Luis Sampayo, Mikel, Iñaki, Carlos Larrondo

Unidad de Trabajo 6

Backend para CrimeBook

PLAN DE PRUEBAS Dwes06

Desarrollo Web en Entorno Servidor (DAW\_DWES)

[1. Introducción 2](#_Toc38210697)

[2. Objetivos del trabajo 2](#_Toc38210698)

[3. Diseño de la aplicación 2](#_Toc38210699)

[3.1. DB 2](#_Toc38210700)

[3.2. Equipo 2](#_Toc38210701)

[3.3. Partida 4](#_Toc38210702)

[3.4. Resolución 6](#_Toc38210703)

[3.5. prueba 8](#_Toc38210704)

[4. Plan de pruebas 8](#_Toc38210705)

[5. Reparto del trabajo 8](#_Toc38210706)

[6. Tabla de ilustraciones 9](#_Toc38210707)

# Introducción

Este trabajo se enmarca dentro del módulo Desarrollo Web en Entorno Servidor y da respuesta a la parte práctica de la Unidad de Trabajo número 6. El trabajo consiste en realizar el backend de una aplicación pensada para poder gestionar partidas de un juego que se desarrollará utilizando dispositivos Smart-phone.

# Objetivos del trabajo

El objetivo principal del trabajo consiste en la implementación de un ***backend*** para gestionar la aplicación web *“CrimeBook”* mediante un grupo de trabajo de 4 personas.

Para ello vamos a desarrollar los siguientes objetivos específicos:

* Analizar los requisitos y el material de partida
* Diseñar la aplicación
* Establecer políticas de trabajo en grupo
* Programar la aplicación
* Probar y verificar la aplicación
* Preparar la presentación del trabajo (Sujeto a la evolución de la pandemia de COVID19)

La concreción de los objetivos específicos la alcanzaremos con los siguientes objetivos operativos:

* Estudiar el enunciado e interpretar el mismo dentro de lo posible
* Abrir el material de partida entregado para ver el punto de partida, prestando especial atención a la BBDD con los registros necesarios
* Listar las funcionalidades que requiere la aplicación y su relación con las diferentes tablas de la BBDD entregadas
* Establecer los cambios necesarios en la BBDD para conseguir una compatibilidad con las funcionalidades de la aplicación
* Adjudicar el rol de scrum-master dentro de la técnica de gestión de trabajo en grupo scrum
* Distribuir el trabajo por parte del scrum master
* Establecer el sistema para gestionar las diferentes aportaciones del grupo mediante la herramienta github
* Programar el código PHP, HTML, CSS, SMARTY, por parte de cada componente del equipo
* Fusionar las partes en un todo solucionando problemas de compatibilidad
* Establecer el programa de pruebas
* Realizar el programa de pruebas
* Desarrollar la memoria del plan de pruebas
* Trabajar la presentación (Sujeto a la evolución de la pandemia de COVID19)

# Diseño de la aplicación

Las clases que forman parte de la aplicación y los cambios relacionados con ellas en la base de datos son:

## DB

En esta clase se hacen todas las consultas a la BBDD, lecturas, escrituras, borrados y modificaciones.

## Equipo

Sirve para gestionar los datos que se encuentran en la tabla equipos de la BBDD. En negrita las claves primarias. Tiene 5 atributos privados que coinciden con los campos de dicha tabla:

**id**: Es un identificador único para cada equipo

codigo: Es un valor aleatorio entregado a cada equipo en el momento de apuntarse

nombre: Nombre del equipo

tiempo: Tiempo que ha dedicado ese equipo a una partida

idPartida: Partida en la que participa el equipo

Como métodos tenemos en primer lugar el constructor que genera un nuevo objeto con todos y cada uno de los atributos inicializados con los valores que pasa una variable de tipo columna con un registro completo obtenido de la tabla equipo de la BBDD.

Además existen los getters necesarios para poder visualizar todos y cada uno de los atributos.

