

## PROGRAMACIÓ

### Trabajo 4: Saber y Ganar 2.0

Programación  
CFGS DAW

Autores:

Lionel Tarazón [lionel.tarazon@ceedcv.es](mailto:lionel.tarazon@ceedcv.es)

Encarni Serrano [encarni.serrano@ceedcv.es](mailto:encarni.serrano@ceedcv.es)

2019/2020

#### Licencia



**Reconocimiento – NoComercial – CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

# Saber y Ganar 2.0

*Con Interfaz Gráfica y Base de Datos ;-)*



Fuente de la imagen: <https://www.rtve.es>

## ÍNDICE

<b>1. OBJETIVO</b>	<b>4</b>
<b>2. DESCRIPCIÓN DEL TRABAJO</b>	<b>4</b>
2.1 Interfaz gráfica	4
2.2 Base de datos	4
<b>3. CRITERIOS DE CALIFICACIÓN</b>	<b>5</b>
3.1 Requisitos mínimos para aprobar	5
3.2 Puntuación	5
3.3 Consideraciones generales	5
<b>4. ENTREGA</b>	<b>5</b>



**Lee atentamente** todo el documento, varias veces si es necesario, **subraya** o **toma notas** de los aspectos más importantes, y tenlo a mano para consultarlo durante las fases de diseño y desarrollo.

## 1. OBJETIVO

Hacer una versión mejorada del anterior trabajo, con interfaz gráfica de usuario y base de datos.

## 2. DESCRIPCIÓN DEL TRABAJO

Esta segunda versión del videojuego deberá incluir las mismas funcionalidades descritas en el trabajo 3 (jugar partidas, usuarios humanos y CPU, distintos tipos de preguntas, ranking, histórico de partidas, gestión de usuarios, etc.) añadiendo interfaz gráfica y base de datos tal y como se describe más abajo.

**NOTA:** Podéis utilizar vuestro propio código del Trabajo 3. Seguramente podréis aprovechar bastantes cosas, aunque otras habrá que cambiarlas o rehacerlas. Mejorar un proyecto software y “enfrentaros” a vuestro propio código es una experiencia de la que se aprende mucho.

### 2.1 Interfaz gráfica

Toda la interacción con el usuario se realizará mediante interfaz gráfica.

Una posibilidad es hacer una sola ventana (JFrame) con distintas pestañas (JTabbedPane), cada una de ellas dedicada a las distintas funcionalidades principales (jugar partida, ranking, etc.).

Otra opción es desarrollarlo mediante distintas ventanas (un JFrame por funcionalidad) de modo que haya una ventana principal con botones (a modo de menú principal) que permitan acceder a las otras ventanas. Algunos detalles sobre esta segunda opción no están en la teoría y tendréis que investigarlas por vuestra cuenta, pero tendrá un acabado más profesional y se valorará positivamente en la nota.

En cualquier caso queda a discreción del alumno diseñar la interfaz y sus componentes de la forma en que considere más apropiada. En la medida de la posible, ésta deberá ser intuitiva y sencilla de utilizar para el usuario.

### 2.2 Base de datos

Los datos que en la primera versión estaban almacenados en ficheros ahora deberán estar en una base de datos tipo MySQL. Deberá contener las tablas necesarias para almacenar la información de usuarios, preguntas, partidas que se han jugado y el ranking. Queda a discreción del alumno el diseño de la base de datos que considere más apropiada.

Como la información de las preguntas deberá estar en la base de datos, tendréis que desarrollar un pequeño programa auxiliar (a parte del programa principal) que vuelque la información de los archivos ‘diccionario.txt’ e ‘ingles.txt’ a la base de datos (este volcado solo es necesario hacerlo una vez). Este pequeño programa también deberá entregarse.

### 3. CRITERIOS DE CALIFICACIÓN

#### 3.1 Requisitos mínimos para aprobar

- El **código compila** y se pueden **jugar partidas**.
- **Interfaz gráfica de usuario**.
- Acceso a **base de datos**.

#### 3.2 Puntuación

- **Diagrama de clases UML (1 punto)**: Un diseño apropiado mediante POO.
- **Base de datos y programa de volcado de datos (1 punto)**: Diseño de la base de datos y el programa utilizado para volcar las preguntas de ficheros a base de datos.
- **Jugar partida (6 puntos)**: Se pueden jugar partidas entre varios jugadores y distintos tipos de preguntas tal y como se especifica en los requisitos.
- **Ranking (0,5 puntos)**: Se puede visualizar el ranking de usuarios.
- **Histórico de partidas (0,5 puntos)**: El programa registra el histórico de partidas que se han jugado y permite visualizarlo.
- **Jugadores registrados / gestión de usuarios (1 punto)**: El programa incluye un registro de usuarios y permite gestionarlos (ver listado de usuarios, añadir usuario y eliminar usuario).

Como puede observarse, la funcionalidad más importante es la de jugar partidas, que junto con el diagrama de clases y lo relacionado con la base de datos permite obtener una nota máxima de 8 (siempre y cuando incluya interfaz gráfica y base de datos). Las otras tres funcionalidades (ranking, histórico y gestión de usuarios) permiten obtener hasta 2 puntos más, hasta una total de 10.

#### 3.3 Consideraciones generales

- **Diseño POO apropiado**: Clases, atributos, métodos, visibilidad, abstracción, herencia, etc.
- Manejar las posibles **excepciones**.
- Uso de **estructuras de datos** (Arrays y/o ArrayList y ResultSet).
- Buena **estructuración y legibilidad del código**.
- **Comentarios explicativos** (un breve comentario por clase y método).

### 4. ENTREGA

Deberás entregar por el aula virtual:

1. **Diagrama de clases UML** actualizado (formato PDF o imagen).
2. **Base de datos** exportada en un archivo sql.
3. **Programa auxiliar de volcado de preguntas a la base de datos** (proyecto completo en zip).
4. **Programa principal Saber y Ganar 2.0** (proyecto completo en zip).

La fecha límite de entrega es el **viernes 22 de mayo a las 23:55h**.