**GUIÓN PARA DESARROLLAR EL MÓDULO DE PROYECTO**

**Curso 2017 – 2018**

Contenido

[1. Análisis Previo 2](#_Toc502066745)

[1.1. Descripción del producto o proyecto a realizar. 2](#_Toc502066746)

[1.2. Descripción de la Empresa 2](#_Toc502066747)

[1.3. Planificación del proyecto 2](#_Toc502066748)

[2. Modelado: Análisis y diseño 2](#_Toc502066749)

[2.1 Diagrama de Casos de Uso 2](#_Toc502066750)

[2.2 Diagrama de ER y Relacional y scripts 3](#_Toc502066751)

[2.3 Diagrama de Clases 3](#_Toc502066752)

[2.4 Diseño de Interfaces 3](#_Toc502066753)

[2.5 Implementación y pruebas 4](#_Toc502066754)

[3. Entrega y exposición 4](#_Toc502066755)

[4. Formato de entrega del proyecto 4](#_Toc502066756)

[5. Evaluación del proyecto 4](#_Toc502066757)

## Análisis Previo

### Descripción del producto o proyecto a realizar.

**Nombre del *proyecto/producto***, **descripción general** y decir **qué hará el producto**. Si vais a desarrollar una nueva funcionalidad para un producto ya existente o lo hacéis por completo vosotros...

Establecer una lista de funcionalidades qué vais a desarrollar (todas o parte) de manera textual, y un plan con las prioridades del resto del desarrollo.

Es el *cliente* o la empresa quien os define este apartado (hablar y comentar este punto con los tutores en empresa o vuestros clientes/jefes en la empresa dónde estéis. Si se trata de un proyecto desarrollado fuera de la empresa de FCTs, simular que un cliente ficticio os requiere las funcionalidades listadas y el producto descrito.

El cliente no tiene por qué ser técnico, puede expresar que desea algo o tiene un problema y vosotros sugerir una solución.

Además de Requisitos Funcionales (los más importantes) también deben definirse no funcionales. Por ejemplo: seguridad, accesibilidad, portabilidad, flexibilidad para poder ser ampliado en el futuro, o escalabilidad.

### Descripción de la Empresa

1. Descripción general de la empresa
2. Descripción del negocio y mercado de la empresa.
3. Pequeño estudio de mercado, otras aplicaciones parecidas a la que vosotros deseáis. Pueden ponerse pequeñas capturas y links a las mismas con un pequeño párrafo de descripción. Este apartado prohibido más de una página.
4. Justificar el desarrollo del proyecto o producto para la empresa. Oportunidad de negocio o beneficio. Diferencias con las aplicaciones del punto 3 anterior. *Objetivo* del proyecto o producto

Los puntos uno y dos van juntos o están relacionados, nos dan el contexto donde vais a desarrollar la aplicación o producto.

Los puntos 3 y 4 también están relacionados, el objetivo es la justificación del producto o proyecto.

### Planificación del proyecto

Dividid el proyecto en fases especificando contenido y plazos de ejecución, definiendo tanto recursos materiales como personales para realizar el proyecto.

* Identificar las fases del proyecto especificando su contenido y plazos de ejecución
* Prever los recursos materiales y personales necesarios para realizar el proyecto.
* Secuenciar las tareas en función de las necesidades de implementación

Dar una **descripción textual de las fases**, de los recursos y proporcionar un **diagrama de Gantt de las fases**.

## Modelado: Análisis y diseño

### Diagrama de Casos de Uso

Debéis reflejar Actores externos (usuario, servidores de empresas externas,..) Estos son los que usan la funcionalidad principal (pasan información al software y obtienen un resultado o nueva información relevante) o nos dan servicio (nuestro software les invoca pasándoles información y obtienen unos resultados).

Los casos de uso, acordaros que son procesos o funciones de programación (les das unos parámetros y devuelven un resultado) es decir procesos que tienen una entrada y producen una salida representados por flechas normales o sin etiquetas.

Hay que poner las relaciones entre procesos:

* *include* cuando un proceso invoca o usa otro siempre u obligatoriamente,
* *extends* cuando un proceso extiende a otro, es como una invocación condicional (que no se da siempre o en todos los casos). El proceso que extiende es el proceso invocado (condicionalmente). La flecha se dibuja del proceso que extiende al proceso principal o proceso que invoca a la extensión. Por ejemplo, un proceso de Login, en el caso de que la persona no esté registrada, invocará al proceso de Registro. Por tanto el caso de uso de Registro *extiende* al de Login.
* *generalize* para definir un proceso o función genérica y funciones concretas o subtipos de funciones. Por ejemplo un proceso de Login, puede tener los procesos hijos o procesos concretos de: Login Por password, login con Ceritificado electrónico y Login por huella dactilar, etc.)
* *flechas sin etiqueta:* son de transmisión de datos. Se utilizan sobre todo entre los actores y los procesos, pero en ocasiones también se usan para conectar procesos entre sí (se divide algún proceso en subprocesos). Se debe indicar el dato que viaja por la flecha en una etiqueta asociada a la flecha.

