**Informe**:Desarrollo de aplicación web y móvil ChanceUTB

Santiago Mendoza Ramirez - T00026240

Christian Duban - T00017533

Belkis Buelvas Castillo - T000

Este documento presenta la documentación del proyecto ChanceUTB, una aplicación web y móvil que permite a los estudiantes/personal administrativo de la Universidad Tecnológica de Bolívar ofrecer chances (aventones) a otros estudiantes/personal administrativo de la universidad.

**Tecnologías usadas**

**Backend**:

* PHP con Framework MVC Laravel 4.2.

**Frontend**:

* HTML, CSS3 y javascript con Framework AngularJS 1.2.x
* JAVA Android.

**Requerimientos**

Para que la aplicación satisficiera a los usuarios finales de la aplicación, se plantearon ciertos requerimientos funcionales y no funcionales, iniciales, para la aplicación.

1. **Registro de usuarios**: Permitir a los usuarios registrarse en la plataforma, únicamente con un correo institucional de la universidad (@unitecnologica.edu.co o @utbvirtual.edu.co).
2. **Login de usuarios**: Permitir a cada usuario loguearse en la plataforma usando únicamente su correo electrónico con el que creó la cuenta y su contraseña.
3. **Dar de baja un usuario**: Permitir a los usuarios “darse de baja”. Sin embargo, se exige que los datos del usuario queden guardados en la base de datos, pero con la cuenta desactivada, de manera que no pueda acceder a ella.
4. **Editar usuario**: El usuario puede editar la información inicial con la cual creó su cuenta.
5. **Crear vehículo**: Permitir a los usuarios agregar un vehículo a su cuenta. Los usuarios pueden tener ‘n’ vehículos.
6. **Editar vehículo**: El usuario puede editar la información suministrada de su vehículo.
7. **Eliminar vehículo**: El usuario puede, en cualquier momento, decidir eliminar su vehículo. Sin embargo, la información sobre este quedará guardada en la base de datos, solo que este permanecerá inactivo.
8. **Crear chance**: Un usuario, que tenga un vehículo activo en su cuenta, puede crear chances.
9. **Editar chance**: El chance debe poder editarse hasta un máximo de tiempo de 5 minutos. Luego, el chance no podrá ser editado.
10. **Eliminar chance**: El chance puede ser eliminado, siempre y cuando no haya ningún usuario que haya ‘tomado’ ese chance.
11. **Agregarse a un chance**: Un usuario puede agregarse a un chance, siempre y cuando este todavía tenga cupos disponibles.
12. **Ver lista de chances**: Los usuarios pueden ver la lista de chances activos, es decir, aquellos en que la hora de salida todavía no halla pasado.
13. **Calificar pasajeros del chance**: El usuario que ofreció el chance, puede calificar a los pasajeros de acuerdo a su comportamiento, puntualidad y cumplimiento.
14. **Calificar “chofer” del chance**: Los pasajeros pueden calificar a quien ofreció el chance.
15. **Crear lista de amigos**: Los usuarios pueden agregar a otros usuarios a su lista de amigos.
16. **Crear chances ‘exclusivos’**: Crear un chance que solo sea visible para su lista de amigos.
17. **Invitar usuarios**: Invitar usuarios a través de correo electrónico.
18. **Restablecer contraseña**: Permitir que los usuarios puedan restablecer su contraseña desde el correo electrónico con la cual crearon la cuenta.
19. **Disponible Android**.
20. **Disponible iOs**.

**Cumplimiento**

En el transcurso de las 17 semanas del curso, logramos desarrollar la mayoría de los requerimientos planteados inicialmente, con excepción de algunos pocos no esenciales para el funcionamiento de la aplicación, y los cuales pueden ser agregados más tarde sin ningún problema y sin afectar el funcionamiento de la aplicación.

Estos son:

15. Crear lista de amigos

16. Crear chances ‘exclusivos’

17. Invitar usuarios.

18. Restablecer contraseña

20. Disponible iOS

**Código fuente**

Todo el código fuente del backend y el frontend se encuentran en repositorios en GitHub, que utilizamos como control de versiones.

Backend: <https://github.com/sanmen1593/ChancesAPI>

Frontend AngularJS: <https://github.com/sanmen1593/ChancesFrontend>

Frontend Android: <https://github.com/andrescoulson/NuevoRepositorio>

**Documentación**

Utilizamos las rutas de laravel para administrar las peticiones que se realizan a la aplicación. Cada petición debe contener entre las variables que se envían, el token (auth-token) de autorización, que se recibe cada vez que el usuario se autentica. Una vez se ha comprobado que el token es correcto, se busca a quien pertenece, y todo se hará con la cuenta del usuario a quien pertenece el token enviado.

Las rutas, junto con sus métodos, funcionalidad y variables, están definidas en la siguiente tabla:

|  |  |  |  |
| --- | --- | --- | --- |
| Ruta | Método | Funcionalidad | Variables |
| /signup | POST | Registrar usuario | name  lastname  email  email\_confirmation  password |
| /sessions | POST | Login de usuario | email  password |
| /logout | GET | Logout de usuario |  |
| /vehicle | POST | Registrar vehículo | plate  color  brand  model  capacity  type |
| /vehicle | GET | Listar vehículos del usuario |  |
| /vehicle/{id} | GET | Mostrar el vehículo con el id {id} |  |
| /vehicle/{id} | DESTROY | Eliminar el vehículo con el id {id} |  |
| /vehicle/{id} | PUT/PATCH | Actualizar el vehículo | plate  color  brand  model  capacity  type |
| /chanceslist | GET | Listar la lista de chances |  |
| /chance | POST | Crear un chance | fee  date  hour  destination  departure  capacity  comments  route  vehicles\_id |
| /chance/{id} | GET | Ver el chance con el id {id} |  |
| /userofchance | POST | Agregarse a un chance | chances\_id |
| /comments | GET | Listar todos los comentarios |  |
| /comments | POST | Agregar un comentario | chances\_id  text |
| /rated | POST | Calificar a un usuario | comments  rate  chances\_id  raters\_id  rated\_rol  users\_id |

Todas las rutas deben incluir entre los parámetros/variables enviadas, el token de autenticación (con el nombre auth-token), a excepción de las que están sombreadas de color. Aquellas en las que el método sea GET, se puede enviar la variable agregando al final de la url o ruta lo siguiente: **?auth-token=** y se coloca el token.

Para realizar estas funcionalidades, utilizamos las siguientes entidades, representadas en el siguiente **diagrama de bases de datos**:

