

## 2DO EXAMEN PARCIAL

**CARRERA:**

Tecnicatura Universitaria en Programación de Sistemas

**MATERIA:**

Diseño Orientado a Objetos

**BIMESTRE/CUATRIMESTRE Y AÑO**

2DO BIM. /2024

**PROFESOR:**

Lic. Matias Velasquez

### INDICACIONES GENERALES PARA LOS ESTUDIANTES

Antes de comenzar a desarrollar el examen, recuerde que el documento debe tener una portada en el que deben indicarse los siguientes datos: **nombre, número de matrícula, carrera y materia.**

El examen puede ser individual o grupal de acuerdo a lo propuesto por el profesor en las consignas.

La entrega del archivo se debe realizar en tiempo y forma en el espacio correspondiente, sin excepción. Es importante que verifiquen la fecha de entrega en el cronograma del aula.

**En el caso de exámenes grupales todos los estudiantes deben realizar su entrega en la fecha indicada por el cronograma.** Aquel que no efectúe el envío o suba un documento diferente, se lo consignará como **Ausente**, y deberá rendir en instancia de recuperatorio según corresponda.

**Recuerde que para la aprobación de la materia deberá:**

- **Aprobar dos exámenes parciales:** un primer parcial y segundo parcial (integrador) o,

- **Aprobar uno de los dos exámenes parciales y un recuperatorio**, si uno de ellos no hubiera sido aprobado o estuviera ausente.
- **Aprobar el examen final** oral integrador.

La **nota mínima** de aprobación corresponde a **4 (cuatro)** puntos. La **calificación** se realizará mediante un puntaje, del 1 al 10, y/o a través de un porcentaje de aprobación.

Se valorará la participación activa en los foros, el cumplimiento con las actividades y la presencia en las conferencias web.

En caso de tener **inconvenientes técnicos** al momento de subir el archivo, realice captura de pantalla y remita a la casilla de Ayuda Técnica ([ayudatecnica@campusvirtual.uces.edu.ar](mailto:ayudatecnica@campusvirtual.uces.edu.ar)), detallando el número de matrícula, nombre y apellido y asignatura en que sucedió el inconveniente, adjuntando:

- **El print de pantalla si existiese un problema propio del campus**. Es fundamental que sea claro y muestre toda la información: **horario, fecha, intento, error**. Le sugerimos que utilice el botón “Impr Pant Pet Sis”. Sin este registro la instancia no se contabilizará y se lo calificará como **Ausente o Desaprobado** según corresponda.
- **El número de reclamo en caso de ser un problema de luz/conectividad propia del alumno**. Sin este registro la instancia no se contabilizará y se lo calificará como **Ausente**.

***No se considerará entregada ni se evaluará aquella actividad que sea enviada por fuera del espacio correspondiente, como tampoco la que no se encuentre completa y/o corresponda a otra asignatura.***

Una vez que el área reciba esta información se analizará en profundidad y **dentro de las 48hs hábiles se brindará la respuesta correspondiente**.

#### **REQUISITOS FORMALES DE PRESENTACIÓN DEL EXAMEN:**

Dado que la actividad se realizará utilizando GitHub como plataforma principal para la entrega y desarrollo del trabajo, se solicita que en el campus virtual solo se presente un enlace al repositorio correspondiente. No se aceptarán archivos adjuntos ni

documentos adicionales, ya que parte integral de la actividad es utilizar esta herramienta colaborativa para gestionar y compartir el trabajo realizado.

### **TIEMPO DE DESARROLLO:**

Tiempo Límite para la Actividad:

- Fecha de Habilitación de Consignas: 27 de noviembre, a las 08:00
- Fecha Límite de Entrega: 29 de noviembre, a las 23:55
- Número de Intentos: Un único intento

Tiempo Total para Realizar la Actividad:

- El tiempo total para realizar la actividad es de 3 días desde la fecha de habilitación de consignas hasta la fecha límite de entrega.

