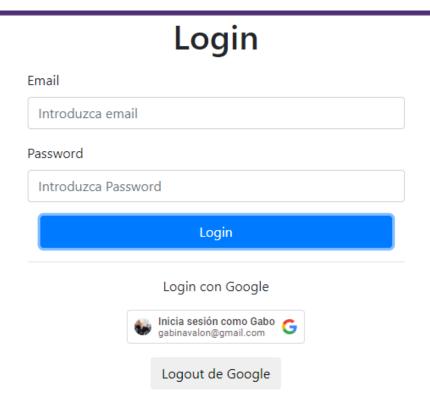
La página inicial cuenta con dos login, uno para Google y otro por formulario, ambos enviaran a la página de gestión de libros si la autentificación es correcta y generarán un localstorage con el token asignado.



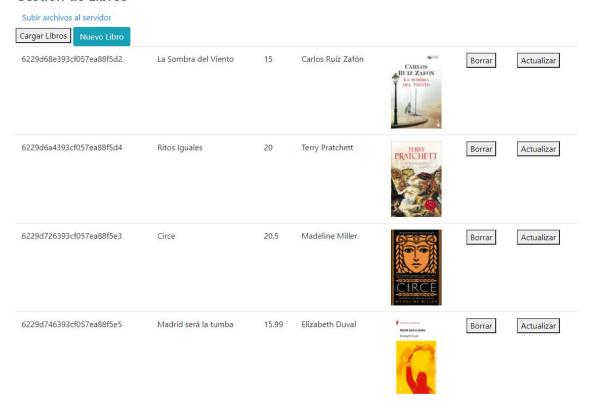
Con el botón de logout se elimina el token y se cierra la sesión de Google. Cuando la autentificación es errónea, nos aparece un alert con el error, si es correcta, vamos a la gestión, sin importar con que login sea.

Tanto en la página de libros como en la de subir libros o archivos, comprobamos que el token que se guarda en el localstorage existe para así permitir el acceso a la página, si no, redireccionamos al login.

```
<script>
if (localStorage.getItem('token') == null) {
   localStorage.setItem('mensaje', "Inicia sesión para entrar");
   window.location.href = "index.html";
}
</script>
```

Gabriel Navalón Soriano, tarea final DAW 2021-2022

Gestión de Libros



En esrta página podemos editar los libros que hay en la base de datos o borrarlos. Todas las operaciones (hasta mostrar) requieren del token. Este se manda por cabeceras en los fetch.

```
const token = localStorage.getItem('token');

function cargaLibroPorId(idLibro) {
   $('#myModal').modal('show');
}
```

```
//delete libros
this.app.delete(
   "/webresources/generic/libros/:id",
   validarJWT,
   async function (req, res) {
    const id = req.params.id;
    await Libro.findByIdAndDelete(id);
    res.status(200).json({
        ok: true,
        msg: "delete API",
        });
   }
);
```

Como la página de subir archivos al servidor y subir libros son básicamente la misma, solo que la segunda cuenta con un formulario para añadir libros a la base de datos, vamos a ver la de subir libros.



Insertamos el libro al igual que en el código de ejemplo, y nos guardará un libro en la base de datos además de guardar la foto en la carpeta /img para poder así mostrarla o consultarla más adelante.

6229e86845d0a5374c576f0b

Frankestein

5 Mary W. Shelley



Borrar

Actualizar