

ÍNDICE DE LA MEMORIA QUE HA DE ENTREGAR EL ALUMNO

1. Portada. Datos del proyecto: Título, Ciclo, Curso, Fecha, Centro, Logo del centro. Datos del alumno y datos de los profesores tutores

2. Documento Descripción del proyecto

2.1. Contexto del proyecto

2.1.1. Ámbito y entorno

2.1.2. Análisis de la realidad.

2.1.3. Solución y justificación de la solución propuesta

2.1.4. Destinatarios.

2.2. Objetivo del proyecto

2.2. Objetivo del proyecto en lengua extranjera.

3. Documento de Acuerdo del proyecto.

3.1. Requisitos funcionales y no funcionales / **Historias de usuario**

3.2. Tareas

3.3. Metodología a seguir para la realización del proyecto.

3.4. Planificación temporal de tareas.

3.5. Presupuesto (gastos, ingresos, beneficio)

3.7. Análisis de riesgos

3.6. Contrato/Pliego de condiciones

4. Documento de análisis y diseño

4.1 Modelado de datos. Análisis y diseño de la base de datos

4.1.1 Diagrama E/R

4.1.2 Diagrama relacional

4.2 Análisis y diseño del sistema funcional (Diagramas que procedan como el de casos de uso, de clases, de secuencia, de actividad, de estados, de flujo...)

4.3 Análisis y diseño de la interfaz de usuario. Mockups.

4.4 Diseño de la arquitectura de la aplicación

4.4.1 Tecnologías/Herramientas usadas y descripción de las mismas

4.4.2 Arquitectura de componentes de la aplicación

5. Documento de implementación, pruebas e implantación del sistema

5.1 Implementación

5.2 Pruebas

5.2.1 Pruebas unitarias

5.2.2 Pruebas funcionales

5.3. Instalación/Despliegue y configuración.

5.4. Manual de usuario

6. Documento de cierre

6.1. Resultados obtenidos y conclusiones.

6.2. Diario de bitácora

7. Bibliografía.

8. Anexos.

DOCUMENTO DE DESCRIPCIÓN DEL PROYECTO

En este apartado realizaremos una descripción detallada del proyecto

CONTEXTO

En este apartado se puede incluir cuatro subapartados:

- **Ámbito del proyecto**

Describe la empresa/cliente para la cual realizas el proyecto, departamentos, situación a qué se dedican... Si es un proyecto que no va dirigido a un cliente en concreto y es por motivación propia explícalo y di qué te ha motivado a realizar el proyecto.

- **Análisis de la realidad**

Realiza también un **análisis de la situación actual** del sistema y **necesidades** encontradas, críticas al sistema actual si lo hubiese...

Deberás especificar si el proyecto va a ser una actualización/ampliación/adaptación de un proyecto ya desarrollado o bien partes de cero. Si es una actualización deberás dejar claro qué es lo que hay hasta ese momento, si vas a emplear componentes, diseños o herramientas ya disponibles... Si en el mercado hay aplicaciones similares analízalas...

- **Solución y justificación de la solución propuesta.**

Soluciones que quieres aportar y justificación de la solución elegida.

- **Destinatarios.**

Usuarios destinatarios del proyecto. Puedes hablar del rol que tendrán cada grupo de usuarios.

OBJETIVO

Descripción general de lo que pretende ser tu aplicación. No hace falta detallar demasiado la funcionalidad ya que luego se hará en la lista de requisitos. Imagina que tienes que explicar a alguien en un párrafo qué aplicación vas a desarrollar...

Puede empezar: “El objetivo del proyecto es desarrollar una aplicación web/móvil/...”

Evitar la utilización del término “página web” no hacemos páginas web sino aplicaciones web...

DOCUMENTO DE ACUERDO DEL PROYECTO

A la hora de describir la funcionalidad del proyecto podemos utilizar o bien requisitos (se utilizaban en metodologías tradicionales) o bien historias de usuario (asociados a las metodologías ágiles)

LISTA DE REQUISITOS FUNCIONALES Y NO FUNCIONALES

Tendrá formato tabla, y cada requisito identificado por las siglas RF (requisito funcional) O RNF (requisito no funcional) seguido de un número identificativo y una descripción del requisito. Al finalizar el proyecto se realizarán las pruebas del sistema y se validará si cada requisito se ha cumplido.

REQUERIMIENTOS FUNCIONALES

Los requisitos funcionales establecen los comportamientos del sistema. Un requisito funcional define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas.

