyen responsabilidades	Baja	Tolerable
No se establece una jerarquía de prioridades	Baja	Tolerable
La comunicación entre el equipo no es suficiente	Moderada	Tolerable

Si las tareas no quedan bien definidas y acotadas se originan fallos en la comunicación y el entendimiento entre los integrantes del equipo, pudiendo en muchos casos originar trabajo extra y la consecuente dilatación en los tiempos. Para prevenir y mitigar este efecto se debe realizar una descripción minuciosa y detallada de las mismas, utilizando un lenguaje sencillo y directo. El plan de actuación ante estas situaciones será la aclaración de los puntos que fueron confusos, mientras que la parte que no llegó a entender las cuestiones planteadas no debe presionar y pedir la aclaración pertinente.

Si las tareas no se encuentran bien distribuidas puede darse exceso de trabajo por algunas de las partes, ocasionando efectos de parada en el proceso de desarrollo. Por otro lado otros recursos pueden quedar ociosos. Para evitar y mitigar este escenario se debe analizar y planificar todo el trabajo a realizar. En el caso de que una planificación incorrecta derive en problemas de este tipo se debe replantear el trabajo.

Un equipo en el que las responsabilidades no queden bien atribuidas puede originar en la falta de implicación y corrección por alguna de las partes. Para prevenir y mitigar este efecto se debe hacer una correcta planificación, involucrando al personal y hacerles entender la importancia de su trabajo.

Si no se establece una jerarquía de prioridades en las tareas puede ocasionar la pérdida de tiempo en tareas menos importantes, mientras que otras más prioritarias y de las que existan dependencias queden paradas. Para evitar esto se debe realizar una planificación y hacer entender a todo el equipo las prioridades marcadas. Bajo estas circunstancias se debe retribuir prioridades y hacer que todo el equipo sea consciente de estas.

Si no se mantiene una comunicación suficiente el proyecto puede verse perjudicado en su ejecución. El equipo directivo puede no conocer el estado verdadero del proyecto y las decisiones tomadas no contar con toda la información posible. Para mitigar y prevenir este escenario se pueden realizar reuniones constantes. Como medida de actuación ante esta situación se debe comunicar a los implicados la importancia de la información que poseen.

6.4. Riesgos de requisitos

Son riesgos que surgen de los requisitos, ya sean debido a que estos han cambiado o que no se han recojido correctamente.

Riesgo	Probabilidad	Impacto
Especificacin de requisitos insuficiente	Moderada	Grave
Captura de requisitos errnea	Moderada	Grave
Casos de uso complejos o mal redactados	Baja	Tolerable
No se han capturado todos los datos que definen o con los que trabaja el sistema	Baja	Tolerable
Aadir nuevas caractersticas sin tener en cuenta la arquitectura interna del sistema	Alta	Tolerable

Si los requisitos no son enumerados y definidos de forma efectiva puede ocasionar un sistema que no hace lo que debe hacer. Para evitar esto se llevar a cabo mltiples reuniones con el cliente para la toma de requisitos, donde el analista tomar parte activa de estas y nunca llegar a presuponer ningn aspecto que no quede bien acotado. En este punto el analista podr optar por algunas de las tcnicas conocidas para la toma de requisitos. Como plan de actuacin se deber poner en contacto con el cliente y pedir la aclaracin o especificacin de los puntos que no quedaron fijados.

De igual forma unos requisitos mal tomados puede tener como resultado un producto que no cumple con las expectativas. Para prevenir y evitar este escenario el analista debe ser muy conciso y minucioso en la toma de requisitos, haciendo que todo quede claro por ambas partes.

Aunque se realice una toma de requisitos completa, queda la posibilidad que el analista no transmita correctamente qu va a hacer el sistema para dar solucin a estos. Se deber poner especial atencin en que los casos de usos queden bien redactados, en un lenguaje simple y con la suficiente sencillez y completud.

Una de las principales actividades de un sistema informticos es procesar datos, dado que el valor de la informacin viene atribuido por los datos que la forman. Si los datos sobre los que opera o definen el sistema no son determinados con exactitud el valor que este aporta disminuye. En la toma de requisitos se ha de poner especial atencin en recoger los datos que construyen el modelo de datos de una forma completa y exacta.

Definir nuevos requisitos del sistemas conociendo la estructura y caractersticas internas de este es una ventaja, no obstante no siempre es posible hacer que los nuevos requisitos se adapten a las posibilidades que el software brinda, ni el cliente tiene porque conocer las restricciones que determinadas decisiones de diseo han impuesto. Para prevenir y mitigar este efecto, y que el software sea ampliable en caractersticas de una forma abierta, se deben disear soluciones verstiles, flexibles al cambio y con las minimas restricciones asociadas.

