**Consultas para hacer:**

**getUser**: recibe un token (dato de cuando un usuario se registra) y devuelve un usuario, para el inicio de sesión.

**getUsers:** devuelve todos los usuarios, para que el admin pueda ver todos los usuarios.

**getProducts:** devuelve todos los productos, para que la página liste todos los productos.

**getProduct:** Recibe un id de un producto y devuelve toda la información de ese id, para mirar a detalle el contenido de un producto.

**getCommentarys:** Devuelve todos los comentarios de la página, para que el admin, pueda ver todos los comentarios hechos en la página.

**getCommentarysUser:** Devuelve todos los comentarios que el usuario en sesión ha hecho.

**getCommentarysProduct:** Recibe un id y devuelve todos los comentarios que se le han hecho a un producto.

**insertUser:** Recibe el nombre, apellido, correo, contraseña, y el tipo, todos son necesarios menos el tipo, si se hace el registro desde la página común, el usuario será común, pero en caso de ser en la página de un admin, este puede decidir si quiere que sea admin o no, retorna un usuario

**authenticateUser:** Recibe un email, y una contraseña y retorna un token, (necesario para el inicio de sesión y seguridad).

**updateUser:** Recibe un input como el de insert, un id (no necesario), en caso de no pasar id, significa que se quiere actualizar en usuario en sesión, si se pasa el id, significa que el que desea actualizar es un admin, y este podrá actualizar cualquier usuario, retorna el nuevo usuario, retorna una cadena con un mensaje de eliminación.

**deleteUser:** Recibe un id (no es necesario), en caso de no pasar id significa que el usuario que se quiere eliminar es el usuario en sesión, si se pasa id, quiere decir que el usuario es un admin y por lo tanto puede eliminar cualquier usuario.

**insertProduct:** Recibe el nombre, una descripción, un precio, ingredientes, y el tipo que solo puede ser (drinks, desserts, tickets, maincourse), necesarios para filtrar por página, todos los campos son necesarios, retorna un producto.

**updateProduct:** Recibe un id necesario, y un input similar al de insert, busca el producto por su id, y cambia los espacios que se desean cambiar, retorna el nuevo producto.

**deleteProduct:** Recibe el id de un producto y elimina el producto con el id, retorna una cadena de texto.

**insertCommentary:** Recibe el id del lugar donde se quiere comentar, y la descripción del producto, el id del que comenta se recibe por medio del usuario que esta en sesión, retorna un comentario.

**updateCommentary:** Recibe el id del comentario del que se quiere actualizar, en caso de que el comentario tenga un idUser diferente a id del usuario en sesión, este no se podrá eliminar, solo el admin puede actualizar cualquier comentario, retorna el nuevo comentario.

**deleteCommentary:** Recibe el id del comentario que se desea eliminar, en caso de que el comentario no tenga el idUser igual al del usuario en sesión no se podrá eliminar, solo el admin puede eliminar cualquier comentario, retorna una cadena con un mensaje de eliminación

Todas las consultas están en un bloque trycatch, en caso de encontrar un error de conexión diferentes a los que se manejan este aparecerá por consola.