Ejemplo

RF-1 El sistema debe permitir agregar clientes. Dado el nombre, la edad, la dirección y el teléfono del cliente se dará de alta el cliente en la base de datos con un identificador único.

RF-2 El sistema debe permitir modificar clientes. Dado el id de cliente, se permitirá la modificación de cualquier campo.

....

REQUERIMIENTOS NO FUNCIONALES

Un requisito no funcional es un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento.

Algunos ejemplos de requisitos no funcionales típicos son los siguientes:

- rendimiento
- disponibilidad
- accesibilidad
- usabilidad en distintos navegadores
- estabilidad
- portabilidad
- costo
- operatividad
- interoperabilidad
- escalabilidad
- concurrencia
- mantenibilidad
- interfaz
- seguridad

- posicionamiento
- codificación
- formatos

Ejemplo:

RNF-1 El sistema debe ser de fácil navegabilidad.

RNF-2 El sistema debe ser seguro y mantener la integridad de la información.

RNF-3 El sistema debe acceder de forma correcta y rápida a la información.

RNF-4 El sistema debe soportar tanto plataformas Windows como Unix.

RNF-5 El sistema debe mantener copias de respaldo para evitar pérdidas de información.

HISTORIAS DE USUARIO

Una historia de usuario es una representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario. Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos (acompañadas de las discusiones con los usuarios y las pruebas de validación). Si optas por una metodología ágil, puedes sustituir la lista de requisitos por historias de usuario --> puedes leer más aquí: <https://www.genbeta.com/desarrollo/historias-de-usuario-una-forma-natural-de-analisis-funcional>

La manera más estándar de construir el enunciado es:

Como [tipo de usuario]

Quiero [hacer, conseguir, obtener algo]

Para [el motivo por el que quiero hacer el Quiero]

ID	Historia de Usuario	Criterios de aceptación
HU1	Como Vendedor, quiero registrar los productos y cantidades que me solicita un cliente para crear un pedido de venta.	<p>Quiero poder dar de alta un pedido rellenando el formulario de alta de pedidos.</p> <p>El formulario debe poder crear una lista de productos con cantidad.</p> <p>El resultado devolverá un id de pedido.</p> <p>No se debe poder dar de alta productos sin cantidad.</p> <p>Cuando pulse “Aceptar o Grabar” debe de registrarse el pedido en el sistema y desencadenar el flujo de aprobaciones.</p>
HU2	Como Supervisor de ventas, quiero consultar un listado de los pedidos de	Cuando pulse Ver pedidos se debe mostrar una lista ordenada por número de

	venta que han sido registrados.	pedido....
HU3	Como Gerente de ventas, quiero consultar los pedidos de venta procesados clasificándolos por vendedor, región y líneas de producto.	...
HU4	Como Analista de logística, quiero seleccionar un pedido de venta y enviarlo al almacén para que procedan con su preparación.	...

DEFINICIÓN DE TAREAS

En su núcleo, un proyecto es un conjunto de tareas, cada una de las cuales representa una pieza de trabajo que debe realizarse para completar el proyecto de forma satisfactoria. En esta fase se definen las tareas generales a realizar en el proyecto.

Ejemplo

- 1) Elección proyecto a realizar. Investigar sobre contexto y objetivo principal del proyecto.
- 2) Entrevistas con cliente para recogida de requisitos funcionales y no funcionales. Creación de historias de usuario. Realización de diagramas de casos de uso.
- 3) Descripción de tareas y planificación temporal de las mismas.
- 3) Realización del presupuesto y acuerdo del proyecto.
- 4) Análisis del proyecto. Realización de diagramas.
- 5) Diseño del proyecto. Realización de diagramas.
- 6) Implementación del proyecto. Aquí puedes concretar tareas concretas de tu proyecto.
- 7) Pruebas de la aplicación
- 8) Implantación

METODOLOGÍA

La metodología que vas a seguir en el proyecto. Una vez que ya tienes las tareas hay que decidir qué metodología vamos a aplicar para realizar las tareas.

Puedes adaptarte a los hitos establecidos por el tutor, que aconseja un modelo de ciclo de **vida iterativo incremental** aplicándolo con una metodología ágil en la que se establecen dos sprints, es decir, dos entregas al cliente en las fechas X e Y (estas fechas te las proporcionará el profesor)

Dentro de cada sprint se desarrollan las diferentes fases del ciclo de vida de desarrollo del software, desde el establecimiento de requisitos/historias de usuario pasando por el análisis y el diseño hasta la implementación, pruebas y despliegue de la aplicación.