6.5. Riesgos de soluciones

Son riesgos asociados con el diseo de soluciones

Riesgo	Probabilidad	Impacto
Diseño de solución errónea o demasiado restrictiva	Moderada	Grave
Cambios versiones y características de las soluciones software utilizadas	Baja	Tolerable
No se encuentra un léxico adecuado	Baja	Tolerable
La gramática es confusa	Moderada	Tolerable
No se ha seguido el principio de reutilización de código	Baja	Tolerable
Código inteligible	Baja	Tolerable
Errores de seguridad	Moderada	Grave

El tomar como base una solución errónea puede ocasionar que el software no produzca los resultados esperados. Si los errores son detectados a tiempo el impacto puede ser bajo, pero si el error persiste en varias iteraciones del proceso de desarrollo puede ocasionar pérdidas cuantiosas. Para prevenir esta situación se deben pasar auditorías de calidad en todas las iteraciones, no solo en el sentido de que la solución se ha aplicado correctamente, sino también de que estas cumplen unos criterios mínimos de optimización, seguridad, flexibilidad, etc. Cuando se detecta que una solución tomada no es correcta se debe analizar el costo de las distintas alternativas para corregirlo y si fuera necesario aplicar la más pertinente para el caso.

Muchas soluciones software utilizadas pueden cambiar de versión y con ella las características que ofrecen, pudiendo quedar algunas de ellas eliminadas. Esto puede ocasionar que el sistema desarrollado no cumpla con lo que antes se hacía. Para evitar esta situación siempre se guardan una versión estable de las soluciones software tomadas. Antes de actualizar el software que da solución a algún aspecto del sistema se debe tener en cuenta el impacto que esta operación tendrá. En el caso de ser necesario actualizar se debe apagar el sistema para el uso de la nueva versión.

Para un lenguaje de programación el disponer de un léxico sencillo, fácil de recordar y común en otros lenguajes es un requisito necesario, si esto no se logra el resultado es un lenguaje que el mercado no está dispuesto a usar. Para prevenir esto se ha de analizar y medir cada palabra que componga el léxico, sometiendo a evaluación por todo el equipo y por personas ajenas al mismo. Es posible tomar como referencias otros lenguajes disponibles en el mercado. Si se detecta o se determina que una determinada del léxico no es adecuada se debe de cambiar.

De igual forma que en el punto anterior, una gramática confusa puede originar un lenguaje poco usado, cuyo costo de aprendizaje sea elevado. Nuevamente para evitar este escenario se debe analizar y evaluar la estructura de la gramática elegida. Si se detecta una gramática confusa se debe analizar y proponer otras alternativas.

Una solución desarrollada que no sea reutilizable implica la múltiple realización del trabajo. Para mitigar y prevenir este efecto se debe modularizar y acotar toda función que se pueda reutilizar. Las funciones, clases y demás recursos de programación deben tener un propósito

concreto y bien definido. Si se detecta que una determinada parte del sistema se está reescribiendo por no seguir un principio de reutilización se debe invertir tiempo en invertir esta situación.

Hacer que el sistema se conforme de partes de código que no sean fáciles de entender o seguir puede ir en contra del mantenimiento del proyecto y de la corrección del sistema. Para prevenir este escenario se debe seguir unas reglas de estilo uniforme que construyan un código limpio y fácil de entender. Al detectar este tipo de código se ha de invertir tiempo en clarificar la sección afectada.

Al escribir programas complejos y extensos es muy común que se den fallos de seguridad que permitan explotar vulnerabilidades como podría ser el desbordamiento de buffer o algún tipo de inyección. Para evitar esto se podrá realizar auditorías de seguridad al código desarrollado. Si se detecta algún fallo de seguridad en la fase de desarrollo este debe ser notificado y corregido.

6.6. Riesgos de costes, tiempos y recursos

En esta sección se exponen los riesgos derivados de cálculos en cuanto a costes, recursos y tiempo.

Riesgo	Probabilidad	Impacto
Número de empleados insuficiente	Baja	Tolerable
Recursos necesarios no previstos	Baja	Tolerable
Planificación optimista	Moderada	Tolerable
La planificación de la siguiente iteración tarda más de lo esperado	Moderada	Tolerable
El presupuesto es insuficiente para afrontar el desarrollo	Baja	Crítica

Si el número de empleados contratados para el proyecto es insuficiente el tiempo necesario para el desarrollo del mismo puede incrementar considerablemente. Por otro lado contratar empleados conlleva un coste monetario. Es por tanto que se debe llegar a una configuración óptima. Para prevenir este escenario se ha de llevar a cabo una planificación realista llevada a cabo desde la experiencia y de una forma objetiva. Si la realidad difiere de la planificación realizada y se precisa de más empleados, habrá que asumir el coste de contratación además del tiempo de incorporación que varía en función del perfil.

Es posible que en la planificación inicial no se contemple la totalidad de los recursos materiales necesarios. Esto podría originar retrasos en las entregas y tiempos de espera. Para mitigar este efecto se puede tener un fondo reservado para posibles gastos adicionales no previsto, no obstante lo ideal es hacer una planificación exacta en cuanto a los recursos requeridos.

