

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

PROGRAMACIÓN 3
Examen 1
(Primer semestre 2024)

Indicaciones generales:

- **Duración:** 3 horas.
- **Materiales o equipos a utilizar:**
La sección teórica se realizará sin material de apoyo.
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.
- 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:

(Seleccionar una sola pregunta y resolverla en el cuadernillo)

Pregunta 1 (1.0 punto)

¿Por qué aplicamos el patrón **Singleton** para acceder a la base de datos?


```
public static void main(String[] args) {  
    Connection conn = DBManager.getInstance().getConnection();  
    //...  
}
```



Pregunta 2 (1.0 punto)

¿Por qué es necesario definir una clase de tipo **interface** cuando trabajamos con ADO? ¿Por qué en el **main** no trabajamos directamente con la clase que lo implementa?

```
public interface ProductoDao {  
    //..  
}  
public class ProductoMySQL implements ProductoDao {  
    //..  
}  
public static void main(String[] args) {  
    ProductoDao productDao = new ProductoMySQL();  
    List<Producto> resultado = productDao.listar();  
    //...  
}
```



PARTE PRÁCTICA:

Pregunta 3 (19 puntos)

(Resolver empleando VISUAL STUDIO y .NET Framework 4.8.1)

La oficina del Vicerrectorado de Investigación (VRI) está requiriendo de una aplicación web que le permita gestionar toda la información relacionada a los grupos de investigación que están conformados en nuestra universidad (PUCP). Para ello, se ha diseñado la estructura de base de datos que se muestra en la Figura 01.

Como se puede apreciar en la Figura 01, los datos que se desean registrar de un grupo de investigación son: (1) el nombre del grupo de investigación, (2) su acrónimo o siglas, (3) el departamento académico al cual está adscrito (ya que todo grupo de investigación por reglamento de la universidad debe pertenecer a un departamento), (4) el tipo de investigación que realiza (el cual puede ser BASICA o APLICADA), (5) la fecha de fundación del grupo, (6) el presupuesto anual que tiene asignado el grupo por parte del Vicerrectorado, (7) una descripción del grupo, (8) una foto del grupo de investigación, y ciertos indicadores que permitan saber si posee o no con cierta infraestructura, por ejemplo, (9) si posee laboratorio, (10) si posee equipamiento especializado, y (11) si posee ambiente de trabajo dentro de las instalaciones de la universidad. Asimismo, al momento de registrar un grupo de investigación, se desea que el sistema permita especificar quiénes serán los integrantes de ese grupo de investigación. Los integrantes de un grupo de investigación son miembros de la comunidad PUCP, es decir, estudiantes y profesores. Por el momento, no se está considerando que otro tipo de miembro de la comunidad PUCP, como el personal administrativo también forme parte de los grupos de investigación. Esto será evaluado en un futuro próximo o una vez que se cuente con una herramienta de soporte al proceso. Es importante resaltar que, si bien los estudiantes y profesores comparten ciertos atributos como el código PUCP, el nombre y el apellido paterno, también existen diferencias en ellos, pues los profesores tienen una dedicación, mientras que los estudiantes tienen un CRAEST.

Dado el caso de estudio mencionado, se le solicita realizar la programación de las funcionalidades de **registro, búsqueda, listado, y visualización de grupos de investigación**, utilizando el lenguaje de programación C# y ASP .NET con Web Forms. Para esto, se cuenta con un proyecto base, por lo que se solicita descargar desde PAIDEIA el proyecto y realizar la programación de estas funcionalidades. Los alumnos que quieran trabajar con **RadioButtonList** y **CheckBoxList** descarguen la segunda versión del proyecto base. Es importante resaltar que **NO** se requiere implementar la funcionalidad de **modificar y eliminar grupos de investigación**.

