



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2513 - TECNOLOGÍAS Y APLICACIONES WEB

Documentación Proyecto Semestral

16 de junio de 2019

1º semestre 2019 - Profesor Gabriel Vidal

Mauricio López Socías - Francisco Paniagua - Luis Loyola

Introducción

Re - z es el proyecto desarrollado por GrupoIncompleto durante el semestre 2019-1 para el ramo IIC2513 - Tecnologías y Aplicaciones Web, dictado en la Pontificia Universidad Católica de Chile. consiste en una plataforma que permite al usuario registrarse, crear, ver, guardar y comentar recetas. Se encuentra construida utilizando principalmente NodeJS, KOA y React.

Montando el proyecto

Prerequisitos

- PostgreSQL
- Node.js v8.4.0
- Yarn

Variables de Entorno:

Se deberá crear un archivo llamado `.env` en la raíz del proyecto, que debe contener las siguientes variables de entorno:

- Sendgrid:
La aplicación utiliza el servicio de Sendgrid para controlar las notificaciones el e-mail del usuario.
 - SENDGRID_USER: Nombre de usuario de Sendgrid

- SENDGRID_PASS: Contraseña de usuario de Sendgrid
- Base de datos:
La aplicación requiere poder autenticarse con la Base de datos local.
 - DB_USERNAME: Nombre de usuario de PostgreSQL
 - DB_PASSWORD: Contraseña de usuario de PostgreSQL
- Imágenes:
La aplicación utiliza los servicios de Cloudinary para manejar las imágenes.
 - CLOUDINARY_CLOUD_NAME: Nombre de nube.
 - CLOUDINARY_API_KEY: Llave de API.
 - CLOUDINARY_API_SECRET: API secret.
- Telegram:
La aplicación es capaz de enviar notificaciones a los usuarios por medio de un BOT de telegram.
 - TELEGRAM_BOT_TOKEN: Token de BOT.

Setup

- Instalar Dependencias:

```
yarn install
```
- Correr migraciones:

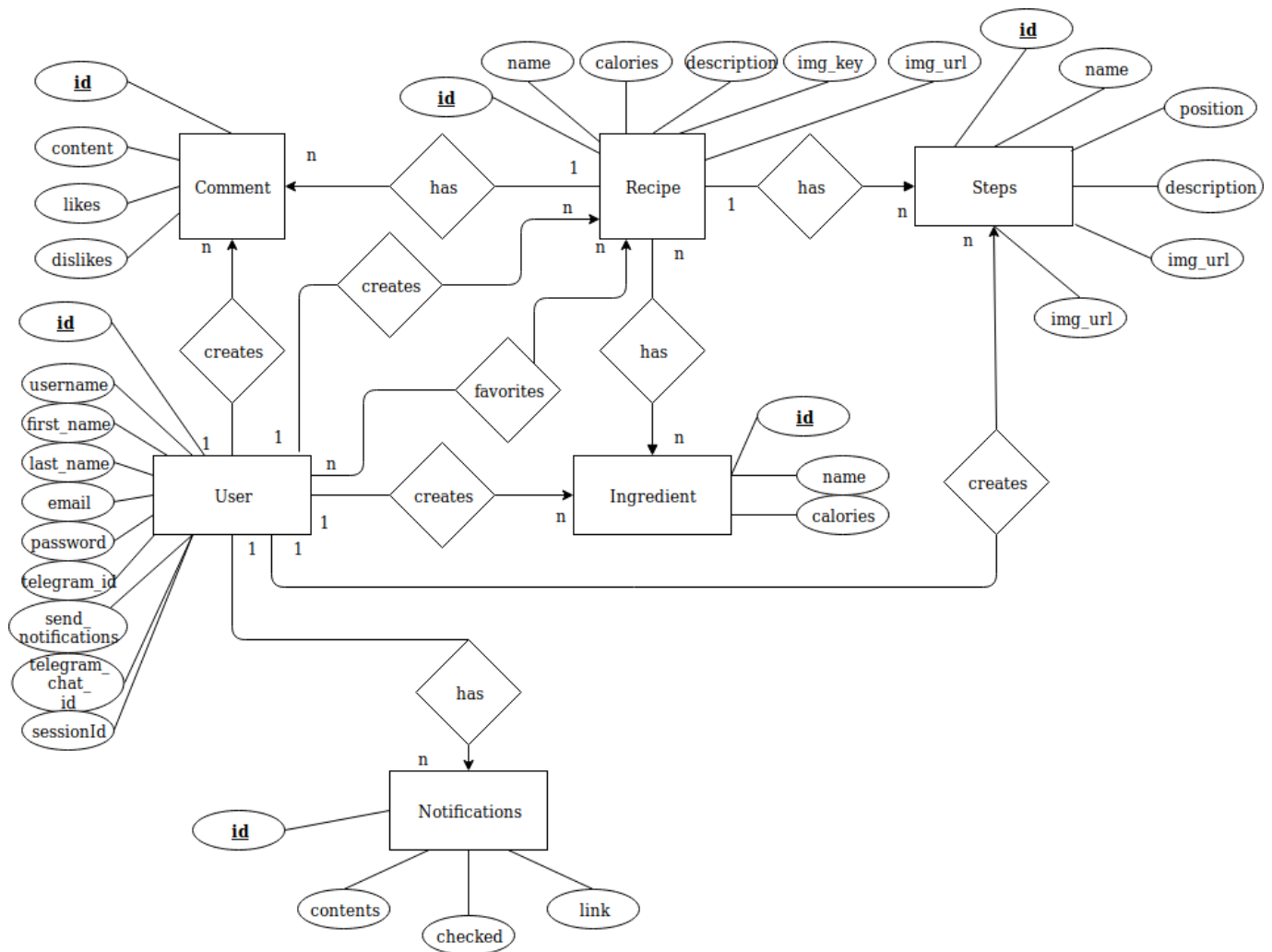
```
./node_modules/.bin/sequelize db:migrate
```
- Correr la aplicación:

```
yarn start
```

