

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

PROGRAMACIÓN 3
Examen 2
(Segundo semestre 2024)

Indicaciones generales:

- **Duración:** 3 horas.
- **Materiales o equipos a utilizar:**
La sección teórica se realizará sin material de apoyo y antes de la sección práctica.
En la sección práctica podrá utilizar material de apoyo: diapositivas, ejemplos de clase, grabaciones de clase, código fuente y documentación oficial de Microsoft u Oracle.
Todo el material que va a utilizar para el desarrollo del examen debe descargarse antes de iniciar con el mismo.
- No está permitido el uso de ningún material o equipo electrónico adicional al indicado.
- **La presentación, la ortografía y la gramática influirán en la calificación.**

Puntaje total: 20 puntos

PARTE TEÓRICA: (Resolver en PAIDEIA) (2 puntos)

Ingrese a PAIDEIA y conteste en la misma PLATAFORMA dos preguntas teóricas asignadas.

PARTE PRÁCTICA: (18 puntos)

(Resolver empleando NETBEANS 22, JDK 21, VISUAL STUDIO 2022, .NET Framework 4.8.1, GlassFish Server 7.0.18)

Trabaje en la UNIDAD T: para evitar perder la información.

Asegúrese de configurar el GlassFish en el Netbeans antes de iniciar con la programación.

Caso de estudio:

La institución **Horizontes Académicos S.A.** es una empresa dedicada a ofrecer becas de posgrado a personas que han culminado sus estudios superiores y cuentan con título universitario. Para cumplir con este encargo, la empresa requiere implementar una interfaz web que le permita a todos los interesados registrarse como postulantes a estas becas que se ofrecen. El formulario deberá solicitar datos como: el DNI, el nombre, el apellido paterno, la universidad de procedencia, el promedio acumulado en pregrado, los logros destacados (específicamente, si perteneció al tercio superior, si participó como integrante en algún grupo de investigación y si fue un deportista calificado). Además, el formulario deberá solicitar la modalidad de estudios preferida para cursar los estudios de posgrado.

La Fig. 01 muestra el formulario solicitado. Asimismo, debe tener en cuenta que, para el registro de la información de un postulante, es requerido que se muestren las universidades desde la BD en el componente *dropdownlist*.

The image shows a web form titled "Registrar Postulante" for the Pontificia Universidad Católica del Perú. The form is part of a system for registering applicants for postgraduate studies. It includes a sidebar with the PUCP logo and a "Registrar Postulante" button. The main form area contains several input fields and a dropdown menu. The fields are: DNI (87309981), Nombre (MANUEL), Apellido Paterno (DELGADO), Universidad Origen (PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU), Promedio Acumulado Obtenido en Pregrado (17.28), Logros en Pregrado (1/3 SUPERIOR, PERTENENCIA A GRUPO DE INVESTIGACION, DEPORTISTA CALIFICADO), and Modalidad de Estudios Preferida (PRESENCIAL, VIRTUAL, HIBRIDA). At the bottom, there are buttons for "Regresar" and "Guardar".

Fig. 01. Formulario web de registro de postulante solicitado

Modo de implementación:

La forma de implementar el formulario web debe seguir las indicaciones comunicadas el día viernes 29 de noviembre del presente (las cuales se presentan como anexo al final del enunciado). De forma resumida, se detalla que el sistema constará de tres componentes principales:

- Servidor RMI (Java): que ejecutará las reglas de negocio y se encargará de la conexión con la base de datos.
- Servidor Web (Java): que será cliente del servidor RMI (pues consumirá objetos remotos proporcionados por este) y ofrecerá servicios web de tipo SOAP con las funcionalidades requeridas.
- Cliente Web Front-End (C#): que implementa la interfaz gráfica y consume los servicios web.

Considere que el usuario ingresa correctamente la información en el formulario (no son requeridas validaciones). Utilice el Script SQL que se encuentra en PAIDEIA, así como el archivo .aspx para la implementación de lo solicitado. En la Figura 02 se muestran las tablas y relaciones que genera el Script SQL. Asimismo, en la Figura 03 se muestra el diagrama de clases de la capa model.

