Estiven Montoya

estivenmoto6@gmail.com

Version 3.0.0

Este manual técnico presenta los aspectos técnicos del software.  
Prohibida su venta.

Manual técnico

SHAIR|SENA

Tabla de contenidos

[Prerrequisitos del Sistema 3](#_Toc175144722)

[Lenguajes y Frameworks 4](#_Toc175144723)

[REACT 4](#_Toc175144724)

[EXPRESS 4](#_Toc175144725)

[Scripts 6](#_Toc175144726)

[Diagrama de casos de uso 7](#_Toc175144727)

[Diagrama de clases 8](#_Toc175144728)

[Diccionario de datos 9](#_Toc175144729)

[Procedimiento de restauración del sistema 12](#_Toc175144730)

[Restauración de la Base de Datos MySQL 12](#_Toc175144731)

[Conectar al servidor MySQL: 12](#_Toc175144732)

[Crear la base de datos: 12](#_Toc175144733)

[Restaurar el respaldo: 12](#_Toc175144734)

[Verificar la restauración: 12](#_Toc175144735)

[Reinstalación y Configuración del Servidor Express 12](#_Toc175144736)

[Instalar dependencias: 12](#_Toc175144737)

[Iniciar el servidor: 12](#_Toc175144738)

[Verificar la conectividad: 12](#_Toc175144739)

[Reconfiguración del Frontend en React 12](#_Toc175144740)

[Instalar dependencias: 12](#_Toc175144741)

[Configurar el entorno: 13](#_Toc175144742)

[Construir la aplicación: 13](#_Toc175144743)

[Verificar el despliegue: 13](#_Toc175144744)

[Procedimiento del backup 14](#_Toc175144745)

[Backup de la Base de Datos MySQL 14](#_Toc175144746)

[Conectar al servidor MySQL: 14](#_Toc175144747)

[Generar el archivo de respaldo: 14](#_Toc175144748)

[Verificar el respaldo: 14](#_Toc175144749)

[Backup del Código del Servidor Express 14](#_Toc175144750)

[Identificar los archivos importantes: 14](#_Toc175144751)

[Crear un archivo comprimido: 14](#_Toc175144752)

[Almacenar el backup: 14](#_Toc175144753)

[Backup del Código del Frontend en React 14](#_Toc175144754)

[Identificar los archivos importantes: 14](#_Toc175144755)

[Crear un archivo comprimido: 15](#_Toc175144756)

[Almacenar el backup: 15](#_Toc175144757)

[Scripts de prueba 16](#_Toc175144758)

[Configuración de Cypress 16](#_Toc175144759)

[Instalar Cypress: 16](#_Toc175144760)

[Iniciar Cypress: 16](#_Toc175144761)

[Escribir Pruebas para el Frontend en React 16](#_Toc175144762)

[Crear un archivo de prueba: 16](#_Toc175144763)

[Escribir Pruebas para el Backend en Express 16](#_Toc175144764)

[Crear un archivo de prueba para el backend: 16](#_Toc175144765)

[Ejecutar las Pruebas 16](#_Toc175144766)

# Prerrequisitos del Sistema

El servidor donde se ejecutará SHAIR necesita los siguientes requisitos de hardware para funcionar correctamente.

* 16GB RAM (Recomendado para un uso masivo)
* Core i5 décima generación / Ryzen 5 serie 7000
* Gráficos compatibles con DirectX
* 1TB de almacenamiento HDD/SSD (Recomendado para un uso masivo)
* Sistema operativo Windows 11/Windows Server
* Apache 5.2.1 (Versión del servidor: 10.4.28-MariaDB)
* MySQL (Última versión)
* React Native 18.3.1
* Express JS 4.19.2

# Lenguajes y Frameworks

SHAIR usa dos tecnologías base para su funcionamiento, React y Express. A continuación, se listan las dependencias y sus versiones necesarias a instalar para ejecutar el software de manera correcta.

