

LÓGICA CÓDIGO:

1. Diseño estructura de BBDD para almacenar info de: 1- USERS, 2- SERVICIOS REQUERIDOS
 1. MySQL
 2. Tablas con datos de cada database
 3. Relacionar tablas id - foreign key
2. API REST para hacer CRUD en la BBDD.
 1. Express para la API
 2. Definir rutas y controladores para cada CRUD (Rutas: urls que los users pueden visitar para hacer una acción, controladores: funciones que se ejecutan para procesar las acciones)
 3. Usar verbos HTTP para el CRUD (GET, POST, PUT, DELETE).
 4. URLs únicas y que representen un recurso de la BBDD.
3. Instalar módulos necesarios: mysql2, express, etc
4. Autenticación y autorización: login y roles (anon, registrado):
 1. Middleware passport??? para autenticación y definir roles (Passport es un middleware popular de Node, proporciona estrategias de autenticación incluyendo inicio de sesión con correo y contraseña, google o FB, etc.
 2. Crear ruta para el inicio de sesión y otra para registro de users.
 3. Encriptar para almacenar contraseñas (bcrypt).
 4. Almacenar los datos de inicio de sesión y registro en la base de datos y permitir que los usuarios accedan a la funcionalidad completa de la aplicación.
5. Funcionalidad: visualizar lista de servicios por los anons y que puedan iniciar sesión o registrarse.
 1. Ruta para la lista de servicios y ruta para cada servicio en particular
 2. Anons pueden iniciar sesión/registrarse desde esas páginas.
6. Funcionalidad: nuevo servicio por los users registrados.
 1. Ruta para users registrados accedan a crear un servicio (pedir autenticación e inicio de sesión)
 2. Formulario: crear servicio requerido: título, descripción, archivo
7. Funcionalidad: escoger servicio de otro user y hacer comentarios, subir fichero y marcar como resuelto?
 1. Ruta para que los users vean toda la info de un servicio requerido y elegirlo.
 2. Solo registrados para acceder y con inicio de sesión
 3. Formulario para que hagan comentarios (o chat?) y subir archivo resuelto
 4. Archivos: almacenamiento y solo accesibles por el propietario del servicio requerido
8. Funcionalidad: marcar ya resuelto
 1. Ruta para que users marquen servicio como resuelto
 2. Al marcar como resuelto: desaparece de la lista de servicios disponibles
 3. Aparece en el perfil del user como servicio que ha hecho????? Bonus
9. Opcional: Funcionalidad de gestión del perfil de user
 1. Ruta: users registrados con inicio de sesión accedan a su info personal + foto, biografía etc
 2. Crear un controlador para manejar las solicitudes entrantes en esta ruta y realizar operaciones CRUD en los modelos de datos de Usuario.
 3. Formulario para cambiar datos? Y se guarden en la BBDD
 4. Middleware para proteger la ruta que solo debe ser accesible para el propietario del perfil de usuario