Imagen que contiene Icono

Descripción generada automáticamenteLogotipo

Descripción generada automáticamente

Proyecto: Programa de Reciclaje Comunitario

Semana: 8

**Nombre de estudiantes:**

Beatriz Adriana Quijada\_61851282

**Sede de estudio:**

Ceutec SAP

**Docente:** Elvin Bobadilla

**Sección:** 74

**Fecha de entrega: 12/02/24**

## 1. Objetivos del Sistema

* Plataforma Interactiva: Crear una plataforma en línea que facilite la participación de la comunidad, permitiendo a los residentes registrar y seguir su contribución al reciclaje, así como acceder a información relevante sobre prácticas sostenibles.
* Campañas Educativas: Desarrollar campañas de concienciación a través de la plataforma para informar a la comunidad sobre los beneficios del reciclaje, proporcionando recursos educativos, tutoriales y noticias relacionadas con la conservación del medio ambiente.
* Sistema de Recompensas: Implementar un sistema de recompensas para incentivar la participación continua en el programa de reciclaje, reconociendo y premiando a aquellos que contribuyen de manera significativa a la causa ambiental.

## 2. Alcance del Sistema

Defina el alcance en base a los productos que entregará. (Sistema, documentos, presentación)

1. **Sistema de Participación:**
   * **Productos Entregables:**
     + Plataforma web interactiva accesible desde navegadores.
     + Sistema de registro de usuarios con perfiles individuales.
     + Funcionalidades para que los residentes registren y gestionen su contribución al reciclaje.
     + Herramientas de seguimiento de participación y estadísticas personales.
2. **Documentos Educativos:**
   * **Productos Entregables:**
     + Documentos educativos sobre los beneficios del reciclaje.
     + Tutoriales interactivos en formato multimedia.
     + Sección de noticias y actualizaciones sobre prácticas sostenibles y conservación del medio ambiente.
3. **Campañas Educativas:**
   * **Productos Entregables:**
     + Campañas multimedia a través de la plataforma.
     + Contenido visual y de video educativo.
     + Calendario de campañas y eventos de concienciación.
4. **Sistema de Recompensas:**
   * **Productos Entregables:**
     + Sistema de puntos para registrar y cuantificar la participación.
     + Catálogo de recompensas disponibles.
     + Módulo de gestión de recompensas para la asignación y seguimiento.
     + Certificados o reconocimientos virtuales para los participantes destacados.
5. **Presentación y Capacitación:**
   * **Productos Entregables:**
     + Presentación de la plataforma a la comunidad.
     + Sesiones de capacitación para explicar el funcionamiento del sistema y las recompensas.
     + Material de capacitación en formato digital.
6. **Mantenimiento y Soporte:**
   * **Productos Entregables:**
     + Documentación de mantenimiento y actualizaciones periódicas.
     + Sistema de soporte para resolver consultas y problemas técnicos.

## 3. Plan de Trabajo

1. **Semana 1-2: Configuración del Entorno y Desarrollo Backend:**
   * Configuración de Node.js, Express y MongoDB.
   * Desarrollo de modelos y rutas para usuarios y participación.
2. **Semana 3-4: Desarrollo Frontend con Angular:**
   * Implementación de servicios para la comunicación con el backend.
3. **Semana 5-6: Campañas y Recompensas:**
   * Desarrollo de funcionalidades relacionadas con campañas y recompensas.
4. **Semana 7-8: Pruebas y Optimización:**
   * Implementación de pruebas unitarias y de integración.
   * Optimización de rendimiento y corrección de errores.
5. **Semana 9: Despliegue y Entrenamiento:**
   * Despliegue del sistema en un entorno de producción.
   * Sesiones de entrenamiento para usuarios y administradores.

## 3. Requerimientos para el sistema.

Elabore un Modelo de Requerimientos en el que defina en forma general cada uno de los requerimientos planteados-funcionales. Enfóquese en las salidas y las entradas que tendrá el sistema. Elabore los modelos de Casos de uso necesarios para satisfacer los requerimientos planteados del Sistema. elaboración de la interfaz (lado del cliente).

