

PROYECTO FINAL TERCER TRIMESTRE

1º DAM

1. Consideraciones generales para todos los proyectos



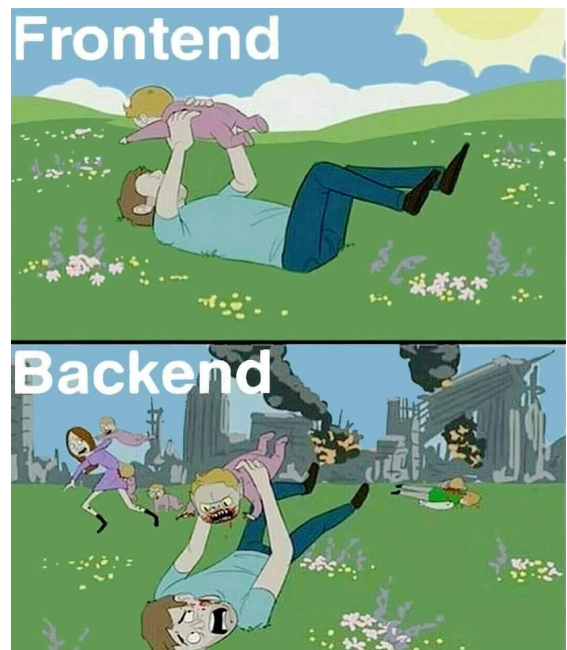
Como ya hemos hablado en clase, este año el proyecto final de curso tendrá características especiales. Se gestionarán menos entidades y tendrá menos funcionalidades. Sería “una parte” de un proyecto más completo.

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 elijas algo que te guste o de lo que sepas, para poder tener más claras las especificaciones o características del programa.

Os facilitaremos una lista de posibles temas, pero si quieres elegir uno que no esté en la lista, debes **redactar un documento** indicando en qué consiste, y presentarlo 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.



2. 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 es individual y debe consistir básicamente en el desarrollo de una aplicación web 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, Bootstrap y JS.
- Implementación de la aplicación web usando Spring MVC, Spring Boot y Thymeleaf.



· Mediante la interfaz gráfica 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.

- La aplicación contará al menos con dos entidades y una asociación entre ellas, pudiéndose realizar al menos todas las operaciones CRUD (insertar, actualizar, borrar, buscar) de ambas.
- Deberán **capturarse y tratarse** adecuadamente cuantas excepciones sean necesarias para que el programa no aborte. Cualquier circunstancia de error deberá ser tratada convenientemente.
- Se puede investigar e incorporar, de cara a mejorar la nota, algún **elemento nuevo, de los disponibles en el 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.
- 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...).
- Deberán ser usadas características avanzadas de Java, como pueden ser expresiones Lambda y Streams.
- 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 **la parte dinámica** de la aplicación, integrada en la parte **estática**, por lo que debe realizarse un diseño adaptativo que tenga cabecera, menú, contenido lateral y pie de página. En la versión de escritorio debe estar maquetado utilizando, al menos, 2 columnas. La parte estática debe tener al menos 3 páginas con información variada sobre la empresa. 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 Javascript. Estas operaciones comprenden, al menos, la validación de datos de formularios 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 al menos 4 casos de pruebas diferentes.

3. Documentación a entregar junto al proyecto

La entrega se realizará a través de Git. Debes invitar a los 3 profesores (Rafa, Luismi y Ángel) a tu repositorio. Se concretará en clase cómo hacerlo.

Documentación:

- Plantilla de Historias de Usuario
- Diagrama de clases.
- Plan de pruebas.

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.

**Cuando empiezas a programar
en un nuevo lenguaje
sin leer la documentación**



4. Evaluación del Proyecto



En primer lugar, hay que decir 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.

Se facilitarán las rúbricas utilizadas en cada módulo.

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.

5. 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 los distintos criterios y 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: 30 %

Base de datos: 30 %