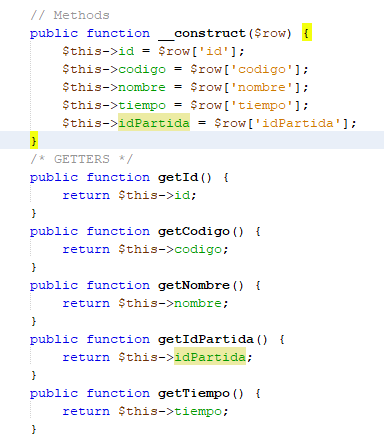


Ilustración 1: Clase Equipo

En cuanto a las modificaciones en la tabla estas son

1. El ID se ha puesto autoincrementable, para evitar repeticiones y librar al usuario de la aplicación de esa responsabilidad.
2. Para conseguir que el código sea único y generado por el sistema se ha diseñado para que coja una parte de los microsegundos del sistema <https://www.php.net/manual/es/function.microtime.php>

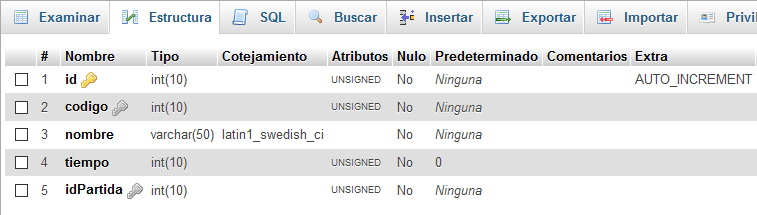


Ilustración 2: Estructura de la tabla Equipos



Ilustración 3: Datos de prueba de la tabla Equipos

## Partida

Sirve para gestionar los datos que se encuentran en la tabla partidas de la BBDD. En negrita las claves primarias. Tiene 8 atributos privados que coinciden con los campos de dicha tabla:

**id**: Es un identificador único para cada partida

nombre: Nombre de la partida

fechaCreacion: Fecha de creación de la partida

duración: Duración de la partida

fechaInicio: Fecha en que se empieza a jugar la partida

idJuego: Para saber cada partida a qué juego pertenece

userName: Nombre del usuario

finalizada: Para saber si se ha finalizado la partida o no

Como métodos tenemos en primer lugar el constructor que genera un nuevo objeto con todos y cada uno de los atributos inicializados con los valores que pasa una variable de tipo columna con un registro completo obtenido de la tabla partidas de la BBDD.

Además existen los getters necesarios para poder visualizar todos y cada uno de los atributos.

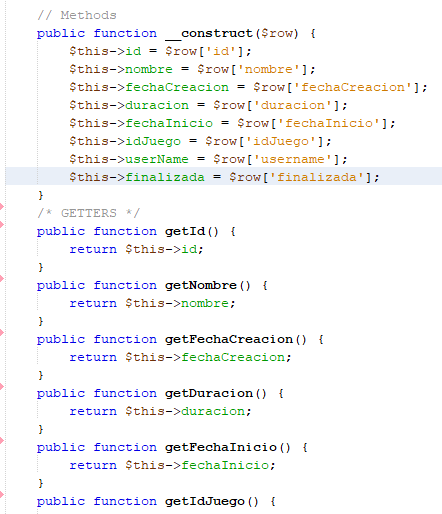


Ilustración 4: Clase Partida

En cuanto a las modificaciones en la tabla estas son:

1. En la tabla partidas se ha creado un campo para saber si la partida ha finalizado o no. Se podría sumar a la fecha de inicio la duración y comparar con la fecha actual del sistema pero como la duración es un número sin contextualizar del que no se sabe si son horas, días o segundos, no está claro como hacerlo.
2. El ID de la partida se ha puesto autoincrementable para evitar repeticiones y librar al usuario de la aplicación de esa responsabilidad.
3. El nombre se ha completado con la duración. Porque, los nombres deben ser únicos, además el enunciado pide que se haga una “copia” de una partida, a la que solo se le cambie la duración, cosa claramente incompatible con tener nombres únicos, así si se modifica una partida (que es esto a lo que el enunciado llama modificar) cambiando la duración de la misma, se hace una partida nueva que tiene los mismos datos de origen que la que se ha “modificado”, salvo el ID que es autoincrementable, el nombre que varía en su parte final al incluir la duración como parte del mismo nombre y por defecto se pone el finalizado a ‘no’. El resto de los campos se copian. Puede parecer algo extravagante pero lo que pone en el enunciado es:

*“Duración de la partida.* *(el administrador selecciona o modifica el tiempo quiere que dure una partida de un determinado juego. De esta forma se pueden crear partidas del mismo juego y con diferente duración).* “