Temporizador y Número de Intentos:

- Se otorga un único intento para realizar la actividad.
- No hay un temporizador específico para esta actividad, pero se espera que los estudiantes la completen dentro del período de 7 días asignado.

### **CRITERIOS DE EVALUACIÓN:**

- ✓ Comprensión e interpretación de la consigna formulada.
- ✓ Dominio adecuado de los temas abordados.
- ✓ Aplicación de la teoría a situaciones prácticas.
- ✓ Manejo del vocabulario específico.
- ✓ Pertinencia y claridad en la redacción.
- ✓ Estilo académico, correcta construcción gramatical, puntuación y acentuación.
- ✓ Cumplimiento de las pautas de presentación de trabajos escritos.

### **CONSIGNAS**

El segundo parcial consta de tres partes: primero en re-entregar las dos actividades obligatorias (Actividad Obligatoria N°3 y Actividad Obligatoria N°4) con las correcciones correspondientes, segundo sumar la aplicación de los patrones de diseño, y tercero optimizar el uso del readme.md.

## Aplicación de Patrones de Diseño

El gimnasio "Fitness Pro" enfrenta problemas porque sus procesos son manuales y poco eficientes. Estos problemas afectan la administración de socios, reservas de clases, manejo de pagos, y más. Como desarrolladores, su tarea es diseñar una solución que haga más eficiente la gestión del gimnasio aplicando los patrones de diseño.

### Objetivo:

Deben aplicar al menos un patrón de diseño para mejorar el diseño del sistema de gestión del gimnasio "Fitness Pro". Estos patrones les ayudarán a crear un sistema que sea fácil de modificar, expandir, y mantener en el futuro.

### Requisitos de entrega:

#### 1. Selección e Implementación de un Patrón de Diseño:

- Identificar un problema específico en la gestión del gimnasio descrita en el enunciado.
- Seleccionar un patrón de diseño que aborde ese problema (ejemplo: *Singleton*, *Observer*, *Factory Method*, *Strategy*, entre otros).
- Implementar el patrón de diseño en un diagrama de clases del proyecto, con la lógica correspondiente.

#### 2. Creación de un Documento Explicativo (**PatronDeDiseno.md**):

Este archivo debe incluir:

- **Propósito y Tipo del Patrón:**

Una breve explicación del problema y cómo el patrón seleccionado lo soluciona.

Ejemplo: "El patrón **Observer** se utiliza para notificar a los sistemas de reservas y membresías cuando un socio realiza una acción que los involucra, como registrarse en una clase."

- **Motivación:**

Un detalle más profundo del problema que enfrentaba el sistema y cómo el patrón de diseño ayuda a resolverlo.

Ejemplo: "El control manual de acceso al gimnasio genera largas filas.

El patrón **Singleton** permite implementar un sistema centralizado de

acceso con una única instancia que asegura integridad y sincronización en los datos de los socios."

- **Estructura de Clases:**

Incluir un diagrama UML que muestre cómo las clases del proyecto se relacionan entre sí al aplicar el patrón.

- **Ejemplo de Código:**

Incluir un fragmento de código que demuestre la implementación del patrón en el proyecto (puede ser pseudocódigo o un lenguaje como JavaScript, Python o Java).

### 3. **Actualización del [README.md](#):**

Incluir en el archivo una sección que explique:

- Qué patrón fue implementado.
- Qué problema específico del sistema ayuda a resolver.
- Un enlace al documento explicativo ([PatronDeDiseno.md](#)).

### 4. **Nueva Versión del Diagrama de Clases:**

- Crear una versión actualizada del diagrama de clases del sistema que refleje cómo el patrón de diseño ha sido integrado.

### 5. **Simplificación del Archivo [README.md](#)**

Reorganizar el contenido del archivo [README.md](#) para hacerlo más navegable y legible. En lugar de incluir todo el contenido en un único archivo, se deben dividir las secciones extensas en diferentes páginas dentro del repositorio, enlazándolas desde el archivo principal.

