

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA**

**LENGUAJE DE PROGRAMACIÓN 2**

**Examen 1  
(Primer semestre 2022)**

**Indicaciones generales:**

- Duración: 2 horas.
- Materiales o equipos a utilizar:  
ESTÁ PERMITIDO USAR APUNTES, DIAPOSITIVAS DE CLASE E INTERNET.  
El Internet será utilizado solo para consultar documentación de programación.  
Está prohibido toda forma de conversación con otras personas.  
Se les recuerda que, de acuerdo con el reglamento disciplinario de nuestra institución, constituye una falta grave copiar del trabajo realizado por otro estudiante o cometer plagio para el desarrollo de este examen.
- **El uso apropiado de buenas prácticas de programación (sobre todo aquellas vistas en clase) será tomado en cuenta para la calificación.**

Puntaje total: 20 puntos

**Parte Práctica:** (20 puntos)

El Departamento de Ingeniería de nuestra universidad desea contar con un software que le permita gestionar la información relacionada a los cursos que ofrece a los estudiantes en los distintos semestres. En este sentido, se le ha solicitado que realice la programación de las funcionalidades de “registro, listado y selección” de cursos. Para esto, se cuenta con un diseño previo de la interfaz (Figura 01 y Figura 02). Se le solicita descargar el proyecto base en C# o JAVA de PAIDEIA y realizar la programación de estas funcionalidades.  
Para este caso específico, **NO** se requiere implementar las funcionalidades de modificar y eliminar curso.

**Formulario de Gestión de Cursos**

Nuevo Guardar Buscar Cancelar

**Gestión de Cursos del Departamento de Ingeniería - PUCP**

**Datos Generales** Docentes Asignados

Nombre del Curso: LENGUAJE DE PROGRAMACIÓN 2

Clave del Curso: INF282

Semestre: 2022 - 1

Modalidad del Curso: ☒ Virtual ☐ Presencial ☐ Semi-presencial

Especialidad: INGENIERIA INFORMATICA

Requerimientos: ☒ Computadora ☐ Dispositivo Móvil ☐ Dispositivo Especializado

Sumilla: Programación bajo sistemas operativos de ambiente gráfico, orientados a objetos, con interfaces gráficas de usuario, concurrentes y distribuidos.

Silabo del curso: C:\Users\Freddy Jr\Downloads\INF282-2022-1-SILABO.PDF **Subir** Descargar

**Lenguaje de Programación 2 (INF282)**

PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU

Figura 01. Interfaz gráfica – Datos Generales del Curso

Un curso tiene un identificador único, un nombre, una clave de curso conformada por seis caracteres, el semestre en el que se dicta el curso, el tipo de modalidad que puede ser virtual, presencial o semipresencial, la especialidad a la

que pertenece el curso, los requerimientos del mismo que pueden ser computadora, dispositivo móvil y/o dispositivo especializado, una sumilla descriptiva, el sílabo en formato PDF y una foto representativa (Ver Figura 01). Asimismo, se requiere registrar a los docentes que son asignados al curso (Ver Figura 02). Para ello, es necesario resaltar que el Departamento maneja información tanto de profesores PUCP como profesores del extranjero que también podrían incorporarse como docentes del curso. En el caso de un profesor del extranjero se requiere visualizar además de los campos comunes que aparecen en la figura las siglas de la afiliación (universidad) a la cual pertenecen. En el caso de los profesores PUCP se desea visualizar además de los campos comunes, su categoría (ver Figura 02).

**Formulario de Gestión de Cursos**

Nuevo Guardar Buscar Cancelar

**Gestión de Cursos del Departamento de Ingeniería - PUCP**

Datos Generales **Docentes Asignados**

Código PUCP:  ...

Nombre Completo:  + -

	Código PUCP	Nombre Completo	Categoría/Filiación
▶	20112728	FREDDY PAZ	ASOCIADO
	0000982X	RONY CUEVA	AUXILIAR
	E1010271	ANTONI GRANOLLERS	UDLL
	E4092031	CLARISSE DE SOUZA	PUC-RIO
	E0918710	JUAN GONZALES	BUAP

Figura 02. Interfaz gráfica – Docentes Asignados al Curso

Se ha diseñado la base de datos que se estaría utilizando (Figura 03). En PAIDEIA encontrará un Script que permitirá generar todas las tablas y sus relaciones, así como los procedimientos almacenados que deben emplearse y algunos registros. Utilice el script SQL que se ha proporcionado en PAIDEIA dentro de su instancia de base de datos MySQL de AWS Academy.