## REACT

* @testing-library/jest-dom: 5.17.0
  + Librería predeterminada de React
* @testing-library/react: 13.4.0
  + Librería predeterminada de React
* @testing-library/user-event: "^13.5.0
  + Librería predeterminada de React
* axios: 1.7.2
  + Librería para realizar solicitudes al backend
* date-fns: 3.6.0
  + Librería para formatear la fecha
* framer-motion: 11.2.4
  + Librería para generar animaciones
* jwt-decode: 4.0.0
  + Librería para decodificar tokens
* react: 18.3.1
  + Librería predeterminada de React
* react-dom: 18.3.1
  + Librería predeterminada de React
* react-hook-form: 7.52.1
  + Librería para gestionar formularios
* react-router-dom: 6.23.1
  + Librería predeterminada de React
* react-scripts: 5.0.1
  + Librería predeterminada de React
* web-vitals: 2.1.4
  + Librería predeterminada de React

## EXPRESS

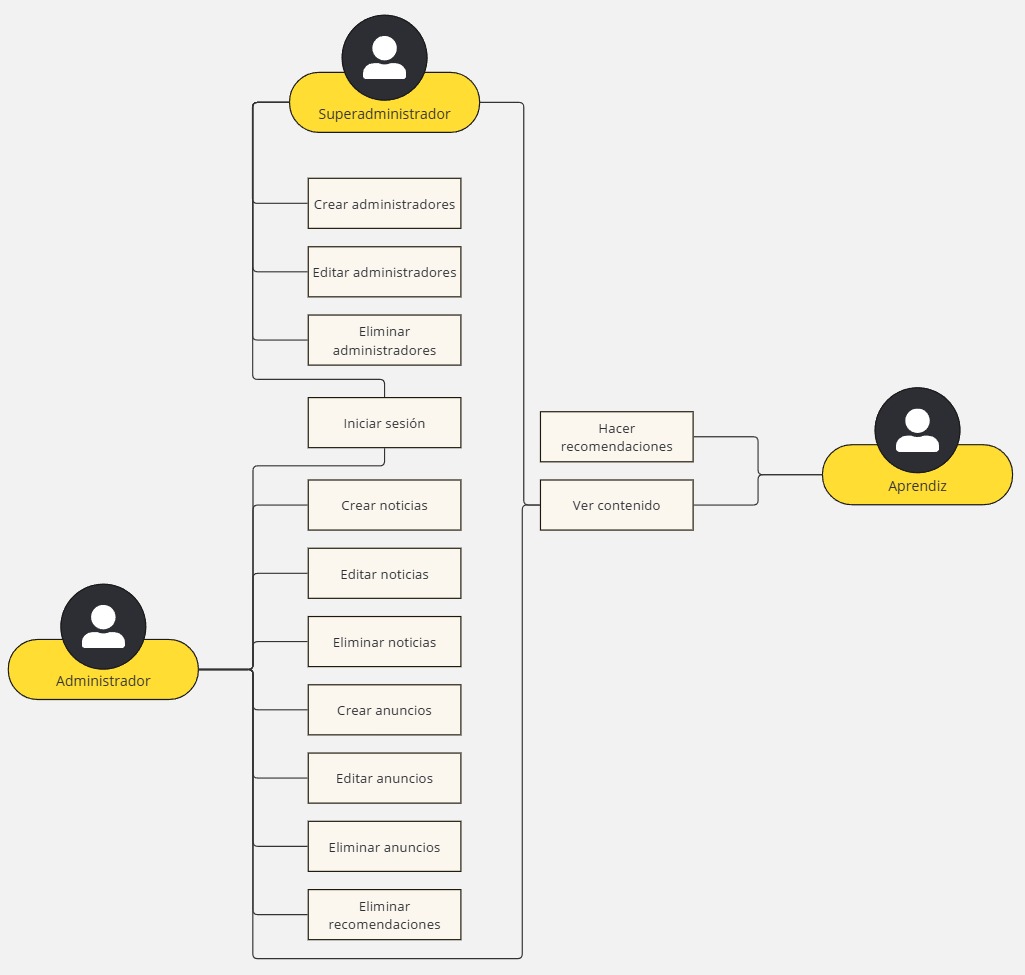
* bcrypt: 5.1.1
  + Librería para encriptar y desencriptar contraseñas
* body-parser: 1.20.2
  + Librería para formatear el texto
* cors: 2.8.5
  + Librería para gestionar el permiso de las CORS
* express: 4.19.2
  + Librería predeterminada de Express
* express-fileupload: 1.5.0
  + Librería para gestionar la carga de archivos
* jsonwebtoken: 9.0.2
  + Librería para codificar y generar tokens
* multer: 1.4.5-lts.1
  + Librería para gestionar la carga de archivos
* mysql2: 3.9.7
  + Librería para realizar comandos SQL
* sequelize: 6.37.3
  + ORM para realizar comandos SQL

