

## ACTIVIDAD OBLIGATORIA N°1

### **CARRERA:**

Tecnicatura Universitaria en Programación de Sistemas

### **MATERIA:**

Diseño Orientado a Objetos

### **BIMESTRE/CUATRIMESTRE Y AÑO**

1° CUAT. /2025

### **PROFESOR/A:**

Lic. Matias Velasquez

## INDICACIONES GENERALES PARA LOS ESTUDIANTES

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

La actividad es individual 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.

### **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 DE LA ACTIVIDAD:**

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. Por lo tanto, el enlace proporcionado deberá dirigirse al repositorio de GitHub que contenga todo el contenido relacionado con la actividad, incluyendo los diagramas, documentos y cualquier otra información relevante. Por ejemplo: <https://github.com/TuUsuario/Nombre-del-Repositorio>

#### **TIEMPO DE DESARROLLO:**

Tiempo Límite para la Actividad:

- Fecha de Habilitación de Consignas: 11 de marzo, a las 00:00

- Fecha Límite de Entrega: 31 de marzo, a las 23:55

Tiempo Total para Realizar la Actividad:

- El tiempo total para realizar la actividad es de 20 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 20 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**

#### **Consigna del Primer Entregable - Diseño Orientado a Objetos**

**Objetivo:** El objetivo de este primer trabajo es sentar las bases conceptuales y de diseño para el sistema que desarrollaremos a lo largo del cuatrimestre, aplicando el paradigma de programación orientada a objetos (POO).

#### **Problema a Resolver durante el Cuatrimestre:**

Un pequeño centro de salud está empezando a digitalizarse y necesita una herramienta para organizar sus turnos médicos. Actualmente, todo lo manejan en una agenda física y eso está generando problemas: se pierden citas, hay pacientes que no reciben confirmación de sus turnos, y a veces se asignan dos pacientes al mismo horario. También necesitan llevar un registro del historial de turnos de cada paciente y asegurarse de que los médicos no tengan más turnos de los que pueden atender. Además, les gustaría que el sistema les avise automáticamente cuando un turno es

confirmado, cancelado o modificado, tanto a los pacientes como a los profesionales. Quieren que la información de contacto de los pacientes y médicos esté protegida y solo sea accesible para el personal autorizado.

En la agenda de turnos se registra la fecha y hora del turno, el estado del turno (confirmado, cancelado, pendiente), el médico asignado y el paciente correspondiente. También se debe guardar un motivo del turno y observaciones si las hubiera.

De cada paciente se necesita almacenar su nombre completo, número de documento, fecha de nacimiento, información de contacto (teléfono, correo electrónico) y un historial de turnos asociados. Para cada profesional de la salud, se debe registrar su nombre, matrícula profesional, especialidad, horario de atención y datos de contacto.

El sistema debería permitir registrar tanto a los pacientes como a los profesionales de la salud, asignar turnos según la disponibilidad de cada profesional, y llevar un historial de turnos realizados. También debería poder enviar notificaciones, ya sea por correo electrónico o por mensaje de texto, cuando haya algún cambio en un turno.

Para este primer entregable, se espera que los estudiantes identifiquen las posibles clases, atributos y métodos necesarios para resolver este problema, aplicando los principios de POO

### Formato de Entrega:

1. **Repositorio en GitHub:** Crear un repositorio público llamado `SistemaGestionTurnos`.
2. **README.md:** Documento principal que describa brevemente el proyecto y contenga una sección de “Anexos”, que enlace a un archivo `anexos.md`.
3. **anexos.md:** Archivo que dirija a otro archivo llamado `introduccion.md`.
4. **introduccion.md:** Debe contener lo siguiente:
  - **Descripción del paradigma orientado a objetos:** Definir qué es la programación orientada a objetos y por qué es importante.
  - **Los cuatro fundamentos de POO:** Describir con ejemplos del mundo real (Usar diagramas o bocetos simples, no es necesario UML ni código en esta etapa)
  - **Requisitos iniciales del sistema:** Describir al menos cinco requisitos funcionales.

- **Casos de uso:** Desarrollar cinco casos de uso con el formato adecuado:
  - Nombre del caso de uso
  - Actor(es) involucrado(s)
  - Descripción breve
  - Flujo principal de eventos
  - Precondiciones
  - Postcondiciones
- **Boceto inicial del diseño de clases:** Incluir una imagen incrustada en el archivo y un enlace para visualizarla en línea. Por el momento no es necesario que el diseño inicial siga un formato UML, puedes usar alguna herramienta como [excalidraw](#), [Draw.io](#) o [umletino](#).

Ejemplo del nuevo **README .md** prouesto

<b>Nombre de la aplicación - Diseño del Sistema</b>
<b>Apellido Nombre</b>
<b>Materia - Carrera</b>
<b>Profesor</b>
<b>Año</b>

## 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

---

- Anexos

(Asegurate de que los permisos de lectura sean los correctos y que los enlaces se vean de la forma adecuado correspondiente a la construcción de un documento en [markdown](#))

Ejemplo del [anexos.md](#) prouesto

## Anexos

---

- Anexo - Introducción al Diseño Orientado a Objetos

Ejemplo del [introduccion.md](#) prouesto

## Anexo - Introducción al Diseño Orientado a Objetos

---

\_Definir qué es el paradigma orientado a objetos y por qué es importante.\_

## Los cuatro fundamentos de POO

---

Describir con ejemplos del mundo real (Usar diagramas o bocetos simples, no es necesario UML ni código en esta etapa)

## Requisitos iniciales del sistema

---

Describir al menos cinco requisitos funcionales.

## Casos de uso

---

Desarrollar cinco casos de uso con el formato adecuado

## Boceto inicial del diseño de clases

---

Incluir una imagen incrustada en el archivo y un enlace para visualizarla en línea.

### **PUNTAJE POR CONSIGNA:**

- **Formato de Entrega (1 punto)**
  - Repositorio, estructura de archivos y organización correcta, usando el formato markdown correcto.
- **Introducción a POO (2 puntos)**
  - Definición del paradigma y su importancia.
  - Descripción de los cuatro pilares con ejemplos.
- **Requisitos iniciales del sistema (1 punto)**
  - Al menos cinco requisitos funcionales claros y relevantes.
- **Casos de uso (2 puntos)**
  - Cinco casos de uso completos con todos los elementos necesarios.
- **Boceto inicial del diseño de clases (4 puntos)**
  - Identificación correcta de clases, atributos y métodos.
  - Coherencia del diseño con los principios de POO.
  - Presentación visual clara y comprensible.

**Total: 10 puntos**

Nota: Los siguientes entregables se construirán sobre esta base, expandiendo y refinando el diseño del sistema a lo largo de las unidades subsiguientes.