**Modelo de Requerimientos - Funcionales:**

1. **Registro de Participantes:**
   * **Entradas:**
     + Datos del usuario (nombre, dirección, correo electrónico).
   * **Salidas:**
     + Confirmación de registro.
2. **Sistema de Puntos y Contribuciones:**
   * **Entradas:**
     + Registro de acciones de reciclaje por parte de los participantes.
   * **Salidas:**
     + Actualización del puntaje del participante.
     + Estadísticas de participación.
3. **Contenido Educativo y Campañas:**
   * **Entradas:**
     + Acceso a documentos educativos y tutoriales.
   * **Salidas:**
     + Visualización de contenido educativo.
     + Información sobre campañas y eventos.
4. **Sistema de Recompensas:**
   * **Entradas:**
     + Selección de recompensas por parte de los participantes.
   * **Salidas:**
     + Confirmación de selección.
     + Certificados virtuales.
5. **Interfaz de Usuario Intuitiva:**
   * **Entradas:**
     + Interacción del usuario con la plataforma.
   * **Salidas:**
     + Diseño de interfaz fácil de usar.
     + Confirmación de acciones.

**Casos de Uso:**

1. **Registrar Participante:**
   * **Entradas:**
     + Datos del usuario.
   * **Salidas:**
     + Confirmación de registro.
2. **Realizar Acción de Reciclaje:**
   * **Entradas:**
     + Tipo de material reciclado, cantidad.
   * **Salidas:**
     + Actualización del puntaje del participante.
3. **Acceder a Contenido Educativo:**
   * **Entradas:**
     + Selección de documentos o tutoriales.
   * **Salidas:**
     + Visualización del contenido educativo.
4. **Seleccionar Recompensa:**
   * **Entradas:**
     + Elección de una recompensa.
   * **Salidas:**
     + Confirmación de selección.
     + Certificado virtual.

**Elaboración de la Interfaz (Lado del Cliente):**

1. **Página de Inicio:**
   * **Componentes:**
     + Formulario de registro.
     + Destacados de campañas y recompensas.
2. **Perfil del Participante:**
   * **Componentes:**
     + Información del usuario.
     + Puntaje y estadísticas de participación.
     + Acciones de reciclaje recientes.
3. **Contenido Educativo:**
   * **Componentes:**
     + Sección con documentos y tutoriales.
     + Calendario de campañas.
4. **Catálogo de Recompensas:**
   * **Componentes:**
     + Lista de recompensas disponibles.
     + Detalles de cada recompensa.
5. **Registro de Acción de Reciclaje:**
   * **Componentes:**
     + Formulario para ingresar tipo y cantidad de material reciclado.
6. **Certificados:**
   * **Componentes:**
     + Certificados virtuales.