Ahora la aplicación debe estar corriendo de manera local en <http://localhost:3000/>

Base de datos:

Diagrama E/R



Utilización de React

Se utiliza react para manejar las notificaciones. Se actualiza tanto el ícono de la campana como la notificación misma. Para la conexión de los componentes de react con el back-end, se utiliza una API interna, que funciona con las cookies del usuario loggeado.

API interna: Comunicación con React para notificaciones

- Obtener notificaciones:
 - Método: GET
 - Ruta:

/notifications/API/

- Argumentos: N/A
- Marcar notificación como leída:
 - Método: POST
 - Ruta:

`/notifications/read/<id>`
- Argumentos:
id: ID de notificación a marcar como leída.

API externa

Las funcionalidades de agregar ingrediente a receta y remover ingrediente de receta, se encuentran protegidas mediante koa-jsonwebtoken, por lo tanto si se requiere usar estas funcionalidades, es necesario, autenticarse primero mediante un post a `/api/auth` con el email y el password, y luego usar el token como se indica en la tabla a continuación.

- Obtener ingredientes:
 - Método: GET
 - Ruta:

`/api/ingredients/`
 - Argumentos: N/A
 - Retorna: Json con lista de ingredientes.
- Ingredientes de una receta:
 - Método: GET
 - Ruta:

`/api/ingredients/<recipeid>`
 - Argumentos:
recipeid: ID de de la receta cuyos ingredientes se quiere obtener.
 - Retorna: Json con lista de ingredientes de la receta indicada.
- Obtener Token the autenticación:
 - Método: POST
 - Ruta:

`/api/auth/`

- Argumentos: N/A
- body: email, password
- Retorna: token de autenticación.
- Agregar ingrediente a receta:
 - Método: POST
 - Ruta:

`/api/recipes/<recipeid>/addingredient`
 - Argumentos:
recipeid : Id de receta a agregar ingredientes
 - Header: Authorization: Bearer {authentication token}
 - body: name, calories
 - Retorna: Json con el ingrediente agregado a la receta.
- Eliminar ingrediente de receta:
 - Método: POST
 - Ruta:

`/api/recipes/:recipeid/
removeingredient/:ingredientid`
 - Argumentos: recipeid : Id de receta a eliminar un ingrediente
ingredientid : Id del ingrediente a eliminar
 - Header: Authorization: Bearer {authentication token}
 - body: name, calories
 - Retorna: Confirmación del ingrediente eliminado.