PLANIFICACIÓN TEMPORAL DE TAREAS

La organización de tareas en hitos y fases proporciona cierta estructura al proyecto y lo hace más fácil de evaluar en cuanto a su progreso.

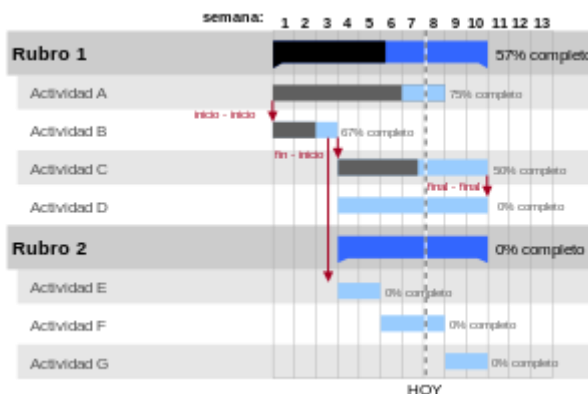
Realizaremos la planificación temporal del primer sprint y del segundo sprint de acuerdo al calendario establecido de las distintas tareas (ver fechas entrega para el curso actual). Se deberán marcar dichos hitos en vuestro diagrama. Además de los hitos establecidos por el cliente (o en su defecto tutor de prácticas) habrá que hacer una planificación temporal de las tareas a realizar en cada Sprint. A cada tarea habrá que asignar una temporalización. Por ejemplo en el primer Sprint podríamos señalar la temporalización de las siguientes tareas:

- 1) Elección proyecto a realizar. Investigar sobre contexto y objetivo principal del proyecto.
- 2) Recogida de requerimientos y creación de historias de usuario. División de las historias de usuario en 2 sprints, según la prioridad de las mismas.
- 3) Presupuesto y acuerdo del proyecto
- 4) Análisis del primer sprint del proyecto
- 5) Diseño del primer sprint del proyecto
- 6) Implementación del primer sprint del proyecto
- 7) Pruebas
- 8) Entrega a cliente
- 9) Análisis del segundo sprint del proyecto
- 10) Diseño del segundo sprint del proyecto
- 11) Implementación del segundo sprint del proyecto
- 12) Pruebas

Cada una de estas tareas tendrán a su vez subtareas. Por ejemplo la implementación podría dividirse en fases según el proyecto elegido por ejemplo Implementación alta usuarios, implementación modificación usuarios... La fase de diseño tendrá los mockups, diseño bbdd....El nivel de detalle depende del alumno.

Para representar tareas a lo largo del tiempo es muy útil realizar un diagrama de Gannt. Se puede utilizar software tipo Gantt Project o similares.

El diagrama de Gantt es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.



Si usas metodologías ágiles puedes echar un ojo a programas tipo StoriesOnBoard...igual te son más útiles, o puedes usar cualquier otro software de gestión de tareas (tipo Jira).

PRESUPUESTO y CONTRATO

Creamos el presupuesto (tal y como aprendisteis en EIE). Importante estimar el número de horas que os va a costar el proyecto y en función de eso, el tipo de las tareas y los gastos fijos/variables extraer el presupuesto.

ANÁLISIS DE RIESGOS

Se identifican los riesgos inherentes a la ejecución del proyecto definiendo el plan de prevención de riesgos y los medios y equipos necesarios.

CONTRATO

Se puede poner como anexo de la memoria. Iría enlazado al presupuesto. Creación y firma del contrato del proyecto con el cliente (según modelo colgado en la plataforma u otro modelo). Se puede añadir un apartado de pliego de condiciones si se desea.

DOCUMENTO DE ANÁLISIS Y DISEÑO DEL SISTEMA

El objetivo fundamental de este apartado consiste en la representación explícita de todas las funciones y subfunciones de que consta el sistema que se vaya a desarrollar así como la estructura de las relaciones entre las mismas y la descripción de los conjuntos de datos del sistema, etc.

En este apartado se realizará la definición del sistema desde el punto de vista lógico o conceptual exclusivamente prescindiendo de los aspectos tecnológicos.

