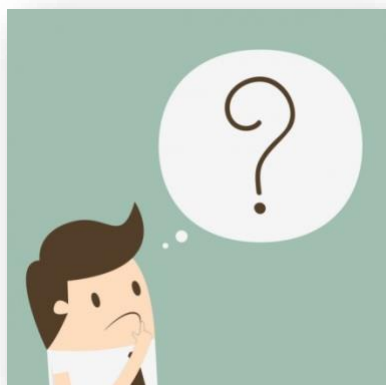


**PROYECTOS TERCER TRIMESTRE****1º DAM****Consideraciones generales para todos los proyectos**

Cuando te matriculaste en este módulo ¿tenías en mente alguna aplicación que te gustaría desarrollar? Seguramente alguna vez a lo largo de las unidades de este módulo habrás pensado en alguna aplicación práctica de lo que estabas aprendiendo para uso personal, o para algún conocido o amigo.



Sabemos por experiencia que **los mejores proyectos son siempre los que más motivan a las personas** que los desarrollan. Eso es aplicable a todos los órdenes de la vida, pero especialmente a la programación. Para que una aplicación sea buena, debe resultar útil, y no hay mejor medida de la utilidad de una aplicación que someterla a los requerimientos de la realidad.

¿Qué queremos decir con esto?

Que a continuación, en los apartados siguientes te vamos a hacer algunas sugerencias sobre posibles **proyectos** que puedes realizar, con el objeto de que elijas uno. Pero esos proyectos, al no ser reales, no pueden ser detallados, ni tienen en consideración detalles que la práctica demostraría imprescindibles en muchos casos.

Por el contrario, si tú mismo te propones desarrollar una aplicación con el objetivo concreto de usarla, sí que vas a tener mucho más claro lo que necesitas que esa aplicación te haga, **y así vas a disponer de unos requerimientos de mucha más calidad que te van a ayudar en la realización del trabajo. Puedes realizar la toma de requisitos con personas cercanos que “sepan” de esa actividad**, aunque no tendremos el tiempo suficiente para que ese proyecto sea totalmente operativo en la realidad.

Y si además de cumplir con la tarea de este módulo tu trabajo resulta útil para ti, o para algún conocido tuyo, mucho mejor.

- Si es uno propio que tú mismo te has propuesto, deberás **redactar un documento** indicando en qué consiste, y presentárselo a los profesores para que te den el visto bueno, o te indiquen las mejoras que debes tener en cuenta para que cumpla con los requisitos generales que debe cumplir cualquier proyecto.

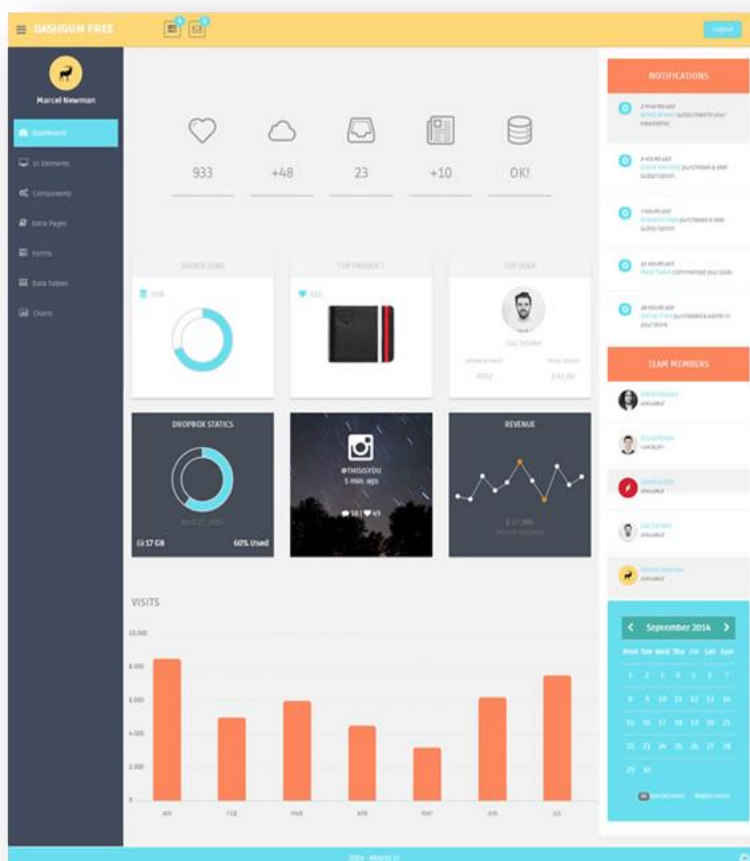
Pero claro, entre la multitud de ideas que se os pueden ocurrir, no todas van a tener la misma complejidad, ni la misma dificultad, ni el mismo tamaño. Y como queramos o no, tenemos que evaluarlos con unos criterios lo más objetivos y equitativos posible, es necesario que fijemos las características básicas que deberá incluir cualquier proyecto que realicéis, tanto si lo proponéis vosotros mismos como si cogéis uno de los enunciados que mostramos como propuestas en esta unidad.

