

# PROYECTO EN GRUPO

DISEÑO DE SISTEMAS SOFTWARE - 2017/2018

## Objetivos de la práctica

- Diseño de una aplicación mediante el uso de los conceptos y patrones de diseño estudiados en las clases teóricas.
- Desarrollo y entrega en los plazos previstos de una aplicación funcional a partir del diseño realizado.
- Trabajo en equipo, usando herramientas de gestión de proyectos y de código fuente.
- Presentación del trabajo realizado mediante una demostración de las funcionalidades implementadas y justificando adecuadamente, en caso necesario, las partes del proyecto que no se han podido desarrollar.

## Metodología de trabajo

El proyecto se realizará en equipos de 4 personas, cuyos miembros pertenecerán al mismo turno de prácticas. Solamente en casos excepcionales y debido al número de alumnos en cada turno se permitirá la formación de grupos de 3 o 5 personas.

Las primeras 5 semanas se dedicarán al aprendizaje de las herramientas y la especificación inicial del proyecto. Tras la primera entrega se realizará el desarrollo del proyecto en 2 iteraciones de unas 5 semanas de duración cada una.

Al comienzo de cada iteración el equipo seleccionará los requisitos funcionales que se desarrollarán a lo largo de la iteración, y cuyo resultado será un software funcional que se mostrará mediante una presentación al final de cada iteración.

Cada equipo mantendrá un repositorio de código privado para el control de versiones.

## Descripción del proyecto a implementar

La temática del proyecto será propuesta por cada grupo teniendo en cuenta las siguientes restricciones:

- El proyecto se implementará como una aplicación web en el lenguaje PHP, usando el framework Laravel.
- La aplicación tendrá una arquitectura en capas, al final del desarrollo deberá tener al menos las siguientes capas:
  - Capa de lógica de negocio y acceso a datos.

- Capa de servicios, proporcionará funcionalidades de lógica de negocio que involucren a más de una entidad de negocio.
- Capa de presentación usando el patrón MVC.
- Todas las funcionalidades de lógica de negocio implementadas deberán ser accesibles a través de la capa de presentación.

Además habrá que tener en cuenta las siguientes restricciones tecnológicas:

- El acceso a datos estará integrado en las clases de lógica de negocio usando el framework Eloquent ORM incluido en Laravel.
- Las vistas serán páginas HTML generadas a partir de plantillas Blade. El uso de otras tecnologías de presentación como CSS o JavaScript se valorará aunque es opcional.
- La base de datos usará el motor SQLite.

## Evaluación

Para la evaluación del proyecto se tendrá en cuenta tanto el diseño como la implementación. La implementación deberá corresponderse siempre al diseño entregado, y cuando no sea posible implementar un diseño se deberá modificar los diagramas para reflejar los cambios realizados durante la implementación. También se deberán respetar las recomendaciones de uso del framework Laravel y las buenas prácticas recomendadas por el profesorado.

El proyecto se evaluará en tres partes con el siguiente peso:

Semana	Contenidos	% nota de prácticas
26 feb - 2 mar	<ul style="list-style-type: none"> <li>● Descripción del proyecto</li> <li>● Diagramas de diseño (v1)</li> <li>● Implementación del diagrama</li> </ul>	10%
16 abr - 20 abr	<ul style="list-style-type: none"> <li>● Diagramas de diseño (v2)</li> <li>● Aplicación funcional sin autenticación de usuarios</li> </ul>	30%
21 may - 25 may	<ul style="list-style-type: none"> <li>● Diagramas de diseño (v3)</li> <li>● Aplicación funcional</li> </ul>	60%

La calificación de la práctica será igual para todos los componentes del equipo, salvo que se detecte alguna anomalía en el funcionamiento dentro del grupo. En ese caso, el profesorado podrá calificar de forma individual la participación de cada uno de los miembros del equipo.

**En caso de plagio, la calificación de la práctica será 0 para todas las personas implicadas.**

# PRIMERA ENTREGA

**Fecha límite de entrega: 2/3/2018**

## Preparación: repositorio en GitHub

Para el desarrollo del proyecto debéis crear un **repositorio privado en GitHub** compartido con todos los miembros del equipo. Debéis añadir también al profesor del turno de prácticas como colaborador con el nombre de usuario que os proporcionará.

## Diagrama de clases

En esta primera entrega debéis realizar un diseño inicial de vuestro proyecto mediante un diagrama de clases con las siguientes características:

- El diagrama tendrá **3 clases** relacionadas entre sí.
- Se debe indicar el tipo de datos de los atributos.
- Las asociaciones deben especificar la multiplicidad y el nombre de las propiedades.
- De momento no es necesario añadir métodos a las clases.

## Implementación

Una vez realizado el diseño debéis implementar los modelos en vuestro proyecto de Laravel. Debéis implementar también todas las migraciones y seeders necesarios para inicializar la base de datos, y al menos un caso de prueba para probar cada relación.

