# PRÁCTICA 6 Y 7. NODEJS – MONGODB

## Objetivos

* Diseño y la manipulación de objetos
* Diseño y manipulación de bases de datos
* Detección y modelado de relaciones entre clases
* Aplicación de todos los conocimientos sobre programación aprendidos hasta el momento.
* Uso de modificadores de visibilidad
* Uso de excepciones.
* Uso de patrones de desarrollo.

El **objetivo** de esta práctica es realizar una **pequeña aplicación con interfaz web estándar**, utilizando **una base de datos no estructurada**. Para la realización de esta práctica es necesaria la utilización de buenas prácticas de Ingeniería del Software, entre las que se encuentran:

* Uso de un sistema de hospedaje de proyectos:
  + Github (<http://www.github.com>)
  + Bitbucket (<http://bitbucket.org>)
* Uso de un sistema de control de versiones:
  + Git (<http://git-scm.com/>)
  + Mercurial (<http://mercurial.selenic.com/>)
* Uso de una metodología de desarrollo basada en pruebas/comportamiento (**No se va a realizar)**
  + Test Driven Development (TDD)
  + Behaviour Driven Development (BDD)
* Sistema de Gestión de Proyectos **(No se va a realizar)**
  + SCRUM
  + Kanban
* Se pueden utilizar otras técnicas como pair programming, etc.

## Condiciones de la Entrega:

Se recuerda que los alumnos deberán defender en clase de prácticas el trabajo realizado, siendo necesario que asistan ambos miembros de cada pareja de prácticas, en caso de que uno de ellos no se presente, ese alumno no será evaluado y por tanto suspenso en la convocatoria ordinaria. La nota de cada miembro es diferente en función del trabajo realizado y la defensa que se haga.

Igualmente es necesario recordar que es necesario que la documentación entregada contenga, diseño y código fuente. En la fase de diseño deberá entregarse el diagrama de clases.

El funcionamiento de la práctica no garantiza el aprobado de esta, se valorarán otros aspectos como la robustez y eficiencia de esta, el correcto diseño y perfecta utilización de las técnicas y metodologías comentadas en clase de teoría, la documentación entregada y el cumplimiento de los requisitos definidos en el enunciado.

Toda esta información se subirá a través de Campus Virtual en un espacio habilitado para tal efecto, antes de la hora de la defensa para que quede constancia de la entrega. Posteriormente a esta fecha no se permitirá subir ningún otro documento. Toda la documentación deberá ir comprimida en un único archivo con extensión .rar o .zip. Preferiblemente la documentación debería ir en formato PDF y el código fuente en su correspondiente archivo .java.

También se recuerda que el hecho, tanto de copiar una práctica como dejarla para que otros la copien, es una falta muy grave que será sancionada.

Para facilitar la localización y corrección de los entregables, los ficheros de entrega se enviarán utilizando un nombre en el que conste el número de entregable y el número del equipo. Por ejemplo, para la práctica 6 del equipo GA1, el fichero deberá llamarse P6‐GA1.pdf.

Acompañando a cada entregable se enviará un informe en el que conste el nombre de cada miembro del equipo y el porcentaje de su participación y realización para dicho entregable.

# PARTE 1 - ESI APP STORE

En la Escuela Superior de Informática se pretender implantar una aplicación web para la gestión del software utilizado en las prácticas de laboratorio de las distintas asignaturas.

Dicha aplicación debe cumplir los siguientes requisitos:

* Accesible desde cualquier navegador web
* La aplicación debe permitir la gestión de las asignaturas del plan de estudios
* Debe haber un módulo de gestión de usuarios, en el cual se permita el acceso autenticado de los usuarios y determine el perfil o rol de cada usuario:
  + **Administrador**: tiene la capacidad de crear nuevos usuarios, modificar usuarios, eliminar usuarios, asignar roles, crear asignaturas, y asignar alumnos y profesores a las asignaturas
  + **Profesor**: puede modificar la información de sus asignaturas y añadir y editar las instrucciones de instalación de las aplicaciones
  + **Alumno**: puede consultar la información acerca del software de laboratorio de cada una de las asignaturas en la que está matriculado.
* Las asignaturas deben:
  + Poder tener varios profesores responsables de ellas
  + Tener varios alumnos matriculados
  + Proporcionar información acerca del proceso de instalación de las aplicaciones
  + Soportar enlaces externos (URLs)

Se pide implementar un sistema web que resuelva el problema propuesto.

