

# UNNOBA

## Programación Orientada a Objetos

### 2018

## Práctica 2: Patrones, Modelado OO

### Presentación y objetivos

El objetivo de esta actividad es la generación de un modelo de clases que cumpla con las especificaciones que se detallan en el documento Trabajo de cursada 2018, además se deberá realizar la codificación en lenguaje Java de cada una de las clases definidas en el modelado con sus atributos y relaciones correspondientes.

### Materiales y recursos

Para el desarrollo de la actividad deberán utilizarse los siguientes materiales y recursos:

- Clases 1 y 2.
- Clase Patrones de Diseño.
- Libro Design Patterns.
- Foro de actividades prácticas.
- Bibliografía disponible en biblioteca:
  - An introduction to object-oriented programming.
  - Análisis y diseño orientado a objetos con aplicaciones.
  - Fundamentos del diseño y la programación orientada a objetos.
  - Head first object-oriented analysis and design.
  - The java programming language.
- Software de desarrollo:
  - Netbeans o Eclipse.

### Actividades

1. Realizar un diagrama de clases en UML con las clases necesarias para poder cumplir con las especificaciones del Trabajo práctico incremental para la cursada 2018.
  - a. Se deberá confeccionar el diagrama mediante la utilización de un software para tal fin como ArgoUML o Violet UML. Luego, para su entrega, se deberá incluir una imagen que sea el resultado de la exportación del diagrama.
2. A partir del diagrama de clases previamente elaborado se pide codificar cada una de las clases.
  - a. Crear un proyecto utilizando como IDE Netbeans o Eclipse.
  - b. El proyecto tiene seguir las especificaciones de una aplicación web y utilizando Maven como herramienta para la gestión y construcción del proyecto.

- c. El modelo de clases deberá incluirse dentro del paquete `ar.edu.unnoba.poo2018.model`
- d. Se pide respetar la convenciones de nombrados de Java para clases, métodos, atributos y constantes.
- e. Se deberá utilizar Git como herramienta de versionado para el código del proyecto.
  - i. Conocer el propósito de los siguientes comandos de git y utilizar aquellos necesarios para comenzar a versionar el código:
    - 1. `git init`
    - 2. `git add`
    - 3. `git commit`
    - 4. `git pull`
    - 5. `git push`
    - 6. `git merge`
    - 7. `git branch`
    - 8. Archivo `gitignore`
  - ii. Como modelo de branches propuesto se pide tener un branch con el nombre `development`. Este será el branch que tendrá todo el código del trabajo incremental. Luego, al momento de comenzar a trabajar en una nueva práctica se deberá crear un branch a partir de `development` para poder trabajar sobre las actividades específicas de la práctica. Al finalizar la práctica el branch deberá ser mergeado con el branch `development`. La convención de nombrado para cada uno de los branches de las prácticas es `features-poo2018-tp[N]`, donde [N] será el número de la práctica
  - iii. El versionado de código será local, es decir, no será solicitado requerido subir los cambios de código a un repositorio central.

## Entrega

La entrega podrá ser grupal con un máximo de 2 personas por grupo y se deberá entregar subiendo a Virtual un único archivo adjunto en formato “.zip”. El archivo adjunto deberá contener la imagen del diagrama de clases, todo el código fuente de la solución planteada, notas e información adicional que considere pertinente para la correcta ejecución del programa realizado.