En la Figura 02 y Figura 03 se muestra el diseño de la interfaz gráfica web que se ha elaborado y que permitirá tanto el registro como la visualización de los datos de un grupo de investigación. Asuma que los profesores y los estudiantes ya se encuentran registrados en la base de datos. Lo único que tiene que registrar es su relación como integrantes de un grupo de investigación en conjunto con los datos de este. Con respecto al diseño de la base de datos, en PAIDEIA encontrará el Script SQL (**ScriptSQL_ExamenFinal.sql**) que permitirá generar todas las tablas y sus relaciones, así como los procedimientos almacenados que deben emplearse y algunos registros iniciales. Utilice el Script SQL que se ha proporcionado dentro de su instancia de base de datos MySQL de AWS Academy.

Para este ejercicio, no es necesario que realice validaciones. Asuma que el usuario ingresa correctamente toda la información.

Cuando registre un grupo de investigación, utilizará una única función en donde se registre tanto los datos del grupo de investigación como de la relación de integrantes asociados al mismo. No utilizará dos funciones/métodos separados para este propósito.

Por el contrario, cuando se muestre la información de un determinado grupo de investigación, una vez que el usuario ha seleccionado el grupo de investigación del cual se desea visualizar información, se llamará a un método de la clase GrupoInvestigacionMySQL "**GrupoInvestigacion obtenerPorID(int idGrupoInvestigacion)**" que traerá todos los datos del grupo y a otro método de la clase MiembroPUCPMySQL "**BindingList<MiembroPUCP> listarPorIdGrupoInvestigacion(int idGrupoInvestigacion)**" que traerá la lista de integrantes asociados a ese grupo. Es decir, en este caso, a diferencia del registro, se utilizarán tres funciones/métodos: (1) uno para listar todos los grupos de investigación registrados en la base de datos con información básica, (2) otro método para traer TODOS los datos de un determinado grupo de investigación que ha sido seleccionado por el usuario, porque se desea visualizar su información detallada, y (3) un último método para traer la lista de integrantes de aquel grupo de investigación que ha sido seleccionado por el usuario a fin de visualizar su información asociada.

Cuando muestre miembros PUCP o integrantes de un grupo de investigación en el componente **gridview**, los datos a mostrar son: (1) Tipo (la palabra "Profesor" o "Estudiante") dependiendo del tipo instancia, (2) el Código PUCP, (3) el nombre completo, (4) la dedicación o CRAEST dependiendo de si es Profesor o Estudiante. Programe un único método "**BindingList<MiembroPUCP> listarPorNombreCodigoPUCP(string nombreCodigoPUCP)**" para devolver y listar tanto profesores como estudiantes. Lea el campo "**fid_tipo_miembro_pucp**" que devuelve el procedimiento almacenado (**LISTAR_MIEMBROS_PUCP_X_NOMBRE_CODIGOPUCP**) e instancie (con la palabra reservada **new**) una variable **MiembroPUCP** de tipo **Profesor** o **Estudiante** según el carácter que se haya leído. Al momento de programar la lectura de datos, optimice el código de tal manera que no se repitan instrucciones de lectura que se ejecutarían tanto para la lectura de un **Profesor** como del **Estudiante**.

A continuación, se proporciona un video del funcionamiento solicitado: <https://youtu.be/79UnWhPXx9Q>