# Scripts

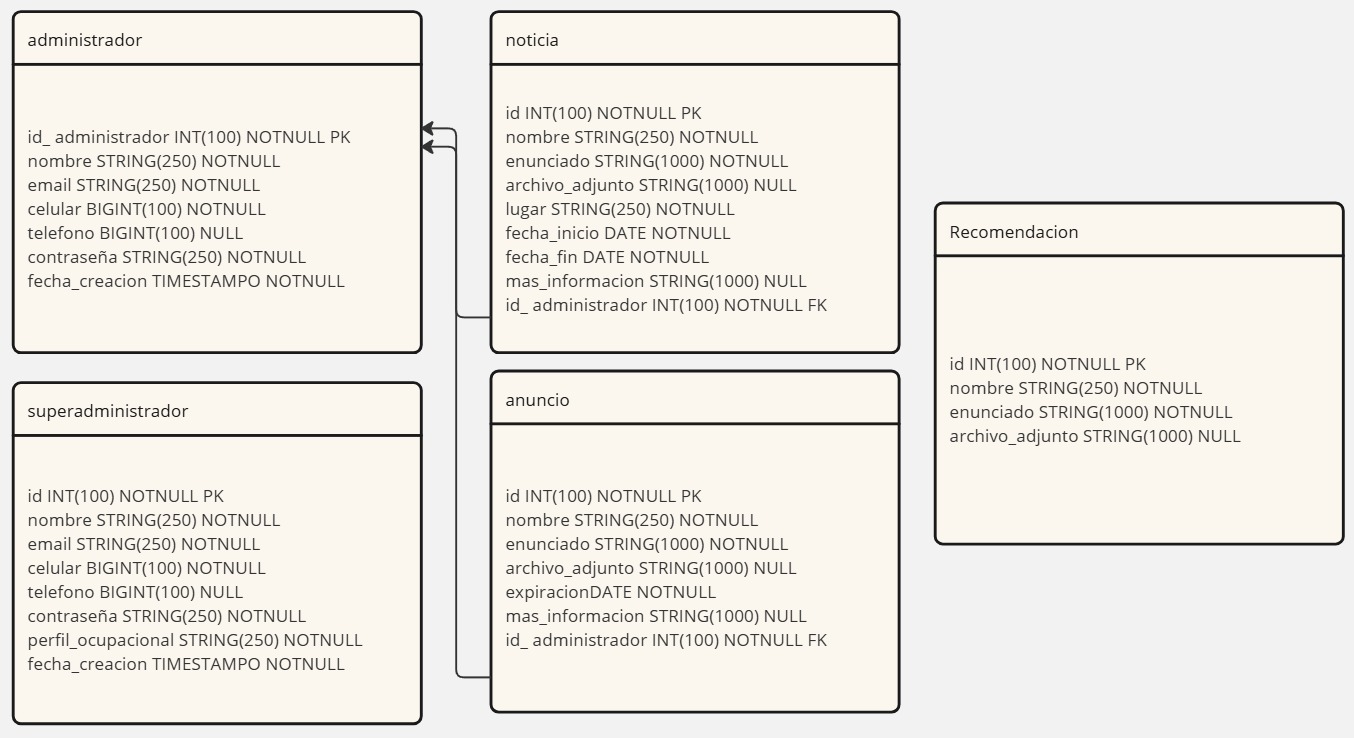
Se presentan múltiples scripts para el proyecto, a continuación se muestran cuales son:

* npm start: react-scripts para iniciar el software de React
* npm build: react-scripts para pasar el software a produccion
* npm test: react-scripts para realizar test en el software
* npm eject: react-scripts para ejecutar el software
* npm install: react-scripts para instalar todas las dependencias necesarias
* node App.js: express-scripts para iniciar el software de Express

# Diagrama de casos de uso



# Diagrama de clases



# Diccionario de datos

A document with a number of text

Description automatically generated with medium confidence

A screenshot of a document

Description automatically generated

A screenshot of a computer

Description automatically generated

# Procedimiento de restauración del sistema

## Restauración de la Base de Datos MySQL

### Conectar al servidor MySQL:

Utiliza un cliente de MySQL como phpMyAdmin, MySQL Workbench o la línea de comandos de MySQL para conectarte al servidor de base de datos.

### Crear la base de datos:

Ejecuta el siguiente comando para crear la base de datos:

CREATE DATABASE shair;

### Restaurar el respaldo:

Para restaurar la base de datos desde un archivo de respaldo .sql, utiliza el siguiente comando:

mysql -u usuario -p shair < /ruta/al/archivo/respaldo.sql

Ingresa la contraseña cuando se te solicite.

