

Lunar Lander ALY

Manual de desarrollo

Información sobre bases de datos	2
Front End	4
Cookies	4
Registro	4
Login	4
Controles	5
Configuración	5
Registros y listas	5
Back End	6
Controllers	6
Gson	6
Listeners	6
Model	6
pcl_utilidades	6
Servlets	6

1. Información sobre bases de datos.

Definición de la base de datos

La base de datos Lunar Lander consta de tres tablas:

- Usuarios: Se guarda información sobre usuarios así como sus datos de inicio de sesión.
- Configuraciones: Se guardan configuraciones del juego asociadas a un usuario.
- Puntuaciones: Se guardan datos relacionados con las partidas asociadas a un usuario.

Esta estructura se crea a través de JPA Entities en PostgreSQL.

Junto a estas estructuras se crean unas clases controller con algunas funciones básicas o necesarias para que los servlets manejen la base de datos.

Usuarios

Nombre	Tipo de dato
Id_user	Serial, Primary Key
Nombre	Varchar
Apellido	Varchar
Usuario	Varchar
Contraseña	Varchar

Configuraciones

Nombre	Tipo de dato
Id_configuracion	Serial, Primary Key
Id_user	Integer, Foreign Key
dificultad	Varchar
modelo_luna	Varchar
modelo_nave	Varchar
nombre_configuracion	Varchar

Puntuaciones

Nombre	Tipo de dato
Id_Puntuacion	Serial, Primary Key
Id_user	Integer, Foreign Key
Id_configuracion	Integer, Foreign Key
Fuel	BigDecimal
Velocidad	BigDecimal
Ini_time	timeStand
End_time	timeStand

2. Front End.

Login/Registro

Al hacer registro o login se comprueba que los campos no esten vacios y se recoge la información necesaria en cada caso.

• Cookies

Utilizamos cookies para recordar y comprobar inicios de sesión en la aplicación.

Al entrar en el sitio web si existen cookies se comparan con los datos de la base de datos.

Tiene que existir el usuario de la cookie. Si este existe se compara la contraseña de la cookie con la del usuario.

Si es válido se permite acceso al sitio web del juego.

La cookie se crea al realizar un login con éxito.

• Registrar usuario

Se efectúa mediante una petición AJAX al servlet PostUsuario.

En esta petición se envían los siguientes parámetros:

- Nombre: Nombre del dueño de la cuenta.
- Apellido: Apellido del dueño de la cuenta.
- Usuario: Nombre de la cuenta para realizar un login.
- Contraseña: Contraseña de la cuenta para realizar el login.

En caso de recibir una respuesta negativa del servlet se muestra el mensaje de error que sea necesario:

- Campo vacío.
- Errores que envía el servlet.
- Error desconocido para cualquier otro no tratado.

Si se recibe una respuesta positiva del servlet se limpian los campos y se muestra el div de login.

• Login usuario

Se efectúa mediante una petición AJAX al servlet GetConnection.

En esta petición se envían los siguientes parámetros:

- Usuario: Nombre de la cuenta para realizar un login.
- Contraseña: Contraseña de la cuenta para realizar el login.

En caso de recibir una respuesta negativa del servlet se muestra el mensaje de error que sea necesario:

- Campo vacío.
- Errores que envía el servlet.
- Error desconocido para cualquier otro no tratado.

Si se recibe una respuesta positiva del servlet se limpian los campos y se muestra el div del juego.

Juego

• Controles

Controles en navegador versión de escritorio:

Utilizar **espacio** para activar el motor de la nave y frenar la caída.

Para navegar por el sitio web utilizar ratón para clicar en la opciones.

Controles en navegador versión de móvil:

Tocar la pantalla para activar el motor de la nave y frenar la caída.

Tocar los botones de las opciones para navegar por el sitio web.

• Configuración

Permite crear, borrar y seleccionar configuraciones del usuario.