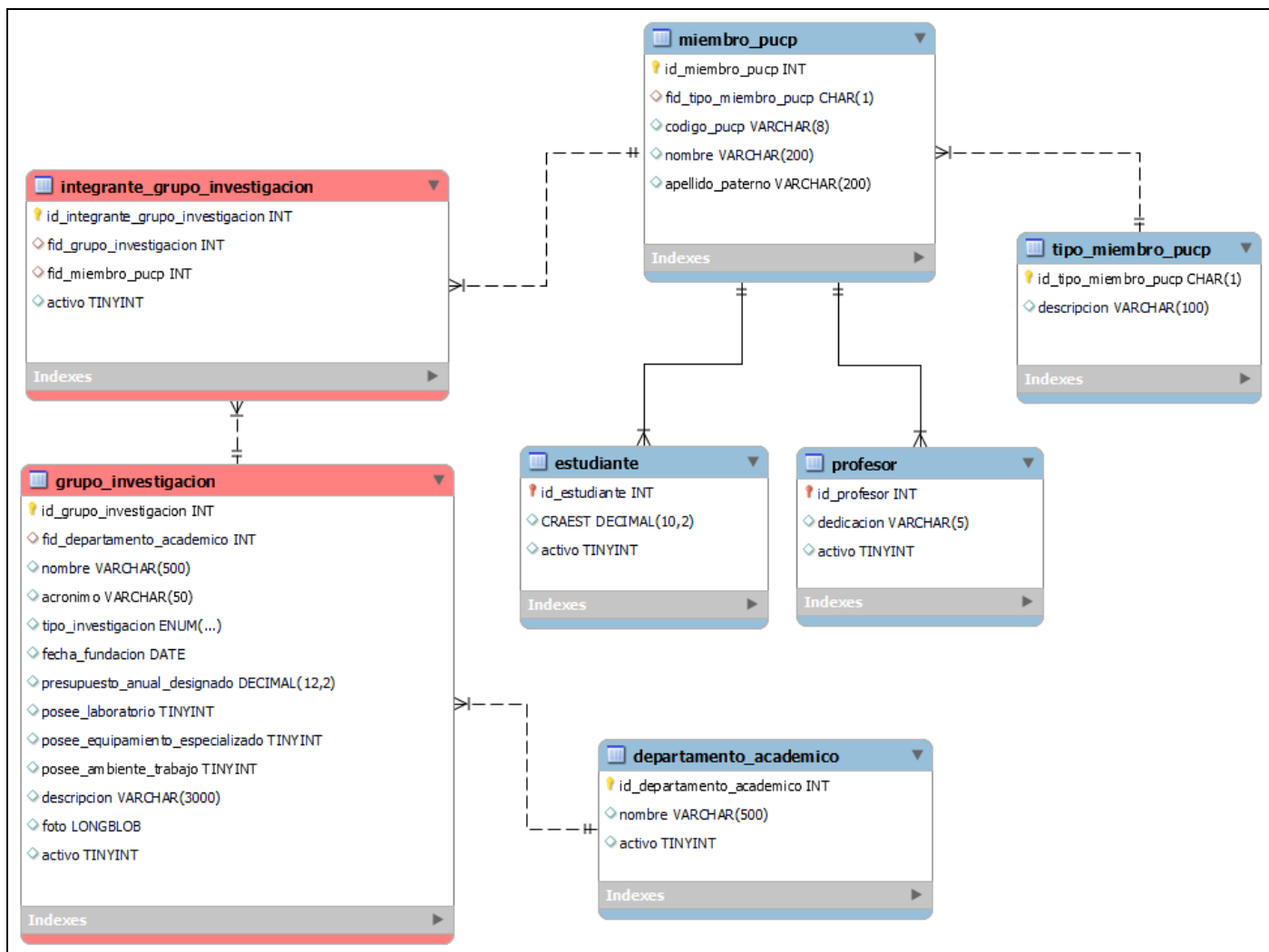


Figura 01. Diagrama EER de la estructura de base de datos.

localhost:62550/GestionarGruposInvestigacion.aspx

Registrar Grupo de Investigación

Datos Generales del Grupo

ID del Grupo: Acrónimo:

Nombre:

Pertenencia al Dpto:

Fecha de Fundación: Presupuesto Anual:

Tipo de Investigación: ☐ BÁSICA ☒ APLICADA

Infraestructura del Grupo: ☒ LABORATORIO ☐ EQUIPO ESPECIALIZADO ☒ ESPACIO DE TRABAJO

Descripción del Grupo:

Foto del Grupo: Sin archivos seleccionados

PUCP - Programación 3

Figura 02. Interfaz gráfica web para el registro/visualización de grupos de investigación – Parte 01.

01/12/2019 2500.00

Tipo de Investigación:
☐ BÁSICA ☒ APLICADA

Infraestructura del Grupo:
☒ LABORATORIO ☐ EQUIPO ESPECIALIZADO
☒ ESPACIO DE TRABAJO

Descripción del Grupo:
 El grupo DUXAIT es un equipo que tiene como objetivo la investigación, el desarrollo y la difusión de trabajos y proyectos de investigación que contribuyan en las áreas de la Interacción Humano-Computador (HCI) y en el Desarrollo Tecnológico.