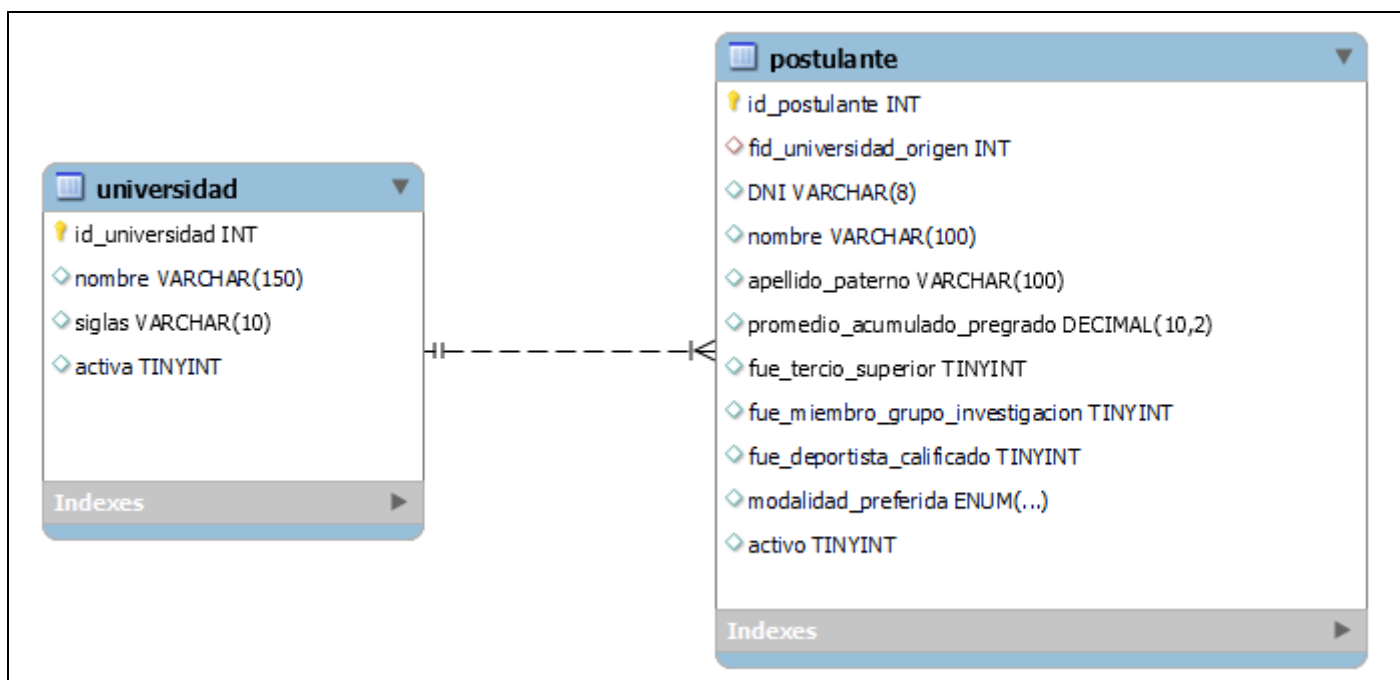


Fig. 02. Diagrama EER de la base de datos

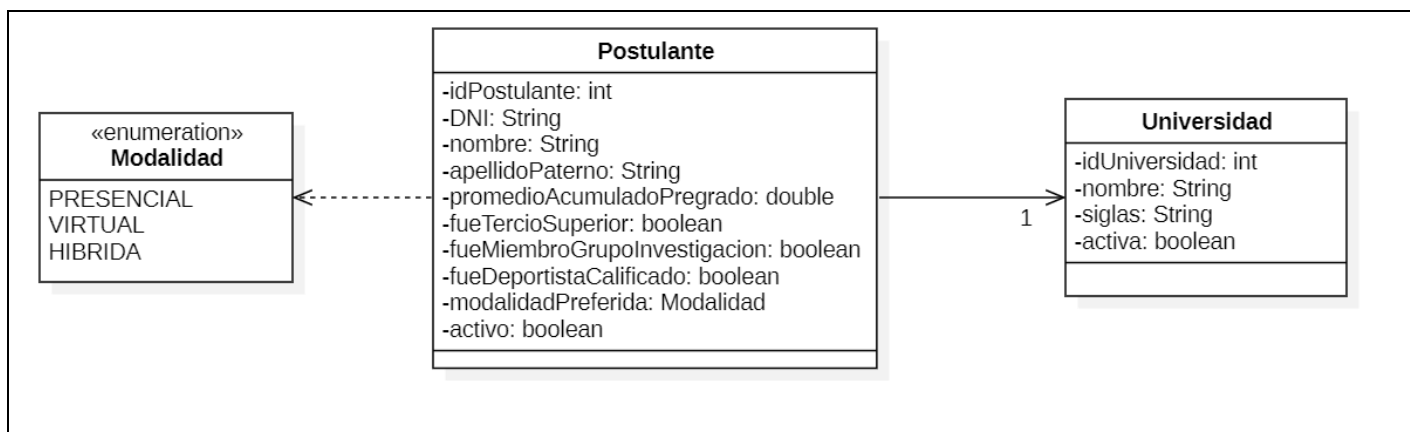


Fig. 03. Diagrama de clases de la capa model

Rúbrica de Calificación (la corrección será progresiva):