## Requisitos mínimos a cumplir por todos los proyectos

¿Pero qué debe hacer tu aplicación, y qué debes tener en cuenta para conseguir una buena calificación de tu proyecto?

**El proyecto consiste básicamente en el desarrollo de una aplicación web por parte de cada uno de los alumnos (de forma individual)** que debe incluir los siguientes puntos:

- Utilización de una base de datos con JPA e Hibernate.
- Diseño de un interfaz gráfico (GUI) apropiado, usando HTML5, CSS3, JS, jQuery y Bootstrap.
- Implementación de la aplicación web usando Spring MVC, Spring Boot y Thymeleaf.



- Mediante el interfaz gráfico debe ser posible realizar las siguientes operaciones en la base de datos:
  - Insertar** datos.
  - Buscar** datos.
  - Modificar** datos.
  - Borrar** datos.
  - Consultar, mostrar y filtrar** datos.
- Con el objetivo de que el aprendizaje se lo más significativo posible, será obligatorio, que los servicios no sean un mero envoltorio de los repositorios (CRUD), por lo que, se debe desarrollar la lógica de negocio definida en los casos de uso.
- Todos los proyectos tendrán la posibilidad de gestionar usuarios, preferentemente con dos tipos de perfiles: usuario normal y administrador. Además, por fuerza, para acceder a la aplicación cualquier usuario deberá *loguearse* en el formulario "Login".
- Deberán **capturarse y tratarse** adecuadamente cuantas excepciones sean necesarias para que el programa no aborte. Cualquier circunstancia de error deberá ser tratada convenientemente.
- Se pide, de cara a mejorar la nota, que se **investiguen e incorporen elementos nuevos, de los disponibles en la API de Java o librerías de terceros**, aunque no hayan sido vistos en el curso, o no se hayan visto con demasiada profundidad como, por ejemplo, la posibilidad de enviar emails, generación de pdf, uso de estadísticas, etc.
- El código debe estar completamente **documentado y comentado**.
- Debe hacerse un uso adecuado de los conceptos de Programación Orientada a Objetos, al gestionar métodos, interfaces, herencia, polimorfismo, colecciones, tipos de clases, etc. Se deben usar patrones de diseño de aquellos estudiados en clase (MVC, DAO...)
- Para poder comprobar el buen funcionamiento de la aplicación, se deberá poder arrancar con **un conjunto amplio y variado de datos de prueba en la base de datos**, almacenados de sesiones anteriores. Dicho conjunto deberá incluir los tipos necesarios para probar el mayor número posible de casos y opciones disponibles. Se trata de incluir todos aquellos datos de prueba que se considere oportuno para que permitan resaltar la calidad de la aplicación desarrollada.
- La aplicación se realizará usando el IDE Spring Tool Suite, haciendo uso de Maven para la gestión de las dependencias con otras librerías externas. **No se aceptará ningún proyecto que incluya alguna de las librerías "habituales" vinculadas directamente como un .jar.**
- El sistema debe ser robusto frente a fallos y no "abortar" por circunstancias imprevistas.
- La aplicación estará **escrita en lenguaje Java**. No deben utilizarse clases o métodos obsoletos ("deprecated").
- Los programas fuente deberán ser fáciles de mantener por cualquier programador experto en Java. Deberán estar bien estructurados y contendrán los comentarios necesarios para un fácil seguimiento del código.
- En el diseño de la interfaz gráfica, se debe diseñar tanto la parte dinámica de la aplicación, como la **estática**, ofreciendo información básica del sitio ("quiénes somos", "contacto", "qué hacemos", etc.). A la hora de implementar el sitio web, deberán tenerse en cuenta criterios para mejorar el **posicionamiento SEO**.
- Se debe realizar algún tipo de operación básica en el lado del cliente, mediante el uso de librerías Javascript como **JQuery** u otras. Estas operaciones comprenden, al menos, la validación de datos y la manipulación del documento (DOM).



- Se pide también utilizar **otras librerías Javascript** de terceros para mejorar la apariencia y funcionalidad del sitio. El uso de estas librerías forma parte de las tareas de investigación del proyecto. Algunos ejemplos de estas librerías son Moments (para la gestión de fechas) o Cleave (para el formateo de los campos de formularios). Se facilitará un listado más amplio de sugerencias.
- Como **Plan de Pruebas**, se deberán documentar e implementar 4 casos de pruebas unitarias diferentes. La implementación de las pruebas se realizará con **JUnit**. Los casos deberán ser de operaciones diferentes y, al menos, uno tiene que probar una operación de la lógica de negocio diferente a las operaciones CRUD.