Lista de Integrantes

Integrante:

Tipo	Codigo PUCP	Nombre Completo	Dedicacion/CRAEST
Profesor	20112728	FREDDY PAZ	TC
Profesor	02400045	HEIDER SANCHEZ	TPA
Estudiante	20245189	SEBASTIAN MIRANDA	89.77

Figura 03. Interfaz gráfica web para el registro/visualización de grupos de investigación – Parte 02.

No puede modificar las clases del proyecto **ResearchPUCPModel**. La única modificación que va a realizar en este proyecto es lo siguiente:

- Agregue una Propiedad (Getter) en la clase **MiembroPUCP** que devuelva el nombre completo.
- Agregue una Propiedad (Getter) en la clase **MiembroPUCP** que devuelva la cadena "Profesor" o "Estudiante", dependiendo del tipo de instancia de la variable de este tipo. Utilice la palabra reservada "is" (que tiene funcionamiento similar a "instanceof" en JAVA).
- Agregue una Propiedad (Getter) en la clase **MiembroPUCP** que devuelva la propiedad Dedicacion o CRAEST dependiendo del tipo de instancia de la variable, es decir, si es Profesor o Estudiante, respectivamente. Utilice la palabra reservada "is". Dado que la propiedad debe devolver el mismo tipo de dato, convierta el valor de CRAEST a **string** para que en cualquiera de los dos casos se devuelva un tipo **string**. Utilice el método `ToString("N2")` para devolver el valor del **double** convertido a una cadena con 2 decimales.
- Finalmente, utilice estas propiedades en los componentes **gridviews** para lograr la visualización solicitada en el enunciado.

Programa el evento **PageIndexChanging** en todos los **gridviews**.

Cuando muestre grupos de investigación en el **gridview**, los datos a mostrar son: Id, acrónimo y nombre.

Procedimientos almacenados a utilizar:

- LISTAR_DEPARTAMENTOS_ACADEMICOS_TODOS
- LISTAR_MIEMBROS_PUCP_X_NOMBRE_CODIGOPUCP
- INSERTAR_GRUPO_INVESTIGACION
- INSERTAR_INTEGRANTE_GRUPO_INVESTIGACION
- LISTAR_GRUPOS_INVESTIGACION_X_NOMBRE_ACRONIMO
- OBTENER_GRUPO_INVESTIGACION_X_ID
- LISTAR_INTEGRANTES_X_ID_GRUPO_INVESTIGACION

Coloque sus datos personales en las clases: **GestionarGruposInvestigacion** y **ListarGruposInvestigacion**.

Una vez que ha culminado su ejercicio, suba su proyecto a PAIDEIA. Se les recuerda que es responsabilidad de cada estudiante asegurarse de haber subido correctamente a PAIDEIA su propuesta de solución a los ejercicios propuestos. No se aceptarán reclamos por equivocaciones respecto a los archivos entregados por este medio. Tampoco se aceptarán archivos entregados por otros medios ni fuera del plazo establecido dentro del examen.

Para asignar la foto a la variable de tipo `byte[]` (debido a que ya se tiene implementada el método)

```
if (Session["foto"] != null)
    foto = (byte[]) Session["foto"];
grupoInvestigacion.Foto = foto;
```

Para enviar un arreglo de byte[] a un Command puede utilizar:

```
comando.Parameters.AddWithValue("_foto", grupoInvestigacion.Foto);
```

Para enviar un enumerado a un Command puede utilizar:

```
comando.Parameters.AddWithValue("_tipo_investigacion", grupoInvestigacion.TipoInvestigacion.ToString());
```

Para leer los datos de la foto puede utilizar:

```
grupoInvestigacion.Foto = (byte[]) lector["foto"];
```

Para leer los datos del enumerado puede utilizar:

```
grupoInvestigacion.TipoInvestigacion =  
(TipoInvestigacion)Enum.Parse(typeof(TipoInvestigacion), lector.GetString("tipo_investigacion"));
```

