

TALENT UO

2024

Eric Vivancos Yagües

UO303984

Instrucciones para el proyecto de gestión de regalos

Tabla de contenido

Descripción General	1
Preparación y Configuración del Entorno	1
Instrucciones para Iniciar el Proyecto	2
Estructura del Proyecto	4
Base de Datos	4
Consideraciones Finales	5

Descripción General

Este proyecto consta de tres partes principales

1. **Backend (API RESTful):** Desarrollado con Node.js y Express.js, maneja la lógica del servidor, la autenticación, la gestión de regalos y amigos.
2. **Frontend (React.js):** Dos interfaces de usuario creada con React en versiones de html y css y también con Ant Design que permite a los usuarios interactuar con el sistema.
3. **Base de Datos (MySQL):** Almacena la información de usuarios, regalos y relaciones de amistad

Preparación y Configuración del Entorno

Requisitos previos

- **Node.js:** Debemos asegurarnos de tener Node.js instalado en el sistema
- **MySQL:** Debemos asegurarnos de tener MySQL instalado y en funcionamiento. También debemos tener acceso a alguna herramienta para gestionar la base de datos, como MySQL Workbench o phpMyAdmin.
- **Git:** Para clonar el repositorio, debemos tener Git instalado

Clonación del proyecto

```
1 git clone https://github.com/ericvivancos/cursoprogramacionuo303984.git
2 cd /mi-repositorio/
```

Configuración de Variables de Entorno

En el proyecto de Backend proporcionamos un archivo .env que contiene las variables de entorno necesarias. En modo producción esto quedaría oculto.

Instrucciones para Iniciar el Proyecto

A. Backend

1. Instalar dependencias

```
bash
cd backend
npm install
```

2. Migrar la base de datos

Proporcionamos un script SQL, que debemos cargar en nuestra base de datos MySQL usando alguna herramienta.

3. Iniciar el servidor

```
bash
npm start
```

B. Frontend

1. Instalar dependencias

```
bash
cd frontend
npm install
```

2. Iniciar la aplicación React

```
bash
npm start
```

C. Iniciar el Proyecto con Docker Compose

Docker Compose es una herramienta que te permite definir y ejecutar aplicaciones con múltiples contenedores. En este proyecto, Docker Compose se utiliza para gestionar tanto el backend como frontend y la base de datos.

1. Preparar el Entorno

Debemos asegurarnos de tener instalado tanto Docker y Docker Compose en nuestra máquina.

2. Iniciar los servicios

El proyecto viene con un archivo 'docker-compose.yml' que define los servicios necesarios:

```
docker-compose up --build
```

Este comando construirá y lanzará todos los servicios definidos en el archivo.

3. Servicios y Puertos

a. Backend Server:

- **Puerto:** 5000
- **Descripción:** Este es el servidor Node.js que maneja la lógica de la aplicación y las API. Está configurado para escuchar en el puerto 5000

b. Frontend Server:

- **Puertos:** 3000/3001
- **Descripción:** Este es el servidor React que maneja la interfaz de usuario. Son dos servidores, en el 3000 el html-css y 3001 con Ant-design

c. MySQL Database:

- **Puerto:** '3306'
- **Descripción:** Este es el servidor de base de datos MySQL que se utiliza para almacenar los datos de la aplicación. El script de la base de datos será automáticamente cargado durante la inicialización.

d. phpMyAdmin (Opcional):

- **Puerto:** 8080
- **Descripción:** Este es un servicio opcional que proporcionará una interfaz web para gestionar la base de datos MySQL.

4. Acceso y Gestión

- Frontend: <http://localhost:3000> o <http://localhost:3001>
- Backend: <http://localhost:5000>
- phpMyAdmin: <http://localhost:8080>.

5. Detener los Servicios

```
docker-compose down
```

Este comando detendrá y eliminará todos los contenedores, redes y volúmenes asociados con el proyecto.

Estructura del Proyecto

A. Backend

- **Src/services:** Contiene la lógica para cada una de las rutas
- **Src/repositories:** Contiene los modelos de la base de datos.
- **Src/routers:** Define las rutas del API.
- **Src/middlewares:** Contiene los middlewares como la autenticación.
- **Src/utills:** Funciones de utilidad, como gestión de tokens

B. Frontend

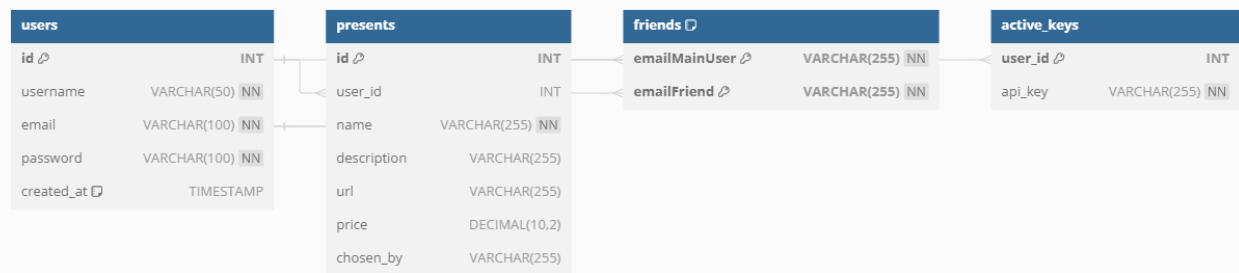
- **Src/components:** Contiene los componentes reutilizables de React.
- **Src/pages:** Contiene las diferentes páginas de la aplicación.
- **Src/context:** Maneja diferentes contextos, como por ejemplo el de la autenticación.
- **Src/Utils:** Funciones de utilidad, como la gestión de consultas a la API.

Base de Datos

Proporcionamos un archivo database.sql y deberá importarlo en su sistema MySQL. Esto creará las tablas necesarias con datos de muestra.

Descripción de las entidades:

- **Users:** Almacena la información de los usuarios registrados
- **Presents:** Almacena los regalos creados por los usuarios
- **Friends:** Almacena las relaciones de amistad entre los usuarios
- **Active_Kets:** Almacena las claves API activas vinculadas a usuarios



Consideraciones Finales

- **Autenticación:** La autenticación se realiza mediante tokens JWT. Debemos asegurarnos de que la variable 'JWT_SECRET' este bien configurada en nuestro archivo .env.
- **Despliegue:** Si en un futuro deseamos desplegar el proyecto en producción, debemos asegurarnos de que todas las variables de entorno estén bien configuradas y utilizar herramientas como 'pm2' o 'build'.