El código del proyecto debe estar en el repositorio de GitHub, excluyendo el archivo de base de datos `database/database.sqlite`.

## Entrega y evaluación

Para la entrega del proyecto debéis descargarlo en formato zip desde el repositorio en GitHub y entregarlo a través de la tarea disponible en Moodle, incluyendo también el diagrama en formato PDF.

En el momento de la entrega debéis asignar la etiqueta “entrega1” al commit en GitHub correspondiente al código entregado.

La primera entrega se evaluará de acuerdo a los siguientes criterios:

Criterio	Puntuación (sobre 10)
Corrección del diagrama de clases	3
La implementación se corresponde al diagrama	3
Se han implementado Seeders con datos de ejemplo	2
Se han implementado los casos de prueba	2

# SEGUNDA ENTREGA

**Fecha límite de entrega: 20/4/2017**

## Descripción de la entrega

En esta segunda entrega comenzaréis la implementación de la aplicación web para vuestro proyecto. Para ello deberéis seleccionar las funcionalidades básicas necesarias para poner en marcha el proyecto, y de las que dependerán el resto de funcionalidades que añadiréis más adelante en la tercera entrega.

## Diagrama de clases

Para esta entrega debéis identificar y diseñar **5 clases** que agrupen las funcionalidades básicas del proyecto. En el diagrama de clases debéis especificar claramente sus **atributos, métodos y relaciones**.

## Mockups

Antes de la implementación de las vistas se debe realizar un diseño de las pantallas mediante mockups que reflejen claramente el flujo de trabajo de la aplicación.

## Implementación

En esta entrega se implementarán las clases del modelo de dominio correspondientes al diagrama de clases, incluyendo la implementación de los métodos de lógica de negocio. También se implementarán las vistas y controladores necesarios para acceder a estas funcionalidades de lógica de negocio. La implementación se hará de forma progresiva a medida que se expliquen los conceptos necesarios en las sesiones presenciales de prácticas. El resultado al final de esta entrega será una aplicación web con los formularios que permitirán realizar listados, consultas, crear, modificar y borrar para las clases implementadas.

### **Requisitos mínimos a implementar:**

- Mostrar listados para todos los objetos del modelo de dominio.
- Borrar cualquier objeto en la base de datos.
- Crear y modificar objetos de 3 tipos distintos (para los demás se puede poblar la base de datos con Seeders).

### **Requisitos opcionales:**

- Paginación de listados.
- Ordenar listados por algún campo.
- Realizar búsquedas de una de las clases con 2 criterios de búsqueda distintos.

## Autenticación y usuarios de la aplicación

**En esta entrega no se debe implementar la autenticación de usuarios.**

En caso de incluir una clase que represente a los usuarios, se debe usar la clase `User` que incluyen los proyectos de Laravel por defecto.

En algunos casos puede ser necesario asociar los objetos creados al usuario que se ha identificado en el sistema. En la próxima entrega veremos cómo recuperar éste usuario, de momento podéis simular esta funcionalidad creando un método estático en la clase `User` llamado `currentUser()` que devuelva una instancia inicializada de esta clase.

## Metodología

La carga de trabajo se debe distribuir de forma uniforme entre los componentes del equipo, y se llevará un seguimiento de la actividad del repositorio en GitHub.

Para la planificación de tareas y coordinación de los miembros del equipo se deberá crear un tablero en la aplicación web Trello (<https://trello.com>) compartido entre todos los miembros y el profesor del turno de prácticas. El tablero tendrá tres columnas: Pendiente, En progreso y Terminado. En este tablero se irán registrando las tareas planificadas y se actualizará reflejando el progreso y dedicación de cada miembro del equipo.

## Presentación en clase

**En la última sesión de prácticas de esta iteración se hará una presentación del proyecto con una duración máxima de 15 minutos, con los siguientes contenidos:**

- Breve presentación del objetivo del proyecto.
- Demostración del proyecto funcional.
- Conclusiones: aspectos positivos y negativos de la metodología y las tecnologías empleadas, dificultades encontradas, posibles mejoras, etc.

## Entrega y evaluación

Para la entrega del proyecto debéis descargarlo en formato zip desde el repositorio en GitHub y entregarlo a través de la tarea disponible en Moodle, incluyendo también los mockups, el diagrama de clases y los materiales de la presentación en formato PDF.

En el momento de la entrega debéis asignar la etiqueta “entrega2” al commit en GitHub correspondiente al código entregado.

La segunda entrega se evaluará de acuerdo a los siguientes criterios:

<b>Criterio</b>	<b>Puntuación (sobre 10)</b>
Diseño y planificación (UML, mockups)	3
Implementación de los requisitos mínimos	4
Implementación de los requisitos opcionales	3

En esta entrega la implementación de casos de prueba no es evaluable, aunque es aconsejable para evitar errores durante el desarrollo.