Para mostrar la imagen y la fecha de fundación por pantalla:

```
string base64String = Convert.ToBase64String(grupoInvestigacion.Foto);  
string imageUrl = "data:image/jpeg;base64," + base64String;  
imgFotoGrupo.ImageUrl = imageUrl;  
dtpFechaFundacion.Value = grupoInvestigacion.FechaFundacion.ToString("yyyy-MM-dd");
```

Rúbrica de calificación:

- No se considerará puntaje alguno si es que el aspecto a evaluar no funciona a nivel de interfaz gráfica, por lo que se solicita que, a medida que va avanzando el back-end, realice la programación del front-end. Asimismo, la corrección de los apartados será progresiva. Para obtener el puntaje de un apartado en específico, todos los anteriores aspectos a evaluar deben estar correctamente desarrollados.

A continuación, se citan algunos ejemplos:

- “no se considerará puntaje por la programación de la funcionalidad de registro, si es que para ese registro son necesarias algunas búsquedas y selecciones en el formulario y estas no se encuentran correctamente implementadas.”
- “no se considerará puntaje por programar la funcionalidad de búsqueda si es que el registro no se encuentra correctamente implementado”.
- “no se considerará puntaje por la programación de un formulario si es que los métodos de acceso a base de datos que son requeridos para su funcionamiento no se encuentran correctamente implementados”.
- El driver de conexión a base de datos debe estar referenciado con ruta relativa (no absoluta).
- Se descontarán puntos significativamente si es que existen errores de compilación.
- Se descontarán puntos por declaración de variables que admiten cualquier tipo de dato como "var" (C#).
- Se descontarán puntos si el código no está optimizado, por ejemplo, repetición innecesaria de código.
- Se descontarán puntos si es que no sigue las instrucciones en referencia a los métodos que deberían implementarse en cada clase.
- Se descontarán puntos por referencias innecesarias entre proyectos.
- Debe utilizar la estructura vista en clase, proyectos relacionados a: DBManager, a la capa de modelo, a la capa de acceso a base de datos y a la vista.

A. Implementación del **DBManager** (0.5 punto).

B. Implementación de las clases **DepartamentoAcademicoDAO** y **DepartamentoAcademicoMySQL** (2.0 puntos).

C. Implementación del listado de departamentos académicos a nivel de interfaz (0.5 punto).

D. Implementación del método "BindingList<MiembroPUCP> listarPorNombreCodigoPUCP(string nombreCodigoPUCP)" en las clases **MiembroPUCPDOA** y **MiembroPUCPMySQL** (2.5 puntos).

E. Implementación de la búsqueda, listado y selección de los miembros PUCP como integrantes del grupo de investigación que se va a registrar (considerando botones agregar y quitar). (3.0 puntos)

F. Implementación del método "int insertar(GrupoInvestigacion grupoInvestigacion)" en la clase **GrupoInvestigacionDAO** y **GrupoInvestigacionMySQL** (3.0 puntos).

G. Implementación de la funcionalidad de registro de un grupo de investigación a nivel de interfaz junto a los integrantes asociados al mismo (2.5 puntos).

H. Implementación del método "BindingList<GrupoInvestigacion> listarPorNombreAcronimo(string nombreAcronimo)" en la clase **GrupoInvestigacionDAO** y **GrupoInvestigacionMySQL** (1.5 puntos).

I. Implementación del método "GrupoInvestigacion obtenerPorID(int idGrupoInvestigacion)" en la clase **GrupoInvestigacionDAO** y **GrupoInvestigacionMySQL** (1.5 puntos).

J. Implementación del método "BindingList<MiembroPUCP> listarPorIdGrupoInvestigacion(int idGrupoInvestigacion)" en la clase **MiembroPUCPDOA** y **MiembroPUCPMySQL** (1.0 puntos).

K. Implementación de la funcionalidad de búsqueda, listado y visualización de grupos de investigación a nivel de interfaz. (1.0 punto).

Profesores del curso:

Dr. Freddy Paz

Dr. Heider Sánchez

San Miguel, 16 de mayo de 2024