**Se solicita considerar validaciones únicamente al momento de asignar o eliminar docentes al curso que está registrando.**

Cuando liste los docentes, desarrollará una única función llamada “listarPorNombreCodigo” que permitirá devolver una lista de objetos de tipo “Docente” donde se encuentren tanto docentes PUCP como docentes externos. Identifique primero el tipo de docente a través de la columna ‘fid\_tipo\_docente’ y luego instancie un objeto de tipo DocentePUCP o DocenteExtranjero dependiendo del tipo. Luego asigne los datos requeridos según sea el caso y finalmente agregue el objeto a la lista de objetos de tipo Docente. Por otro lado, cuando registre un curso, utilizará una única función en donde se registre tanto el curso como los docentes asociados al mismo. No utilizará dos funciones separadas para este propósito. Finalmente, cuando se muestre la información de un determinado curso, una vez que el usuario ha seleccionado el curso del cual desea visualizar la información, se llamará a un método de la clase DocenteMySQL “listarPorIdCurso” y se traerá la lista de docentes asociados a ese curso seleccionado. No está permitido modificar las clases que corresponden a la capa del modelo.

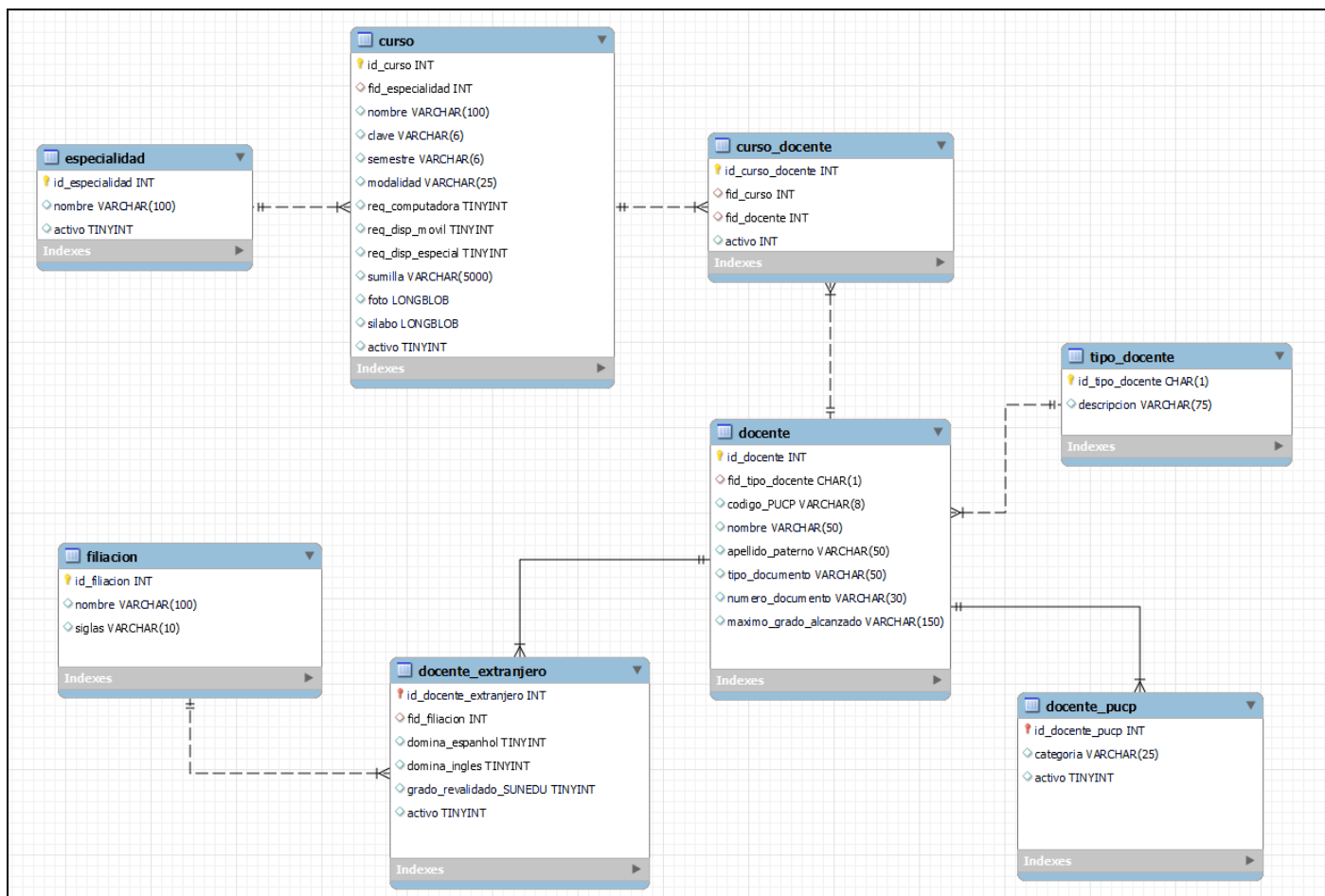


Figura 03. Diagrama ERR de la base de datos

A continuación, se adjunta un video sobre el funcionamiento esperado: <https://youtu.be/uZ7VqsJzVto>

Procedimientos almacenados que debe utilizar:

- LISTAR\_ESPECIALIDADES\_TODAS
- LISTAR\_DOCENTES\_POR\_NOMBRE\_O\_CODIGO
- INSERTAR\_CURSO
- INSERTAR\_CURSO\_DOCENTE
- LISTAR\_CURSOS\_POR\_NOMBRE\_O\_CLAVE
- LISTAR\_DOCENTES\_POR\_ID\_CURSO

Una vez culminado el ejercicio, suba su proyecto a PAIDEIA.

