Buscaminas

Vamos a realizar una aplicación WEB que permita gestionar partidas de buscamina. La aplicación guardará las partidas activas y el histórico de las partidas jugadas (hayan sido ganadas o perdidas); también se contabilizará la cantidad de partidas ganadas.
Asimismo la aplicación permitirá un sistema de gestión de usuarios accesible solo por los administradores.
Nuestra aplicación tendrá los siguientes roles: ***administrador*** y ***jugador***.

Administrador

El **administrador** gestionará: altas, bajas, modificaciones, activaciones y accesos de los usuarios. El administrador será otro jugador que podrá seleccionar como quiere acceder a la aplicación. Si accede como administrador podrá**: listar los usuarios, buscar un usuario concreto, registrar, modificar y eliminar usuarios**. También estará habilitado para **cambiar la contraseña** de un usuario concreto.
Si entra como jugador será un jugador más.

Jugador

El **jugador** podrá **crear partidas personalizadas o estándar**. Si en la url se especifica tamaño de tablero y minas se creará un buscaminas con esas características. En caso contrario se creará un buscaminas con un tamaño y número de minas predefinidos (esta cantidad deberá estar parametrizada en la clase de constantes de la aplicación).
El **jugador jugará** indicando (POST + json) qué casilla quiere destapar, el cliente le informará de lo que pueda ocurrir: no tienes partida creada, no tienes partida abierta, partida abierta y has destapado una casilla no mina, partida abierta y has destapado una mina… En caso de que no haya partida abierta se informa de ello. En caso de partida abierta se informa al cliente de ello y se le enviarán los tableros en json para que el cliente haga lo que estime oportuno con ellos.
El jugador podrá solicitar **rendirse**, (verbo y forma de proporcionar información al servidor depende del programador). Se cerrará esa partida y se informará al cliente de cómo estaban los tableros y de que se ha cerrado esa partida que se considera perdida.

El jugador también podrá solicitar un **cambio de contraseña**, para ello debe proporcionar su email al servidor (piensa el verbo y el modo más correcto de hacer esto).
Finalmente el jugador podrá solicitar el **ranking de jugadores**. Se le devolverá una lista de usuarios ordenada de mayor a menor de más ganadas a menos. De igual manera el verbo y la forma de solicitar el ranking depende del programador.

Consideraciones generales

Respecto a la **planificación** y **otras consideraciones**:

- Se valorará una **planificación de tareas** basada en funcionalidades concretas, realistas y bien estimadas en dificultad y/o duración. Debo tener acceso a ese tablero para ver la evolución.
- Una **base de datos** correctamente diseñada e implementada.
- El uso de **repositorios** y ramas de desarrollo. Debo ser colaborador del repositorio para ver el avance.
- **Tableros** bien organizados y actualizados.
- **Documentación** inicial y final útil, completa (que no tiene por qué ser demasiado extensa) y funcional. Esta documentación se limitará a un breve resumen del uso de la API que será escrito en el archivo README.md del repositorio.
  Esta documentación será necesaria para que el cliente pueda usar esta API. En dicha documentación tiene que haber ejemplos de uso del servicio.

Respecto a la **programación**, se valorará:

- La elección correcta de los verbos.
- Paso por el lugar indicado y en el formato adecuado de los datos desde el cliente al servidor.
- Protección de las rutas, de modo que solo el usuario que sea administrador pueda administrar y un usuario jugador pueda jugar. No se admiten invitados en nuestra aplicación; solo registrados correctamente.
- Se le enviará al cliente en el formato adecuado la información correspondiente en cada solicitud de servicio, teniendo en cueta: códigos coherentes y estándar; y la información que sea necesaria, en el formato adecuado, para que el cliente pueda montar la interfaz del juego con todos los datos necesarios.
- Diseño y acceso a la base de datos de forma correcta.

- Estructuración del código: Controladores, Factorías, Rutas (index.php), Conexión, Modelos….
- Se debe usar la librería MySQLi orientada a objetos.
- La clase Constantes se debe poner en .gitignore.
- La base de datos exportada no debería estar en el repositorio pero, por esta vez, no pasaría nada si la subimos al repositorio.