

# Reglamento - Práctica

...

75.10 Técnicas de Diseño

# Objetivos

# Objetivos

Buscamos que los alumnos puedan: -

- Comprender e incorporar los principios de diseño y las buenas prácticas de desarrollo de software. -
- Comprender y saber cuando y como aplicar los patrones de diseño. -
- Analizar y proponer distintas alternativas de diseño a un problema dado teniendo en cuenta las pros y contra de cada uno.

# Reglamento

# Asistencia

En las clases donde se realizan entregas o presentaciones del trabajo realizado por el alumno o el grupo, se tomará asistencia, ya que en las mismas los alumnos serán evaluados por el ayudante, asentando una nota por dicha entrega y defensa de la misma.

---

# Evaluaciones y Entregas

- No hay evaluación tipo parcial en la práctica, pero hay TPs y entregas que funcionan como evaluaciones y que deberán ser aprobadas para aprobar la cursada.
  - Los TPs que no cumplan los requerimientos o no sean entregados en el tiempo y forma definido para el mismo quedarán **desaprobados**, desaprobando la cursada.
-

# Trabajos Prácticos

- Se realizará un **TP0 individual** cuyo principal objetivo es la configuración del ambiente que se utilizará para los otros TPs.
  - Se realizará un **TP grupal** dividido en 3 iteraciones o entregas consecutivas, donde en cada uno se agregan una serie de requerimientos sobre el mismo.
-

# Trabajos Prácticos

- Solo se podrá desaprobado una sola entrega (incluido TP0, y las 3 del TP general) para mantener la regularidad en la práctica y la materia.
  - La cantidad de integrantes se definirá próximamente en base a la cantidad de alumnos y ayudantes.
-



# Tooling

- Para los trabajos prácticos se utilizará Java 8, IntelliJ como IDE y GitHub como repositorio.
- Para considerar una entrega como válida, todos los chequeos establecidos en Gradle deberán correr sin generar warnings, tanto local como en el servidor de IC.

---

# Tooling

- Si encuentran alguna regla que no aplique o incorrecta incluida en los chequeos incluidos en Gradle, comunicar durante la práctica o por Slack (**#quejas-tooling**) lo antes posible para poder evaluarla y considerar su eliminación.
  - Se pedirá que el código esté en **inglés** (nombre de variables, métodos, atributos, etc.).
-

# GitHub

<https://7510-tecnicas-de-disenio.github.io/>

---

# Criterios de Corrección

- **Compleitud:** Que esté entregado y completo todo los requerimientos del enunciado con su respectiva documentación.
  - **Correctitud:** Que la propuesta de pidió y sin errores funcionales.
-

# Criterios de Corrección

- **Distribución de responsabilidades:**  
Evaluación de cuán bien, cuán criterioso y cuán prolija se hizo la distribución de responsabilidades del problema a resolver entre las clases del diseño.

---

# Criterios de Corrección

- **Aplicación y cumplimiento de los criterios de buen diseño:** Si fueron aplicados y si no fueron violados.
  - **Uso y aplicación de patrones:** Si utilizaron patrones, y si los utilizaron con criterio y bien aplicados.
-

# Criterios de Corrección

- **Documentación** que acompaña a la entrega, organización de los archivos entregados. Pero orientada a contar, explicar justificar las decisiones de diseño tomadas en el desarrollo realizado.

---

# Criterios de Corrección

En caso de que el tp no cubra el 100% de la funcionalidad pedida el grupo quedará **desaprobado**.

Si el tp cubre el 100% la funcionalidad pedida pero viola gravemente un criterio de buen diseño quedará **desaprobado**.

---



# Criterios de Corrección

## No hay re-entregas

La interacción con los ayudantes será importante para alertar a los alumnos durante el desarrollo de la práctica.

---

# Documentación

1. Carátula: Datos de los Integrantes, ayudante asignado, calendario de entregas.
2. Tabla de contenidos.
3. Enunciado del problema a resolver.

---

# Documentación

## 4. Desarrollo de la solución Diseño de la solución

Descripción de la solución propuesta, con foco en el diseño (deben responder a las preguntas de cómo modelaron el dominio del problema, que conceptos representan en su solución y que significa cada uno de ellos, cuál es la estrategia de solución).

---

# Documentación

## 4. Desarrollo de la solución Diseño de la solución

Diagramas de clase orientados a mostrar los puntos claves de solución (las jerarquías de herencia no deberían estar en todos los diagramas, sobre todos en los que se explique puntos de diseño en los que no intervengan). Descripción del diagrama que ponga más en foco el diseño solución que se muestra (como están representados los conceptos en el modelo de clases y cuales son responsabilidades de las clases). En los diagramas de clase se deben omitir los getters/setters.

# Documentación

## 4. Desarrollo de la solución Diseño de la solución

Diagramas UML que validen el diseño propuesto: Los diagramas de secuencia deben coincidir los nombres de las llamadas y parámetros con los del diagrama de clase. Ambos diagramas deben coincidir con el código implementado (los diagramas también se deben actualizar).

---

# Documentación

## 4. Desarrollo de la solución Diseño de la solución

Cualquier descripción adicional que muestre evidencia de falencias del diseño, virtudes (código clausurado ante cambios, inversión de la cadena de dependencia, es decir los principios de buen diseño), implementación de patrones.

---

# Documentación

5. Conclusiones: Es importante que vuelquen la experiencia en el desarrollo del trabajo práctico y los conceptos aprendidos, como así también alguna crítica del mismo.

---

# Notas

La nota estará conformada como:

$0,3 * \text{Nota Concepto individual} +$

$0,7 * \text{Nota Tp Grupal}$

---



# Notas

## **La nota ES subjetiva.**

Si bien, intentamos reducir ésta subjetividad, es imposible tener un esquema objetivo de evaluación ya que cada TP es único en multitud de aspectos.

---