### Verificar la restauración:

Asegúrate de que todos los datos se hayan importado correctamente verificando algunas tablas claves y registros.

## Reinstalación y Configuración del Servidor Express

### Instalar dependencias:

Dirígete al directorio del backend (Express) y asegúrate de que todas las dependencias estén instaladas ejecutando:

npm install

### Iniciar el servidor:

Ejecuta el siguiente comando para iniciar el servidor:

Node App.js

### Verificar la conectividad:

Comprueba que el servidor se esté ejecutando correctamente y que pueda conectarse a la base de datos. Prueba las rutas API principales para asegurarte de que todo funcione como se espera.

## Reconfiguración del Frontend en React

### Instalar dependencias:

Navega al directorio del frontend (React) y ejecuta:

npm install

### Configurar el entorno:

Verifica que las variables de entorno del frontend estén correctamente configuradas (por ejemplo, la URL de la API del backend).

### Construir la aplicación:

Para producción, construye la aplicación ejecutando:

npm run build

Esto generará los archivos estáticos necesarios para el despliegue.

### Verificar el despliegue:

Asegúrate de que el frontend esté correctamente desplegado y pueda comunicarse con el backend. Realiza pruebas funcionales para confirmar que la interfaz se comporta según lo esperado.

# Procedimiento del backup

## Backup de la Base de Datos MySQL

### Conectar al servidor MySQL:

Accede al servidor MySQL utilizando un cliente como phpMyAdmin, MySQL Workbench, o la línea de comandos de MySQL.

### Generar el archivo de respaldo:

Para realizar un backup completo de la base de datos, ejecuta el siguiente comando desde la línea de comandos:

mysqldump -u usuario -p shair > /ruta/al/archivo/backup.sql

Ingresa la contraseña cuando se te solicite.

Nota: Asegúrate de que la ruta donde guardes el archivo de respaldo sea segura y esté accesible para futuras restauraciones.

### Verificar el respaldo:

Revisa el archivo .sql generado para asegurarte de que no haya errores. Esto lo puedes hacer abriéndolo en un editor de texto o importándolo en una base de datos de prueba.

## Backup del Código del Servidor Express

### Identificar los archivos importantes:

Incluye todos los archivos y carpetas relevantes del proyecto Express, como package.json, app.js, rutas, controladores, middlewares, y el archivo .env.

### Crear un archivo comprimido:

Utiliza un comando como zip o tar para comprimir el directorio del backend:

tar -czvf backup\_servidor\_express.tar.gz /ruta/al/proyecto/express

Este archivo comprimido contendrá todos los archivos necesarios para restaurar el servidor.

### Almacenar el backup:

Guarda el archivo comprimido en un lugar seguro, preferiblemente en una ubicación distinta al servidor de producción, como un almacenamiento en la nube o un servidor de backup.

Backup del Código del Frontend en React

### Identificar los archivos importantes:

Asegúrate de incluir todo el directorio del proyecto React, incluyendo src/, public/, package.json, y otros archivos de configuración como .env.

### Crear un archivo comprimido:

Comprime el directorio del frontend utilizando un comando como zip o tar:

tar -czvf backup\_frontend\_react.tar.gz /ruta/al/proyecto/react

### Almacenar el backup:

Al igual que con el servidor Express, guarda el archivo comprimido en un almacenamiento seguro.

# Scripts de prueba

## Configuración de Cypress

### Instalar Cypress:

Primero, instala Cypress en tu proyecto:

npm install cypress --save-dev

### Iniciar Cypress:

Una vez instalado, puedes abrir Cypress ejecutando:

npx cypress open

Esto creará automáticamente la estructura de carpetas de Cypress en tu proyecto.

## Escribir Pruebas para el Frontend en React

### Crear un archivo de prueba:

Dentro de la carpeta cypress/e2e/, crea un archivo para tus pruebas, por ejemplo, frontend\_spec.cy.js.

## Escribir Pruebas para el Backend en Express

Aunque Cypress está más orientado a pruebas de frontend, también puedes realizar pruebas que interactúen con el backend a través de las rutas API.

### Crear un archivo de prueba para el backend:

Puedes usar el mismo archivo de prueba o crear uno nuevo en cypress/e2e/backend\_spec.cy.js.

## Ejecutar las Pruebas

Puedes ejecutar las pruebas directamente en el navegador a través de la interfaz de Cypress o correrlas en modo headless (sin interfaz gráfica) usando:

npx cypress run

Esto es útil para integrar las pruebas en un pipeline de CI/CD.