### 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 proyecto EduSoftDBManager. (1 punto).
- B. Implementación de las clases EspecialidadDAO y EspecialidadMySQL. (1 punto)
- C. Implementación de la búsqueda, listado y selección de especialidades a nivel de interfaz. (1 punto)
- D. Implementación del método "listarPorNombreCodigo" en las clases DocenteDAO y DocenteMySQL. (3 puntos)
- E. Implementación de la búsqueda, listado y selección de docentes a nivel de interfaz (considerando botones agregar y quitar). (3 puntos)
- F. Implementación del método "insertar" en las clases CursoDAO y CursoMySQL. (3 puntos)
- G. Implementación de la funcionalidad de registro de curso a nivel de interfaz. (2 puntos)
- H. Implementación del método "listarPorNombreClave" en las clases CursoDAO y CursoMySQL. (2 puntos)
- I. Implementación del método "listarPorIdCurso" en las clases DocenteDAO y DocenteMySQL. (2 puntos)
- J. Implementación de la funcionalidad de búsqueda, listado y selección de cursos a nivel de interfaz. (2 puntos)

Profesor del curso: **Dr. Freddy Paz**

San Miguel, 16 de mayo del 2022

## ANOTACIONES DE PROGRAMACIÓN

**(En el caso de C#)** Para obtener el año de un datetimepicker utilizar:

```
dtpAnho.Value.Year.ToString();
```

Para convertir de String (yyyy) a Datetime utilizar:

```
dtpAnho.Value = DateTime.ParseExact(_variableQueContieneAnho, "yyyy", null);
```

**(En el caso de JAVA)** Para obtener el año de un JYearChooser utilizar:

```
String.valueOf(dtpAnho.getValue());
```

Para convertir de String (yyyy) a int utilizar:

```
dtpAnho.setValue(Integer.parseInt(variableQueContieneAnho));
```

**(En el caso de JAVA)** Para enviar un arreglo de bytes a un CallableStatement puede utilizar:

```
cs.setBytes("_foto", curso.getFoto());  
cs.setBytes("_silabo ", curso.getSilabo());
```

En caso desee recibir bytes:

```
curso.setFoto(rs.getBytes("foto"));
```

**(En el caso de C#)** Para enviar un arreglo de bytes a un Command puede utilizar:

```
comando.Parameters.AddWithValue("_foto", curso.Foto);  
comando.Parameters.AddWithValue("_silabo", curso.Silabo);
```

En caso desee recibir bytes:

```
if (!lector.IsDBNull(lector.GetOrdinal("foto"))) curso.Foto = (byte[])lector["foto"];
```

**(En el caso de JAVA)** Para asignar la foto o archivo a la variable de tipo byte[]:

```
curso.setFoto(Files.readAllBytes(archivoFoto.toPath()));
```

**(En el caso de C#)** Para asignar la foto o archivo a la variable de tipo byte[]:

```
FileStream fs = new FileStream(_rutaFotoPortada, FileMode.Open, FileAccess.Read);  
BinaryReader br = new BinaryReader(fs);  
_curso.Foto = br.ReadBytes((int)fs.Length);  
fs.Close();
```

**(En el caso de JAVA)** Para mostrar la imagen desde el arreglo:

```
Image img = new ImageIcon(curso.getFoto()).getImage();  
Image newImg = img.getScaledInstance(pbFoto.getWidth(), pbFoto.getHeight(),  
java.awt.Image.SCALE_SMOOTH);  
pbFoto.setIcon(new ImageIcon(newImg));
```

**(En el caso de C#)** Para mostrar la imagen desde el arreglo:

```
MemoryStream ms = new MemoryStream(_curso.Foto);  
pbFoto.Image = new Bitmap(ms);
```

**(En el caso de JAVA)** Para escribir un archivo desde un arreglo de byte[]:

```
FileOutputStream fos = new FileOutputStream(archivoGenerar);  
fos.write(this.curso.getSilabo());  
fos.flush();  
fos.close();
```

**(En el caso de C#)** Para escribir un archivo desde un arreglo de byte[]:

```
File.WriteAllBytes(archivoGenerar, this._curso.Silabo);
```