# PARTE 2 - OPERACIONES DE MANTENIMIENTO

Tras la puesta en explotación de la ESI App Store, los clientes han encontrado ciertos problemas con el desarrollo de la acción y nos han pedido realizar diversas modificaciones.

**Mantenimiento Correctivo**

El mayor problema que se ha encontrado es que el proceso de incorporación de alumnos, profesores y asignaturas es muy tedioso, para solucionar este problema, se ha propuesto la siguiente modificación: ***Se desea permitir la incorporación de dichos datos mediante un fichero de texto plano en formato CSV.***

La aplicación no soporta diversos planes de estudio (grado, máster y doctorado) y ***Se pide incorporar la gestión de planes de estudio***.

**[OPCIONAL]** La aplicación no permite la carga y descarga de archivos, como por ejemplo, el ejecutable del software necesario o el archivo de licencia de uso del software. ***Se pide incorporar esta funcionalidad.***

**Mantenimiento Perfectivo**

Se desea añadir 2 nuevas funcionalidades a la aplicación:

* [OPCIONAL] Un módulo de alertas, de tal manera que los alumnos reciban una notificación cada vez que hay un cambio en una de sus asignaturas
* Un módulo de sugerencias, mediante el cual los alumnos pueden comunicar sugerencias, quejas y errores y que deben generar una alerta dirigida a los administradores.

**Mantenimiento Adaptativo**

Otro de los deseos de los usuarios es poder contar con una interfaz “*responsive”*, es decir adaptable a diversos dispositivos móviles con distintas resoluciones, por lo que se desea que se implemente dicha interfaz.

**Mantenimiento Preventivo**

En aras de mejorar la calidad del sistema y de evitar errores futuros, se pide analizar el sistema heredado y aplicar cuantas pruebas y modificaciones se consideren oportunas para mejorar la aplicación.

# ANEXO 1. PUNTUACIONES CÓDIGO

En este anexo, se encuentran la Tabla 1 la rúbrica del código fuente, donde se detallan las puntuaciones por cada apartado.

*Tabla 1. Rúbrica Práctica 6 y 7. Código Fuente.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Aspecto** | | | | **Escala de calificación de los ejercicios** | | | |
|  | **Peso** | **10 excelente** | **8 notable** | | **6 bien** | **5 suficiente** | **1 insuficiente** |
| Diseño y manipulación de objetos | 30% | Correcto diseño y manipulación de objetos |  | | Uso de objetos de forma incorrecta en algunos casos |  | No diseña, ni utiliza los objetos correctamente |
| Diseño y manipulación de bases de datos. | 30% | Uso correcto de herencia, abstracción y polimorfismo | Uso correcto en la mayoría de las situaciones | |  |  | Uso incorrecto de herencia, abstracción y polimorfismo |
| Detección y modelado de relaciones entre clases | 10% | Relación correcta entre clases |  | | Algunas clases no están bien relacionadas |  | Las clases no están bien relacionadas |
| Diseño y manipulación de Excepciones | 5% | Correcto diseño y manipulación de excepciones | Correcta manipulación de excepciones, pero problemas en su diseño | | Uso de excepciones de forma incorrecta en algunos casos |  | No diseña, ni utiliza las excepciones correctamente |
| Diseño y manipulación de interfaces | 5% | Correcto diseño y manipulación de interfaces |  | |  |  | No diseña, ni utiliza los interfaces correctamente |
| Correcto uso de modificadores de visibilidad | 2,5% | Correcto diseño y manipulación de modificadores |  | |  |  | No diseña, ni utiliza los modificadores correctamente |
| Correcto uso de métodos sobrecargados | 2,5% | Correcto diseño y uso de métodos sobrecargados |  | |  |  | Incorrecto diseño y uso de métodos sobrecargados |
| Aplicación de todos los conocimientos sobre programación aprendidos hasta el momento | 10% | Utiliza correctamente los conocimientos adquiridos anteriormente | Utiliza correctamente los conocimientos adquiridos anteriormente en la mayoría de casos | | Utiliza correctamente los conocimientos adquiridos anteriormente en algunos de casos |  | No utiliza correctamente los conocimientos adquiridos anteriormente |
| Uso de patrones de desarrollo | 5% | Uso correcto |  | |  |  | Uso incorrecto |
| Claridad del código fuente | -10% | El código no está bien estructurado, ni hace uso de comentarios | El código no está bien estructurado o no se hace uso de comentarios | |  | El código fuente no es del todo claro y hace uso de comentarios irrelevantes | El código fuente es claro y hace uso de comentarios descriptivos |