

# PROYECTO DE PROGRAMACIÓN CON BASES DE DATOS – GESTIÓN TARJETAS DE CRÉDITO

## Descripción del proyecto

---

En este proyecto **utilizarás como base el proyecto de Gestión de Tarjetas de crédito en modo gráfico** realizado en las actividades de la unidad anterior. En esa actividad se utilizaban ficheros para cargar y guardar los datos de la aplicación. Ahora, deberás modificarla para que haga uso de bases de datos. La descripción completa y detallada de la implementación de este proyecto se encuentra en las siguientes secciones.

Debes aplicar todas las buenas prácticas aprendidas, entre otras:

- Implementar los métodos de las clases realizando control de errores.
- Implementar los constructores y métodos de la clase reutilizando todo el código posible.
- Implementar las clases sin modificar la interfaz pública de la misma.
- Validar todos los datos de entrada, tanto los introducidos por teclado por el usuario como los parámetros de las funciones.
- Evitar código duplicado y reutilizar código mediante funciones.
- Nombrar las variables, clases y métodos de acuerdo a las convenciones del lenguaje, y usar identificadores descriptivos y representativos de lo que almacena o realiza la variable/función.
- Documentar las clases con comentarios Javadoc.
- Seguir las especificaciones descritas en este guion sin cambiarlas, salvo justificación y previa consulta con el profesor.

**Importante: No se corregirá ningún proyecto en el que haya errores de compilación.**

El objetivo de este proyecto es trabajar con las librerías disponibles en Java para conectar con bases de datos relacionales (JDBC) y con bases de datos orientadas a objetos u objeto-relacional (JPA). Para ello, se utilizará la aplicación Java en modo gráfico creada para el proyecto de Gestión de Tarjetas de crédito. La aplicación gráfica debe tener implementada toda la funcionalidad propia de la aplicación, por lo que se debe añadir la gestión de la información almacenada en las bases de datos.

La tarea se califica entre 0 y 10 puntos desglosados de la siguiente manera:

- **Aplicación con JDBC (5 puntos).** Se debe implementar la conexión a una base de datos Java Derby mediante las librerías de Java de JDBC. Deben existir toda la funcionalidad necesaria para implementar el CRUD de la base de datos.
- **Aplicación con JPA (5 puntos).** Se debe implementar la conexión a una base de datos Java Derby mediante las librerías de Java de JPA. Deben existir toda la funcionalidad necesaria para implementar el CRUD de la base de datos.

## Entrega de la tarea

---

El proyecto se desarrollará con el IDE Netbeans. El nombre del proyecto y de las clases deben llamarse de la forma indicada en la descripción de la tarea. Se comprimirá la carpeta del proyecto Netbeans en un único fichero en formato .ZIP o 7z que se subirá al buzón de la tarea en la plataforma Moodle.

El archivo se nombrará siguiendo las siguientes pautas:

**Apellido1\_Apellido2\_Nombre\_PROG\_ProyectoBasesDatos.zip**

**El proyecto será revisado y defendido en clase antes de poderse enviar a Moodle.**

## Resultados de aprendizaje y criterios de evaluación relacionados

---

En esta actividad se evalúan los siguientes resultados de aprendizaje con los criterios de evaluación que se relacionan para cada uno de ellos:

- RA 8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.
  - a) Se han identificado las características de las bases de datos orientadas a objetos.
  - b) Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.
  - c) Se han instalado sistemas gestores de bases de datos orientados a objetos.
  - d) Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.
  - e) Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.
  - f) Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.
  - g) Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.
  - h) Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.
- RA 9. Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y la consistencia de los datos.

- a) Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales.
- b) Se han programado conexiones con bases de datos.
- c) Se ha escrito código para almacenar información en bases de datos.
- d) Se han creado programas para recuperar y mostrar información almacenada en bases de datos.
- e) Se han efectuado borrados y modificaciones sobre la información almacenada.
- f) Se han creado aplicaciones que ejecuten consultas sobre bases de datos.
- g) Se han creado aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales.

## Rúbrica de evaluación

---

Cada apartado puntuable del proyecto se valorará con la siguiente rúbrica.

#	Criterio	Porcentaje
1	El programa/función implementado no cumple los requisitos, no soluciona de forma algorítmica el ejercicio o las soluciones obtenidas por el programa no son las esperadas, el código no compila o contiene errores.	0%
2	El programa/función implementado soluciona de forma algorítmica el ejercicio pero falla con datos de entrada no permitidos.	25%
3	El programa/función implementado tiene fallos inesperados en situaciones específicas o concretas, es decir, falla para un determinado caso o valor de entrada, pero en general el resultado obtenido es válido.	50%
4	El programa/función implementado cumple los requerimientos pero: <ul style="list-style-type: none"><li>• El código no es legible o no está bien estructurado.</li></ul>	75%
5	El programa/función implementado se ajusta perfectamente a la especificación: <ul style="list-style-type: none"><li>• Se validan los datos de entrada.</li><li>• El resultado obtenido es válido para cualquier dato de entrada.</li><li>• El código es modular y se emplean funciones/métodos adecuadamente.</li><li>• El código es legible y usa comentarios relevantes y/o Javadoc.</li></ul>	100%