Debido a la gran variedad de proyectos que se pueden realizar, el análisis dependerá del tipo de proyecto y la metodología a usar (Si se usan metodologías orientadas a objetos se hará uso de UML, pero si se usan metodologías ágiles quizá se pueda prescindir de ciertos diagramas...) Aun así cuando sea aplicable se deberá incluir información sobre:

- Análisis de los casos de uso. Diagrama de casos de uso para mostrar de forma gráfica los requisitos y los actores.
- Modelado de datos. Análisis y diseño de la base de datos: Diagrama E/R de base de datos, diagrama relacional de bd..(se puede usar Workbench mysql)
- Diseño de la interfaz de usuario: Mockups de la aplicación se puede utilizar Balsamic Mockup o cualquier otro con el que se haya trabajado.
- Diseño del sistema. Podéis meter todos los diagramas UML que queráis (diagrama clases si oo, diagrama flujo datos, eventos, actividades, secuencia...) para el diseño del código de la aplicación. Puedes meter todo aquel diagrama que te sirva para analizar la aplicación desde el punto de vista lógico.
- Diseño de la arquitectura de la aplicación
 - Tecnologías/Herramientas usadas y descripción de las mismas. Se puede añadir una justificación de por qué has elegido esas tecnologías y/o herramientas.
 - Arquitectura de componentes de la aplicación. Se detallará mediante esquemas, dibujos o texto los componentes lógicos y físicos de tu aplicación. Es decir si usas un framework, o un modelo de datos en concreto tendrás una organización física del código fuente por carpetas y o paquetes impuesta por dicho framework/modelo. Por ejemplo en el paquete dao las clases que acceden a bd, en el paquete html los htmls, en el paquete js...Hablar del MVC o cualquier otro que uses...

DOCUMENTO DE IMPLEMENTACIÓN, PRUEBAS E IMPLANTACIÓN DEL SISTEMA

IMPLEMENTACIÓN DEL SISTEMA

Unidades de programación utilizadas (módulos, funciones, procedimientos, scripts, etc.) que merezca la pena ser descritas. Si se quiere incluir código fuente se hará siempre en **Anexos**.

Se puede también incluir un listado organizado de los ficheros/módulos/paquetes con código fuente acompañados de una breve descripción de lo que contienen, si bien esto siempre hay que hacerlo en la cabecera de los ficheros como comentario, se puede incluir aquí todo lo que consideréis relevante.

También se puede comentar aquí el sistema de control de versiones que usas (Github, bitbucket...)(obligatorio)

PRUEBAS

- **PRUEBAS UNITARIAS**

Se aconseja la realización y utilización de pruebas unitarias para ir verificando los diferentes módulos a medida que se van desarrollando. Aunque no es necesario su documentación...

- **PRUEBAS FUNCIONALES=CRITERIOS DE ACEPTACIÓN**

El objetivo de este apartado es demostrar que el sistema globalmente cumple con las especificaciones definidas en los apartados anteriores.

Para ello será obligatorio documentar las pruebas funcionales de la aplicación. Se denominan pruebas funcionales o Functional Testing a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con los requisitos para los cuales han sido creados. El enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida.

Para la documentación de esta parte se puede seguir el siguiente formato

Requisito RF1	
Precondición	
Datos de entrada	
Datos de salida	

Requisito RF5 El sistema permite editar un usuario	
Precondición	Desde el listado de usuarios se pulsa el botón edición Usuario de la tabla usuarios
Datos de entrada	Se modifica los valores en los campos nombre, email, password y repetición de password y se selecciona rol
Datos de salida	a) La pantalla debe tener rellenados los campos con los datos del

	<p>usuario seleccionado. OK</p> <p>b) Si ya existe un usuario con ese email se le indica en un mensaje de error. OK</p> <p>c) Si no rellena los campos obligatorios de nombre, email y contraseña, se le indica en mensaje de error. OK</p> <p>d) Si las contraseñas no coinciden se le indica en un mensaje de error. OK</p> <p>e) En caso de que la validación sea correcta, se modifica el registro en la tabla Users y se cierra la ventana de edición de usuario indicándole que el usuario se ha modificado correctamente. OK</p>
--	---

Si bien si has hecho historias de usuario, los criterios de aceptación te servirán como enunciados de las pruebas.

INSTALACIÓN/DESPLIEGUE Y CONFIGURACIÓN

En este apartado se incluirá el proceso de instalación o despliegue de la aplicación para la implantación de la aplicación en el entorno donde va a funcionar (servidor web, móvil, ERP,...) así como el proceso de configuración necesario para la implantación.

MANUAL DE USUARIO Y/O AYUDA INTEGRADA EN LA APLICACIÓN

Se deberá incluir un manual de usuario de la aplicación y/o desarrollar una ayuda integrada en dicha aplicación. En éste último caso, indicar que se ha desarrollado y cómo acceder a ella.