Se deben proporcionar las **especificaciones (o contratos) de los Casos de Uso más relevantes**, los que directamente puede hacer el usuario, y aquellos que más se reutilicen por parte de otros procesos.

<http://www.umldesigner.org/ref-doc/define-the-application.html>

### Diagrama de ER, Relacional y scripts

El diagrama ER es un diagrama muy importante que define los datos con los que va a trabajar vuestra aplicación (clientes y empleados, libros y autores, etc.)

El relacional solo debéis hacerlo si la implementación de la BBDD es vuestra, si no existe previamente. En caso de estar usando alguna de la empresa no es necesario.

Si usáis BBDD, en los scripts a presentar, debéis poner las consultas (y modificaciones si las hay) que se hagan. Si creáis la BBDD también debéis poner los scripts de creación de tablas e inserción de datos iniciales.

### Diagrama de Clases

* *1ª fase*: Clases de Datos: con atributos y métodos. Relaciones (asociaciones, agregaciones, generalización). Muy parecido al diagrama ER del apartado anterior.
* *2ª fase*: Clases de Diseño: Se añaden las clases necesarias para dar soporte a la funcionalidad. Ejemplos: La clase Cliente, la clase Servidor, posibles subclases de servidores o clientes de distintos tipos. O una clase Empresa). En estas clases debéis poner las funciones identificadas en los casos de uso (lo que debe hacer o permitir la aplicación) y las relaciones con otras clases, principalmente de datos, de la primera fase, para poder realizar el servicio).
* *3ª Fase*: Clases de implementación: Librerías o funciones de utilidad. Ejemplos: para acceso a ficheros/Bases de datos o matemáticas. Estas clases se pueden omitir e indicar en el apartado de implementación (ver más abajo).

Las operaciones deben hacer referencia a la funcionalidad definida en los casos de uso. **Todos los casos de uso principales deben estar reflejados en funciones de alguna clase**. El objetivo de las operaciones es cubrir o implementar la funcionalidad que nos requiere el cliente.

### Diseño de Interfaces

Debe ser un dibujo esquemático de cada pantalla. No capturas de la app final. Elaborar todas pantallas, con la secuencia de navegación, botones y funcionalidades que se lanzan desde cada botón.

Posibles vueltas atrás en las pantallas, listados y paginación (5-10 elementos de la lista por página), acceso al detalle de cada producto. Pantallas para alta y modificación de elementos, etc. Pantallas para pago/compra, pantallas para identificación/acceso...(se puede utilizar la herramienta que se desee).

### Implementación y pruebas

Descripción de las clases extras añadidas durante la implementación. Las librerías utilizadas (por ejemplo: librerías para la gestión de ficheros, librerías matemáticas...), y otras herramientas utilizadas, links a tutoriales y libros.

Comentarios sobre alguna cosa curiosa o problemas importantes encontrados.

Capturas de pantalla para demostrar las pruebas realizadas de la aplicación (que demuestren que la funcionalidad establecida al principio del desarrollo y en el diagrama de casos de uso se ha completado). Probar todas las opciones: login correcto, login incorrecto, listado de productos sin datos, con una o varias páginas en el listado de productos…

Se deberá desarrollar el proyecto analizado anteriormente. En caso de ser un proyecto demasiado extenso no será obligatorio el desarrollo completo de él, sino una parte representativa de su funcionalidad (se puede dejar algún botón sin funcionamiento). A pesar de no ser obligatorio (en caso de proyectos grandes), se tendrá en cuenta el desarrollo completo a la hora de la evaluación.

## Entrega y exposición

Las fechas fijadas para la entrega y exposición de los proyectos son las siguientes:

* Periodo de entrega del proyecto: Primera semana de junio para los alumnos que realizan las FCT en el último trimestre del curso y primera semana de diciembre para los alumnos que realizan las FCT en período extraordinario.
* Exposiciones orales del proyecto: Segunda semana de junio para los alumnos que realizan las FCT en el último trimestre del curso y segunda semana de diciembre para los alumnos que realizan las FCT en período extraordinario.

Una vez entregado el proyecto, se asignará un día y hora para su exposición ante el tribunal. La exposición tendrá una duración máxima de 30 minutos (15/20 minutos exposición del alumno 10/15 minutos preguntas del tribunal) y los alumnos podrán hacer uso del material que crean conveniente para su apoyo a la exposición.

## Formato de entrega del proyecto

Se debe entregar la documentación encuadernada en papel y un CD/DVD con vuestro nombre, curso y el título del proyecto. Dicho disco debe contener los ficheros de los apartados solicitados. La entrega de las fuentes del proyecto no es necesaria, si bien se podrá pedir que se enseñe lo que el tribunal solicite, e incluso podría convocarse al alumno otro día para pedirle la muestra de lo que el tribunal considere necesario para la evaluación del proyecto.

## Evaluación del proyecto

Según la rúbrica que se facilite, tanto para el proyecto, su documentación y su exposición.