Es común que muchas planificaciones se hagan desde un punto de vista optimista, esto conlleva desviaciones entre el tiempo calculado, los recursos invertidos y los compromisos llegados con lo que se materialice en la realidad. Para evitar este escenario, se debe llevar a cabo una planificación realista, con márgenes de actuación.

Retardar la planificación de la siguiente iteración puede llegar a ser un problema que bloquee el proceso de desarrollo, dado que no se han marcados las pautas para continuar. Para mitigar esto la planificación puede desarrollarse de forma paralela al recorrido de la iteración actual.

Es posible que el presupuesto para la realización del proyecto quede insuficiente, esto puede ocasionar el cese de la actividad del mismo. Como método de previsión se pueden estudiar varias alternativas de financiación.

7. Aseguramiento de calidad

Para asegurar una calidad aceptable del producto y los procesos llevados a cabo en el desarrollo se ha tomado como esquema el conjunto de normas ISO 9000. El sistema de gestión de la calidad está formado por procesos y es en sí mismo un proceso en el que ingresan los requisitos del sistema y se obtiene un producto que cumple los requisitos y satisface al cliente.

A continuación se describen los procesos y actividades seguidas para asegurar la calidad.

- Se implementa un ciclo de vida evolutivo en el que se examina y documenta cada decisión tomada, al final de cada iteración se comprueba la calidad del producto obtenido.
- El equipo directivo se asegura de que los requisitos del cliente se determinan y cumplen. El objetivo de esto es aumentar la satisfacción del mismo.
- El equipo directivo realiza una planificación en la que se capturen las acciones a seguir para alcanzar los objetivos perseguidos.
- Los roles y responsabilidades del equipo quedan definidos y atribuidos en cada iteración del ciclo.
- En cada iteración se examina las pruebas de calidad realizadas, la retroalimentación del cliente y la conformidad del producto.
- En cada iteración se contemplan las mejoras del producto en relación con los requisitos y necesidades del cliente.
- Todo el personal tiene los conocimientos y el entrenamiento adecuados para realizar la tarea que en la planificación se le ha sido asignada.

- Se utilizan entrevistas periódicas con el cliente para capturar los requisitos necesarios en cada iteración. Además de validar la corrección del producto obtenido
- Para la toma de requisitos se ha utilizado técnicas de objetivos medibles, de forma que los requisitos impuestos por el cliente se ven como objetivos generales. Estos son analizados repetidamente para obtener los requisitos críticos para el funcionamiento del sistema. En cada iteración se refinan los requisitos generales y se plantean nuevos, aplicándose la misma técnica para determinar los tomados en la iteración.
- Se presentan modelos de caso de uso que deben ser comprendidos y validados por el cliente.
- El diseño del producto obtenido en cada iteración queda documentado mediante modelos.
- Se realiza una revisión, verificación y validación de los diseños propuestos.
- La actividad de desarrollo del producto queda bien documentada en el propio código.
- En cada iteración se definen los casos de pruebas, estas serán llevadas a cabo para comprobar la corrección del producto.

A continuación los criterios para la aceptación o rechazo de los productos obtenidos en cada fase del desarrollo.

- Si se detecta algún requisito ambiguo o mal especificado este queda rechazado. En este caso se debe concertar una entrevista con el cliente.
- Para la aceptación de los requisitos estos deben ser simples y sencillos, estar bien descritos y especificados de forma atómica.
- Si se detecta un caso de uso mal explicado o confuso este será rechazado.
- Los casos de uso para su aceptación deben estar completamente explicados, de una forma esquemática y fácil de entender por el cliente. Además debe quedar bien definidos los actores involucrados y las relaciones entre estos.
- Los modelos de datos serán aceptados si son claros, completos y se encuentran bien estructurados.
- Los diseños de soluciones serán aceptados si cumplen los criterios de versatilidad, adaptación, optimización y claridad impuestos.
- Si en el diseño de la gramática se da alguna ambigüedad esta debe ser redefinida.
- Si la elección de una palabra del léxico es confusa o demasiado compleja esta debe ser redefinida.
- Si alguna clase no cumple el principio de única responsabilidad será rechazada.

- Si alguna clase no cumple el principio abierto/cerrado ser rechaza.
- Si alguna clase no cumple el principio de sustitucin de Liskov ser rechaza.
- Si alguna clase no cumple el principio de segregacin de la interfaz ser rechaza.
- Si alguna clase no cumple el principio de inversin de dependencias ser rechaza.
- Si algn mdulo de cdigo fuente desarrollado no cumple las reglas de estilo este ser rechazado.
- Si algn mdulo de cdigo fuente no se encuentra debidamente docuemtado ser rechazado.
- Si algn mdulo de cdigo fuente no sigue el principio de reutilizacin ser rechazado.
- Si se detecta que no se han capturados todos los casos de prueba escenciales se produce un rechazo.
- Para la aceptacin se deben completar satisfactoriamente todas las pruebas unitarias.