Aspecto de evaluación	Puntaje
A. Implementación del DBManager	2.0 puntos
B. Implementación del Model	2.0 puntos
C. Implementación del Back-End	4.0 puntos
D. Implementación del Servidor RMI	3.0 puntos
E. Implementación del Servidor Web / Cliente RMI	3.0 puntos
F. Implementación del Front-End	4.0 puntos

Profesores del curso:

Dr. Freddy Paz
Dr. Andrés Melgar
Dr. Heider Sánchez

San Miguel, 12 de diciembre de 2024

CONSIDERACIONES GENERALES

- Progresión por secciones:
 - No podrá avanzar a la siguiente sección si no ha completado la anterior.
 - Cada sección debe estar plenamente funcional antes de continuar.
- Uso del material de clase:
 - Puede reutilizar el código de las sesiones de clase y ejemplos trabajados, pero no podrá compartirlo con otros estudiantes durante el examen.
 - No se permite el uso de internet para descargar material o herramientas durante el examen. Todo material debe descargarse antes de iniciar el examen.
- Buenas prácticas:
 - La solución debe seguir el paradigma de Programación Orientada a Objetos (POO), priorizando encapsulamiento, herencia y polimorfismo.
 - Es obligatorio utilizar código limpio, evitando atributos o métodos no relacionados al caso planteado. Se descontará puntaje por código redundante o innecesario.
- Preparación previa:
 - Puede utilizar proyectos plantilla, herramientas configuradas y dependencias necesarias (Bootstrap, JQuery, Font Awesome, etc.). Esto ya que únicamente se le está brindando el Script SQL y un archivo .aspx para resolver el enunciado. NO se brindarán proyectos ya creados como en los laboratorios o semestres previos.

IMPLEMENTACIÓN DEL DBMANAGER

- Implemente un DBManager utilizando el patrón Singleton.
- Los datos de conexión deben leerse desde un archivo de configuración (properties, JSON, XML o TXT). Este archivo debe leerse desde una ubicación dentro del proyecto (ruta relativa).

IMPLEMENTACIÓN DEL MODEL

- Cree un modelo para dar soporte al modelo que se le ha indicado.
- Utilice una instancia vacía de tipo MySQL en un servicio RDS.
- Utilice el SCRIPT SQL para la creación de la base de datos y los procedimientos almacenados, si los necesita.

IMPLEMENTACIÓN DEL BACK-END

- Implemente el soporte al Back-End siguiendo el patrón DAO, conforme a lo trabajado en clase.

IMPLEMENTACIÓN DEL SERVIDOR RMI

- Desarrolle una capa adicional al Back-End para dar soporte a un servidor RMI. Este servidor RMI ejecutará las reglas de negocio implementadas en el paso anterior (Implementación del Back-End)
- La configuración del servidor y el puerto deben leerse desde un archivo diferente al de la base de datos (properties, JSON, XML o TXT). Este archivo debe leerse desde una ubicación dentro del proyecto (ruta relativa).

IMPLEMENTACIÓN DEL SERVIDOR WEB / CLIENTE RMI

- Desarrolle un servidor web que también actúe como cliente RMI, consumiendo objetos remotos del servidor RMI.
- Puede utilizar un proyecto web plantilla con las configuraciones y referencias necesarias (Jakarta).
- La configuración del servidor y el puerto al cual se va a conectar este proyecto web debe leerse desde un archivo (properties, JSON, XML o TXT). Este archivo debe leerse desde una ubicación dentro del proyecto (ruta relativa).

IMPLEMENTACIÓN DEL FRONT-END

- Implemente una interfaz gráfica que consuma las reglas de negocio desde el servidor web.
- Puede utilizar el archivo de la interfaz web (.aspx).
- Puede utilizar un proyecto plantilla con las dependencias descargadas (Bootstrap, JQuery, Font Awesome, etc.).

ENTREGA Y EVALUACIÓN

Entrega: Todos los archivos deben estar correctamente estructurados, con comentarios relevantes en el código, y deben ser enviados a través de la Plataforma PAIDEIA.

CRITERIOS DE EVALUACIÓN:

- Funcionalidad completa: Cada sección debe cumplir con los requisitos planteados.
- Uso correcto de POO: Aplicación efectiva de principios como encapsulamiento, herencia y polimorfismo.
- Código limpio: Legibilidad, modularidad y ausencia de redundancias.
- Eficiencia: Soluciones optimizadas para tiempo y recursos.