### Documentación a entregar junto al proyecto

- **La carpeta del proyecto.** Se incluirá la carpeta completa del proyecto en *Spring Tool Suite*, que contendrá:
  - Todos los archivos con **los programas fuente**.
  - Fichero de creación de la base de datos, si procede.
  - Ficheros estáticos HTML, CSS, JS, javadoc...
- **Documentación:**
  - Diagrama(s) de casos de uso
  - Diagrama de clases.
  - Plan de pruebas.



### Consideraciones

*Sugerencia: Buscad en Internet aplicaciones similares que ofrezcan los mismos o parecidos servicios a los que quieres introducir en tu aplicación, para ver cómo están diseñadas y obtener ideas acerca del funcionamiento de las aplicaciones y de las interfaces de usuario.*

### Listado de Proyectos

1. Gestión de una tienda de ordenadores.
2. Gestión de una bolsa de trabajo para alumnos de FP titulados en los Salesianos de Triana.
3. Gestión de una cafetería.
4. Gestión de una librería.
5. Gestión de un concesionario de coches.
6. Gestión de un club deportivo.
7. Gestión de tienda de productos de cocina.
8. Gestión de reservas de un hotel.
9. Gestión de una hermandad.
10. Gestión de una empresa de transportes.
11. Gestión de una banda de música.
12. Gestión de productos de un Gimnasio.
13. Gestión de una empresa que realiza Eventos deportivos.
14. Gestión de una tienda de videojuegos.
15. Gestión de venta de entradas de un Cine.
16. Gestión de asociación de juegos de mesa.
17. Gestión de las obras de un museo.
18. Gestión de una tienda de vinos y chacina.
19. Gestión tienda de venta de productos para mascotas.
20. Gestión de los productos vendidos en un bar.
21. Gestión de un centro de belleza.
22. Gestión tienda de decoración.
23. Gestión de una peña de fútbol.
24. Gestión tienda de cubos de Rubik.
25. Gestión de taller de reparaciones de móviles.
26. Gestión de parque infantil.
27. Gestión de una empresa de gestión de residuos.
28. Gestión tienda de tatuajes.
29. Gestión tienda de ropa.
30. Gestión clínica dentista.
31. Tienda online de productos de segunda mano.
32. Gestión de una granja de animales.
33. Gestión de tendencias políticas.
34. Gestión de voluntariado ONG's.
35. Gestión de hostings.

## Evaluación del Proyecto

Lo primero que tenemos que decir es que la valoración se hará de una forma global. La calidad del proyecto y de la aplicación se tiene que valorar en su conjunto. No obstante, como este proyecto no tiene una finalidad comercial, sino que su propósito es valorar la consecución de los objetivos de estos módulos profesionales, los puntos en los que nos vamos a fijar pueden ser ligeramente distintos de los que te exigiría el cliente al que le intentas vender una aplicación.



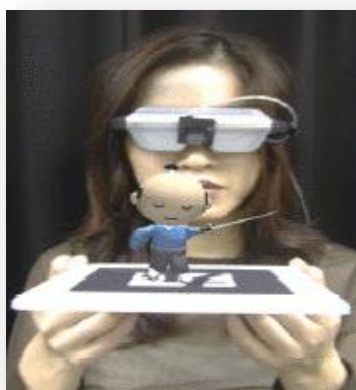
Los perfiles de usuario y administrador deben cumplir:

**Usuario: usuario → Contraseña: 1234**

**Administrador: admin → Contraseña: admin**

Otro aspecto importante, es dar una lista de posibles mejoras para vuestra aplicación en un futuro, que han tenido que quedar aparcadas por falta de tiempo.

## Aplicación desarrollada



Hay muchos apartados que se valorarán dentro de la aplicación. Te los enumeramos a continuación, con una indicación de su valoración global dentro de la nota del proyecto. Debes tener en cuenta que la mayoría de esos apartados están íntimamente relacionados entre sí, por lo que no es fácil separar a veces lo que es un fallo de la funcionalidad de lo que es un fallo del diseño del interfaz gráfico, o lo que es un fallo del acceso a los datos de la base de datos.

Se os facilitará una rúbrica con las distintas puntuaciones de cada módulo.

Cada módulo tendrá su propia nota según la rúbrica que se entregará por cada profesor.

Además, a título informativo, se dará una calificación global de todo el proyecto. Para dicha nota el peso de cada módulo será el siguiente:



Entornos de Desarrollo: 20 %

Lenguajes de Marcas y SGI: 20 %

Programación: 20 %

Base de datos: 40 %