En este apartado se incluirán los subapartados que el alumno considere necesarios para que un potencial usuario de su proyecto pueda adquirir con facilidad las nociones necesarias para su utilización. Se recomienda incluir entre otros los siguientes apartados:

- Descripción general del sistema
- Puesta en marcha
- Manejo de los menús o acceso a cada módulo
- Gestión periódica del sistema
- Realización de copias de seguridad
- Mensajes de error
- Glosario de términos

DOCUMENTO DE CIERRE

RESULTADOS OBTENIDOS Y CONCLUSIONES

El cierre de un proyecto es la culminación del proceso proyectual, y el momento de hacer balance del mismo. Durante el cierre se advierte cómo de bien o de mal se ha terminado y, en especial, si se han alcanzado los objetivos previstos.

Objetivos principales

- Diagnosticar el funcionamiento, tratando de analizar las desviaciones entre las previsiones iniciales y el resultado
 - Analizar si se han alcanzado la funcionalidad prevista.
 - Analizar la desviación en cuanto a la planificación en horas del proyecto
- Conocimientos adquiridos.
- Posibilidades de mejora.

DIARIO DE BITÁTORA

Recogerá todo el seguimiento del proyecto. Cada semana las tareas realizadas. Se podrá considerar cuaderno de bitáctora los commits de github.

PAUTAS PARA EL DESARROLLO DEL PROYECTO

Uso obligatorio de un sistema de gestión de versiones tipo GIT O BITBUCKET

Uso obligatorio de la plantilla de formato

PAUTAS PARA LA ENTREGA DEL PROYECTO

Es necesario puntualizar que una memoria sin un estilo definido, mal estructurada, mal redactada, con faltas de ortografía, y sin un índice de contenidos, bajará la nota, así que es importante cuidar la presentación de la misma. Debéis utilizar el documento de plantilla de la plataforma.

IMPORTANTE: Aunque todos los documentos (Documento Descripción del proyecto, Documento Análisis...etc) se entregan conjuntamente en la memoria, cada documento irá separado del resto por una hoja de portada que contenga únicamente el título del documento al que hace referencia.

Depósito del proyecto

La evaluación del módulo profesional de proyecto requerirá la entrega en secretaría del centro de una memoria escrita en formato digital y/o en formato impreso y del material complementario (código fuente y otros) en la fecha establecida para ello y conforme a los apartados establecidos por el tutor.

Será imprescindible por lo tanto depositar en secretaría:

- Copia impresa y encuadernada de la memoria junto con los Anexos en formato papel
- El trabajo en formato electrónico (usb o similar): Documento único en formato PDF de la memoria con nombre “ProyectoFinalCiclo_NombreApellidos” (incluyendo los anexos) y la documentación adicional relativa al proyecto (código fuente)

El alumno llevará también una copia de la portada, la cual le sellarán en secretaría y se podrá quedar como registro de que ha realizado el depósito.

La documentación anterior será OBLIGATORIA y previa a la defensa pública del trabajo.

Para los alumnos de ERASMUS y los que estén fuera de la comunidad autónoma, lo podrán enviar dentro del plazo previsto en formato digital, al correo de la secretaría del CPIFP LOS ENLACES, comprometiéndose a entregarlo al tutor antes de la exposición en el formato estipulado

La entrega se realizará normalmente una semana antes de la exposición y defensa de los proyectos.

Exposición y defensa de los proyectos

La evaluación del módulo profesional de proyecto requerirá la presentación y defensa pública por parte del alumno del proyecto realizado, ante una representación del equipo docente del ciclo formativo, constituida por al menos tres miembros del mismo, que a tales efectos actuará como tribunal. El alumno realizará una breve presentación de su proyecto, para lo cual podrá utilizar cualquier aplicación de creación de presentaciones, o sistemas o equipos de apoyo disponibles en el centro, previo aviso, u otros que considere necesarios y pueda aportar el propio interesado.

Deberá defender el proyecto justificando las decisiones adoptadas, y responder adecuadamente a las preguntas formuladas por el tribunal.

Las fechas de defensa se publicarán en el hall del centro.

Los alumnos que depositen el proyecto y no se presenten a su defensa serán calificados con NO EVALUADO (NE) en el acta.

De la misma forma, en el caso de que el tribunal detectara plagios o copias de trabajos previamente publicados, los alumnos serán calificados con NO EVALUADO (NE) en el acta.

- Tiempo máximo para la exposición: 20 minutos.
- Aconsejable 12 transparencias (legibles), para reproducir con los medios utilizados en las aulas y de los que el alumnado es conocedor.
- Después de la exposición por parte del alumno, el equipo docente dispondrá de 10 minutos para hacer las preguntas que considere oportunas.
- Cada 30 minutos entra una persona.
- La sesión es pública.