

NOMBRE: \_\_\_\_\_

## Instrucciones:

- 1.- El examen se debe realizar en la cuenta de usuario Examen2, cuya contraseña es Vamos!
- 2.- Antes de empezar, asegúrate que en la máquina no hay ninguna credencial de Github en Windows. Para ello ve al administrador de credenciales de Windows y bórrala si existiera. También borra el archivo D:\users\examen2\.gitconfig
- 3.- Usa la navegación privada en el navegador, para loguearte en github. Por ejemplo en Firefox se abre con control + mayus + p
- 4.- Puedes encontrar los JavaDoc y los apuntes en la Url: <http://192.168.50.250/programacion>
- 5.- Dispones de 10 minutos para descargar las librerías necesarias.
- 6.- Recuerda, no está permitido el uso de internet o la IA, su uso conlleva la calificación de 0 en la prueba.
- 7.- No copies, si se te pilla copiando tendrás un 0 en la prueba
- 8.- No pierdas el tiempo.
- 9.- El examen se debe entregar en github classroom
- 10.- La entrega se cierra a las **21:20**
- 11.- El examen se deberá entregar en tiempo y forma. La no entrega, sea por la circunstancia que sea, imposibilita la calificación del mismo. Lo que implica una calificación de 0.

URL de la tarea de github classroom:

<https://classroom.github.com/a/VvVzFeeo>

### Ejercicio 1:

**1,25 puntos**

Desarrolla un programa que se conecte con el SGBD MySQL instalado en la máquina, debe usar el usuario root, a pesar de que como se ha explicado es una mala práctica.

Este programa deberá ver si en el SGDB existe la base de datos: “supernivel” y si no existe debe crearla. Si al contrario la base de datos existe, el programa deberá indicarnos que no se ha creado la BBDD ya que existía previamente.

#### Ejemplo de ejecución:

En la primera ejecución:

```
Se ha creado la BBDD "supernivel"
```

En ejecuciones posteriores:

```
No se ha creado la base de datos "supernivel" ya que existía previamente.
```

**Ejercicio 2:****2 puntos**

En este segundo ejercicio deberás deserializar un archivo json e incluir todo su contenido en una base de datos que debes crear.

Instrucciones:

Como sabes, un json se asemeja a una clase java, el contenido entre llaves va unido a los atributos de la clase:

```
{
  "localidad": "Almunia de Doña Godina (La)",
  "tasa_graduacion": 33.333333333333,
  "media_cursos_matriculados": 3.25,
  "tipo_estudio": "Grado",
  "estudio": "Grado: Arquitectura Técnica",
  "sexo": "Hombres",
  "tasa_abandono_inicial": 33.333333333333,
  "alumnos_graduados": 2,
  "curso_academico": 2018,
  "alumnos_graduados_en_tiempo": 1,
  "tasa_abandono": 66.666666666667,
  "alumnos_interrumpen_estudios": 2,
  "alumnos_interrumpen_est_ano1": 1
},
```

```
public class Estudios {

    private String localidad;
    private float tasa_graduacion;
    private float media_cursos_matriculados;
    private String tipo_estudio;
    private String estudio;
    private String sexo;
    private float tasa_abandono_inicial;
    private int alumnos_graduados;
    private int curso_academico;
    private int alumnos_graduados_en_tiempo;
    private float tasa_abandono;
    private int alumnos_interrumpen_estudios;
    private int alumnos_interrumpen_est_ano1;
}
```

Descarga <http://192.168.50.250/programacion/ej2.zip>, descomprime y encontrarás un archivo Json (Uz.json) con datos de las carreras universitarias de la Universidad de Zaragoza y un archivo java (Estudios.java), analiza el json y el archivo java, verás que el archivo java tiene todos los atributos creados para la clase Estudios (coincide con la imagen superior).

Abre la consola de administración de Mysql y manualmente, no desde java, crea la base de datos "openData" y la tabla "datos". Guarda en un script sql los comandos necesarios para crear la BBDD y la tabla, debes de entregar dicho archivo, colócalo en la raíz del proyecto y se subirá a github.

Realiza un pequeño programa Java que deserialice el json e inserta los datos en la base de datos "openData", creada manualmente. Las inserciones las debes realizar usando PreparedStatement.

Para corregir este ejercicio, se lanzará el script SQL facilitado, y se ejecutará el programa. Después se verificarán las inserciones realizadas en la BBDD:

```
use openData;
select * from datos;
```

**Ejercicio 3:****6,75 puntos**Descarga <http://192.168.50.250/programacion/ej3.zip>

En este zip encontrarás un archivo sql y programa funcional que debes modificar. Ejecuta y entiende dicho programa.

Crea la base datos supermercado, e importa el script sql.

Se desea modificar dicho programa para que cuando arranque cargue desde la base de datos los datos existentes, y acabe guardando en la misma, para ello se pide:

- Cargar la lista de Clientes (tabla clientes) (0,5 puntos)
  - Cargar la lista de Productos (tabla productos) (0,25 puntos)
  - Incorporar una opción al programa principal que solicite un importe al usuario y que muestre los productos que tienen un importe superior al introducido (1 punto)
  - Cargar la lista de Carros. Para ello, tenéis la tabla carro en la BBDD que asocia un cliente con los productos que tiene en su carro. Vamos a suponer que un cliente no puede tener más de un carro. (3 puntos)
  - Implementa la opción de añadir un producto, guardando el producto dado de alta en la BBDD. (0.25 puntos)
  - Implementa la opción añadir un cliente, guardando el producto dado de alta en la BBDD. (0.25 puntos)
  - Implementa la opción “Realiza una compra” guardando los datos en la base de datos. (1,5 puntos)
- 

*Corrección del examen. Los programas no tienen una única manera de ser codificados correctamente por lo que la corrección será la siguiente:*

- *Para puntuar el programa debe funcionar correctamente para datos válidos y hacer exactamente lo que se pide. Esto puntuará el 75% de la nota atribuida a cada apartado. En caso contrario el apartado no puntúa en absoluto.*
- *Uso adecuado de tipos, nombres, limpieza y claridad del código. 25%*