# ENTREGA FINAL

**Fecha límite de entrega: 25/5/2017**

## Descripción de la entrega

En esta última entrega terminaréis el desarrollo de la aplicación, añadiendo funcionalidades adicionales que impliquen una lógica de negocio más allá del mantenimiento de tablas de la base de datos (métodos CRUD implementados en la segunda entrega).

## Diagrama de clases

Para esta entrega debéis añadir todo lo que sea necesario para la implementación de las nuevas funcionalidades. Se podrán modificar las clases y relaciones del diagrama de la anterior entrega si las nuevas funcionalidades lo requieren.

## Mockups

Todas las funcionalidades no contempladas en la entrega anterior deberán diseñarse previamente mediante mockups, prestando especial atención al flujo de trabajo de la aplicación.

## Implementación

Además de los requisitos mínimos exigidos en la segunda entrega, la implementación del proyecto debe incluir:

### Autenticación y registro de usuarios

- El sitio web tendrá dos partes:
  - **Parte pública** para usuarios invitados, con al menos 3 vistas: página principal, página de información del proyecto y página de contacto, además de las que permiten hacer el login y el registro de usuarios.
  - **Parte privada** para usuarios registrados. Todas las vistas en la parte privada se deben proteger con el middleware de autenticación.
- La parte privada podrá distinguir además entre dos tipos de usuarios:
  - **Administrador**: tendrá acceso a funcionalidades privilegiadas como configuración de la aplicación, gestión de usuarios, borrar y/o modificar datos que los usuarios normales no pueden, etc.
  - **Usuario normal**: tendrá acceso a las funcionalidades para las que esté diseñada la aplicación. Dependiendo de la aplicación se podrá distinguir entre distintos tipos de usuarios o niveles de permisos si es necesario.

## Nuevas funcionalidades y capa de servicios

Se deberán añadir funcionalidades que impliquen la manipulación de varios objetos del modelo de dominio. Estas funcionalidades deberán ubicarse en la capa de servicios, harán uso de los modelos y trabajarán en modo transaccional con la base de datos cuando la lógica de negocio lo requiera.

En caso de duda sobre las funcionalidades a implementar (cantidad y/o detalles sobre los requisitos), se deberá llegar a un acuerdo con el profesor del grupo de prácticas.

## Vistas y validación

Todos los formularios de la aplicación deben ser validados adecuadamente en los controladores, donde se añadirán además todas las validaciones necesarias para garantizar la seguridad y estabilidad de la aplicación.

Se deberá cuidar también la presentación de las vistas. Aunque el diseño web no es un aspecto muy relevante en este proyecto, todas las vistas deben tener al menos un aspecto cuidado y acorde a la finalidad de la aplicación.

## Metodología

La carga de trabajo se debe distribuir de forma uniforme entre los componentes del equipo, y se llevará un seguimiento de la actividad del repositorio en GitHub.

En esta entrega se seguirá usando el tablero de Trello para la gestión de tareas. Durante la primera semana se deberán identificar y registrar todas las tareas posibles, indicando una estimación temporal para valorar la carga de trabajo y seleccionar las funcionalidades que se pretenda implementar de acuerdo al tiempo disponible. Para esto se calcularán **6 horas semanales** (incluyendo las 2 horas presenciales) por cada miembro del equipo, durante un total de 5 semanas. Para un equipo de 4 personas esto hace un total de 120 horas.

## Presentación en clase

**En la última sesión de prácticas se hará una presentación del proyecto con una duración máxima de 15 minutos**, con los siguientes contenidos:

- Breve presentación del objetivo del proyecto.
- Demostración del proyecto funcional.
- Conclusiones: aspectos positivos y negativos de la metodología y las tecnologías empleadas, dificultades encontradas, posibles mejoras, etc.



## Entrega y evaluación

Para la entrega del proyecto debéis descargarlo en formato zip desde el repositorio en GitHub y entregarlo a través de la tarea disponible en Moodle, incluyendo también los mockups, el diagrama de clases y los materiales de la presentación en formato PDF.

En el momento de la entrega debéis asignar la etiqueta “entrega3” al commit en GitHub correspondiente al código entregado.

La entrega final se evaluará de acuerdo a los siguientes criterios:

Criterios	Puntuación (sobre 10)
<b>Diseño y metodología (obligatorio):</b> <ul style="list-style-type: none"> <li>• UML</li> <li>• Mockups</li> <li>• Planificación y seguimiento en Trello</li> <li>• Repositorio y actividad en GitHub</li> <li>• Presentación en clase</li> </ul>	3
<b>Requisitos mínimos (obligatorio):</b> <ul style="list-style-type: none"> <li>• Proyecto funcional</li> <li>• Parte pública y privada</li> <li>• Autenticación y registro de usuarios</li> <li>• Validación de formularios</li> </ul>	4
<b>Requisitos opcionales:</b> <ul style="list-style-type: none"> <li>• Roles y/o permisos</li> <li>• Diseño del interfaz</li> <li>• Interfaz responsive con Bootstrap</li> <li>• Capa de servicios</li> <li>• Otros (acordar con el profesor de prácticas)</li> </ul>	3