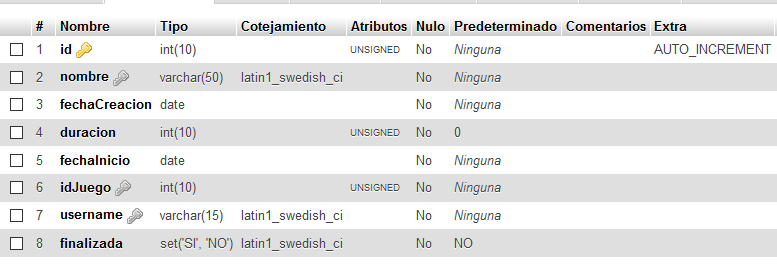


Ilustración 5: Estructura de la tabla Partidas



Ilustración 6: Datos de prueba de la tabla Partidas

## Resolución

Sirve para gestionar los datos que se encuentran en la tabla resoluciones de la BBDD. En negrita las claves primarias. Tiene 5 atributos privados que coinciden con los campos de dicha tabla:

**idPrueba**: Identifica la prueba a la que pertenece la resolución

**idEquipo**: Identifica el equipo al que pertenece la resolución

resuelta: Para saber si se ha resuelto o no

intentos: El número de intentos de resolución

estrellas: Las estrellas obtenidas

Como métodos tenemos en primer lugar el constructor que genera un nuevo objeto con todos y cada uno de los atributos inicializados con los valores que pasa una variable de tipo columna con un registro completo obtenido de la tabla resoluciones de la BBDD.

Además existen los getters necesarios para poder visualizar todos y cada uno de los atributos.

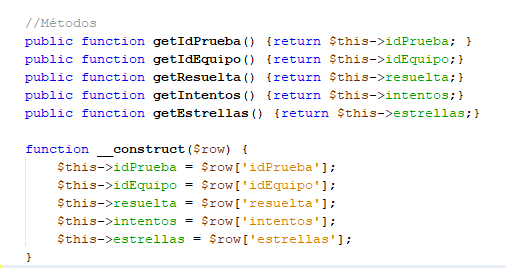


Ilustración 7: Clase Resolución

En cuanto a las modificaciones en la tabla estas son:

1. He añadido más datos de otros equipos para probar

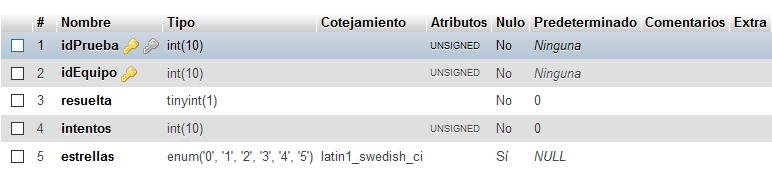


Ilustración 8: Estructura de la tabla Resoluciones



Ilustración 9: Datos de prueba de la tabla Resoluciones

## prueba

**LUIS**

**Etc**

# Plan de pruebas

Supongo que será mejor esperar a tener todo el código funcionando para que las pruebas sean reales.

# Reparto del trabajo

Para el reparto del trabajo se ha realizado la técnica de gestión de equipos de trabajo llamada scrum.

Roles:

Scrum-master: Mikel

Team: Iñaki Irisarri, Luis Sampayo y Carlos Larrondo

|  |  |  |
| --- | --- | --- |
| Iñaki Irisarri | Luis Sampayo | Carlos Larrondo |
|  |  | Página 2 |
|  |  | Página 4 |
|  |  | Página 7 |
|  |  |  |
| Mikel | | |
| Análisis de la aplicación | | |
| Establecimiento del entorno de trabajo y compartición de información GITHUB | | |
| Reparto de tareas | | |
| Seguimiento de tareas | | |
| Ayuda con las dudas de los miembros del Team | | |
| Soporte a los miembros del Team cuando los plazos aprietan | | |
| Integración de las páginas | | |
| Retoques de codificación | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |

# Tabla de ilustraciones

[Ilustración 1: Clase Equipo 3](#_Toc38210682)

[Ilustración 2: Estructura de la tabla Equipos 4](#_Toc38210683)

[Ilustración 3: Datos de prueba de la tabla Equipos 4](#_Toc38210684)

[Ilustración 4: Clase Partida 5](#_Toc38210685)

[Ilustración 5: Estructura de la tabla Partidas 6](#_Toc38210686)

[Ilustración 6: Datos de prueba de la tabla Partidas 6](#_Toc38210687)

[Ilustración 7: Clase Resolución 7](#_Toc38210688)

[Ilustración 8: Estructura de la tabla Resoluciones 7](#_Toc38210689)

[Ilustración 9: Datos de prueba de la tabla Resoluciones 7](#_Toc38210690)