

# Programación Orientada a Objetos 2024

## Práctica 2

### 1. Design principles and Design Patterns (entregar)

- SOLID, es un acrónimo de 5 importantes principios de la programación orientada a objetos.
  - Tomar dos principios, explicar y brindar un ejemplo donde se presenta un incumplimiento y cómo lo resolvería aplicando el principio.
- Sobre el patrón de diseño Singleton, explique:
  - ¿Porque se puede considerar que no cumple con el principio de Single Responsibility?
  - ¿Porque se considera que dificulta realizar tests de unidad?
- Dependency Injection design pattern
  - Explique el patrón
  - ¿Cuál es la diferencia entre Dependency Inversion y Dependency Injection?
  - Brinde ejemplos de código donde se aplique.

### 2. Java Build Tools

- ¿Qué es una Build Tool en Java?
- Describir brevemente:
  - Apache Ant
  - Apache Maven
  - Gradle
- Apache Maven
  - Describa la estructura de directorios de un proyecto Maven en Java
  - ¿Qué es el archivo POM de maven?
  - ¿Cuál es el propósito del archivo application.properties?
  - Explique brevemente el Build Lifecycle de maven y cada una de sus etapas.

### 3. Spring.io

- Realice una investigación sobre el framework Spring en Java, explique brevemente el propósito de:
  - Spring Boot
  - Spring Web

- Spring Data JPA

## 4. Versionado de código

- ¿Qué son las herramientas para el versionado de código?
- ¿Cuáles son las diferencias en las herramientas de versionado de código Subversion (SVN) y GIT?
- GIT
  - Explique los comandos: clone, push, pull, add, commit, push, branch, merge, log.
- Explique que es un workflow o branching model
- Explique en qué consiste la técnica de code review / pull request

## 5. Inicialización del repositorio y proyecto (entregar)

- Antes de comenzar
  - Es necesario disponer de una instancia de un motor de base de datos (MySQL, Postgre) instalado y disponer un schema o base de datos (vacío, sin tablas).
  - Deberá tener instalado JDK 17 o posterior o en sus versiones correspondiente en OpenJDK.
  - Disponer de un usuario en Github y tener instalado localmente Git
    - sobre como instalar Git
    - <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
- Crear nuevo repositorio en Github
  - Ingresar a <https://github.com/new>
  - Completar el formulario
    - en repository name completar poo2024-[sede]-[grupo]
    - en description similar POO 2024 - [sede] - [grupo]
    - marcar como proyecto privado
    - tildar Add a README file
    - Add .gitignore seleccionar JAVA como template
  - Clic en Create repository
  - Una vez creado el repositorio agregar como colaboradores a todos los miembros del grupo y también a los profesores teoría, práctica y consulta (según cada sede)
    - Ingresar a opción Settings
    - Luego ingresar a la opción Collaborators
    - En manage access agregar los usuarios correspondientes mediante el botón add people. El profesor de la práctica informará los datos de las cuentas de los usuarios.
  - Descargar el repositorio mediante el comando git clone siguiendo los pasos indicados en <https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository>
    - Si tienen habilitado el 2FA y van a clonar vía HTTPS requerirán generar un personal access token en

<https://github.com/settings/tokens> y utilizarlo cuando git solicite la contraseña.

- Crear proyecto Java
  - Completar el siguiente formulario en <https://start.spring.io/> para generar la estructura básica de un proyecto maven con las dependencias Spring Web, Spring Data JPA y el driver JDBC para su base de datos.
  - Descargar el proyecto generado y descomprimirlo.
  - Mover el contenido del directorio (incluyendo archivos que pueden estar ocultos como .mvn o .gitignore) dentro de la carpeta del repositorio.
- Primer commit / push
  - Mediante git add agregar al versionado la estructura inicial de directorios y archivos
  - Realice el primer commit agregando como comentario "first commit"
  - Realice el push para subir el commit al repositorio en github.
- Crear una nueva rama
  - Mediante el comando git branch -b develop. Creará un nuevo branch con el nombre develop y se posicionará sobre él.
  - Luego mediante el comando git push -u origin develop subimos del branch recién creado al repositorio en github.
- Primeros cambios
  - Abrir con el IDE el proyecto generado.
  - Identifique y analice el contenido del archivo pom.xml
  - Ejecute el proyecto
  - Determine cuál es la causa por la cual falla al ejecutar.
  - Busque qué configuración debe agregar para poder resolver el inconveniente.
  - Cuando logre resolver el inconveniente, realice un nuevo commit para versionar los cambios que generan el error inicial.
  - Mediante el comando git push suba los cambios al branch del repo en github.
- Realizar un pull request
  - En github dentro del repo creado ir a la opción de menú Pull requests y hacer clic en New pull request.
  - Establecer como base branch a main y en compare a develop. Se van a visualizar las diferencias entre las dos ramas.
  - Clic en Create Pull request
  - Asignar como reviewer a otro miembro del grupo
  - El miembro asignado deberá aprobar el cambio
  - Una vez aprobado el pull request realizar un git pull en nuestro repositorio local

## 6. Entrega

- Un miembro por grupo deberá subir en un único directorio comprimido la resolución del ejercicio 1 (puede ser un documento de texto donde expliquen dos de los cinco principios solid) y la resolución del ejercicio 5 con el

directorio que contiene el repositorio del proyecto de JAVA creado a partir de spring.io con todos los pasos realizados.