Para crear una configuración se escribe un nombre y se modifican los atributos a través del juego.

Pulsando guardar se envía al servlet PostConfiguraciones los datos de configuración (modelo luna, modelo nave y dificultad) junto al nombre proporcionado a esta y la id del usuario a través de una petición AJAX.

Si la respuesta es positiva se verá la nueva configuración para seleccionarla,

Se la respuesta es negativa se mostrará el mensaje de error pertinente o uno de error desconocido.

• Puntuaciones y listas

Las puntuaciones y listas que se muestran en el menú de configuración son lista de usuarios online y el top 10 usuarios que más juegan.

· Lista de usuarios online:

Mediante una petición Ajax al servlet GetUsuariosOnline se solicitan estos datos y se muestran en una tabla en el menú de configuración.

· Top 10 usuarios que más partidas juegan:

Mediante una petición al servlet GetTopJugadores se muestra en el menú de configuración los 10 usuarios que más partidas han realizado.

2. Back End.

• Controllers

Contiene una clase controlador para cada tabla de la base de datos.

• Gson

Contiene una clase con los métodos para utilizar las librerías Gson.

• Listeners

Contiene un context listener para las operaciones de inicialización.

• Model

Contiene una clase plantilla para cada una de las tablas de la base de datos.

• pck_utilidades

Contiene una clase con métodos para utilizar las librerías Gson.

• Servlets

La aplicación usa 8 servlets más clases con la función de complementar las funciones de estos servlets:

PostLogin:

Parámetros: usuario y contraseña

Servlet que nos permite validar si el usuario es correcto.

Puede hacerlo mediante dos formas:

Se buscan las Cookies usuario/password

Mediante los parámetros del request Usuario y contraseña.

Devuelve en el response las configuraciones del usuario validado a través de un String Json.

Contemplados posibles errores de al conectar con la base de datos o no poder validar el inicio de sesión

PostUsuario:

Parámetros: nombre, apellido, usuario y contraseña

Servlet que recibe los datos del usuario por request y los guarda en la base de datos.

Si un usuario ya existe, o no se pudo guardar correctamente saltará el error pertinente.

PostConfiguraciones:

Parámetros: nombre, modelo luna, modelo nave, dificultad e Id de usuario.

Servlet que recibe los datos de una configuración que un usuario quiere guardar. Esta se guarda en la base de datos y si el proceso es correcto devuelve todas las configuraciones a través del response en un String Json, para mantener actualizado el Frontend.

Si la configuración existe, o no se ha podido guardar correctamente salta el error pertinente.

PostBorrar:

Parámetros: id configuración

Servlet que recibe el id de una configuración que el usuario desea borrar y una vez eliminado de la base de datos, devuelve el conjunto de configuraciones restante, en formato Json para mantener actualizado el Frontend.

PostInicioPartida:

Parámetros: id partida, fecha de inicio

Servlet que crea una nueva partida en la base de datos. Devuelve el id de dicha partida en formato Json al Frontend. Se guarda el inicio de partida cada vez que comienza una partida, ya sea al loguearse, como al reiniciar.

PostFinPartida:

Parámetros: id puntuacion, fecha de inicio, fecha fin, velocidad, fuel, id configuración

Con el id proporcionado al inicio de partida se guarda el momento en que la nave toca la base la fecha de finalización y se envía junto a la fecha de inicio, la velocidad, el fuel y la configuración seleccionada a la base de datos.

GetUsuariosOnline:

Parámetros: Ninguno

Función que devuelve un listado con los jugadores Online, a través de un String Json al response de la petición Ajax del Frontend.

Se hace mediante una consulta sobre las partidas activas de los últimos 5 minutos.

GetTopJugadores:

Parámetros: Ninguno

Función que devuelve un listado con los 10 primeros registros de un select que muestra el número de partidas de cada jugador ordenadas de más a menos.

Estos datos se devuelve en formato Json al Frontend para mostrarlos al usuario.