---

## **Requisitos para la Entrega del Nuevo [README.md](#)**

### 1. **Estructura Simplificada del Archivo:**

El archivo principal [README.md](#) debe incluir una introducción breve al proyecto y enlaces claros a las páginas correspondientes dentro del repositorio. La estructura mínima debe ser:

- **Introducción:**
  - Una breve descripción de la aplicación y su propósito.
  - Resumen de los requisitos abordados y análisis realizado en la Actividad Obligatoria N°1.
- **Diagramas y Diseños:**

- Enlace a una página con todos los diagramas UML relacionados con el proyecto (clases, actividades, secuencias, casos de uso, etc.).
  - **Herramientas Agile:**
    - Enlace a una página que explique las herramientas utilizadas (Matriz CLAE, Tarjetas CRC, etc.).
  - **Anexos:**
    - Enlace a una página que detalle los anexos (por ejemplo, la aplicación de principios SOLID, patrones de diseño).
2. **Uso de Markdown para la Navegación:**
- Crear un archivo por cada sección extensa (por ejemplo, [Diagramas.md](#), [HerramientasAgile.md](#), [Anexos.md](#)).
  - Asegurarse de que todos los enlaces dentro del [README.md](#) sean funcionales y apunten correctamente a los archivos dentro del repositorio.
  - Utilizar una estructura de directorios si es necesario para mantener ordenados los archivos relacionados con el proyecto.
3. **Permisos de Lectura:**
- Garantizar que los enlaces funcionen correctamente para cualquier persona con acceso al repositorio.
  - Probar los enlaces para confirmar que redirigen al contenido adecuado dentro del repositorio.

Ejemplo del nuevo [readme.md](#) propuesto:

## Introducción

---

(Descripción de la aplicación y su propósito, incluye los requisitos que se abordan y el análisis que hiciste en la **Actividad Obligatoria N°1.** )

## Diagramas y Diseños

---

- Diagramas UML

- Herramientas Agile
- Anexos

Ejemplo del nuevo **Diagramas.md** prouesto

## Diagramas UML

---

- Diagrama de casos de uso
- Escenarios de casos de uso
- Diagramas de Clases
- Diagrama de Actividades
- Diagramas de Secuencia

Ejemplo del nuevo **DiagramasClases.md** prouesto

## Diagramas de Clases

---

- Diagrama de Clases v1 (Utilizando SOLID)
- Diagrama de Clases v2
- Diagrama de Clases v3 (Utilizando Patrón XXXX)

Ejemplo del nuevo **DiagramasSecuencia.md** prouesto

## Diagramas de Secuencia

---

- Caso de uso 1
- Caso de uso 2
- Caso de uso 3

- Caso de uso 4
- Caso de uso 5

Ejemplo del nuevo **HerramientasAgile.md** prouesto

## Herramientas Agile

---

- Matriz CLAE
- Tarjetas CRC

Ejemplo del nuevo **Anexos.md** prouesto

## Anexos

---

- Anexo - Aplicación de Principios SOLID
- Anexo - Aplicación de Patrón de Diseño XXXXX

Ejemplo del nuevo **PatronDeDiseno.md** prouesto

## Anexo - Aplicación de Patrón de Diseño XXXXX

---

Propósito y Tipo del Patrón: Una breve explicación del problema y cómo el patrón seleccionado lo soluciona.



## **Motivación**

---

Un detalle más profundo del problema que enfrentaba el sistema y cómo el patrón de diseño ayuda a resolverlo.

## **Estructura de Clases**

---

Incluir un diagrama UML que muestra cómo las clases del proyecto se relacionan entre sí al aplicar el patrón. Incluir una imagen incrustada, así como el enlace correctamente referenciado para ver el diagrama en detalle.

## **Ejemplo de Código**

---

Incluir un fragmento de código que demuestre la implementación del patrón en el proyecto (puede ser pseudocódigo o un lenguaje como JavaScript, Python o Java).