## 4. Conjunto de tablas-colecciones de la base de datos.

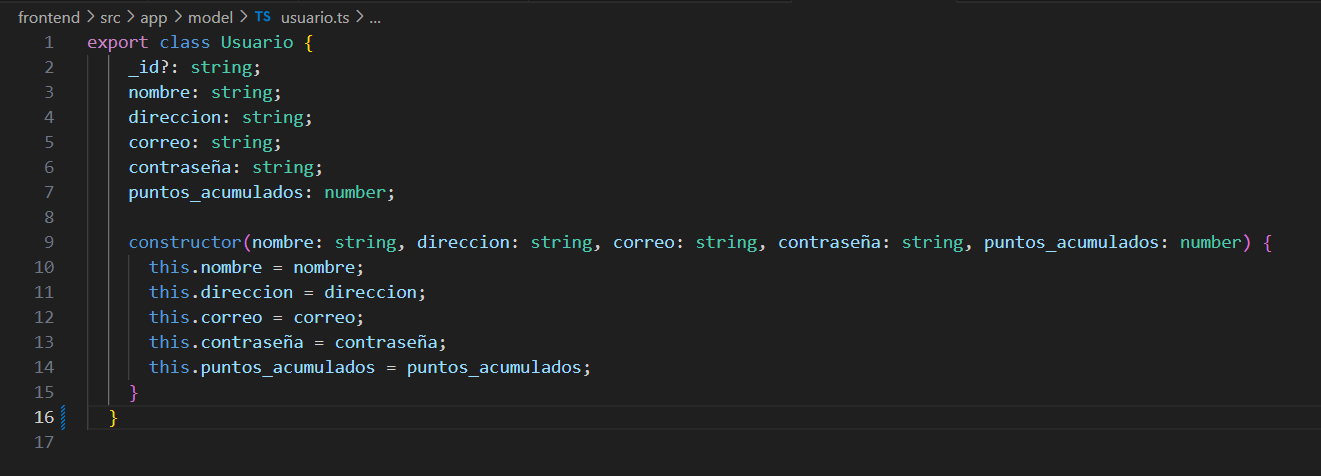
Elabore el diagrama de las tablas-colecciones indexadas de la base de datos que utilizara en el Sistema y que servirá de base para la integración de los datos.

Diagrama

Descripción generada automáticamente

## Modelo de Clases.

Desarrolle el modelo de clases, defina los atributos y operaciones que pertenecen a cada clase. También debería plantear las clases asociación del lado del servidor.



Clase Usuario:

* Atributos:
  + id: Identificador único del usuario.
  + nombre: Nombre del usuario.
  + correoElectronico: Correo electrónico del usuario.
  + direccion: Dirección del usuario.
  + Contraseña: contraseña del usuario
  + puntosAcumulados: Puntos acumulados por el usuario por participar en acciones de reciclaje.
* Operaciones:
  + registrar(): Método para registrar un nuevo usuario en el sistema.
  + iniciarSesion(): Método para que un usuario inicie sesión en el sistema.
  + obtenerPuntosAcumulados(): Método para obtener los puntos acumulados por el usuario.

Clase AccionReciclaje:

* Atributos:
  + id: Identificador único de la acción de reciclaje.
  + idUsuario: Identificador del usuario que realizó la acción.
  + tipoMaterial: Tipo de material reciclado (papel, plástico, vidrio, etc.).
  + cantidad: Cantidad de material reciclado en la acción.
  + fecha: Fecha en que se realizó la acción de reciclaje.
* Operaciones:
  + registrar(): Método para registrar una nueva acción de reciclaje.
  + obtenerPorUsuario(idUsuario): Método para obtener todas las acciones de reciclaje realizadas por un usuario específico.

Clase Recompensa:

* Atributos:
  + id: Identificador único de la recompensa.
  + nombre: Nombre de la recompensa.
  + puntosRequeridos: Puntos requeridos para canjear la recompensa.
* Operaciones:
  + listar(): Método para listar todas las recompensas disponibles.
  + canjear(idUsuario, idRecompensa): Método para que un usuario canjee una recompensa específica.

Clase CampanaEducativa:

* Atributos:
  + id: Identificador único de la campaña educativa.
  + nombre: Nombre de la campaña educativa.
  + descripcion: Descripción de la campaña educativa.
  + fechaInicio: Fecha de inicio de la campaña educativa.
  + fechaFin: Fecha de finalización de la campaña educativa.
* Operaciones:
  + listar(): Método para listar todas las campañas educativas disponibles.
  + participar(idUsuario, idCampana): Método para que un usuario participe en una campaña educativa.

Clase RegistroActividad:

* Atributos:
  + id: Identificador único del registro de actividad.
  + idUsuario: Identificador del usuario asociado al registro de actividad.
  + actividad: Descripción de la actividad realizada.
  + fecha: Fecha en que se realizó la actividad.
* Operaciones:
  + registrar(): Método para registrar una nueva actividad en el sistema.
  + obtenerPorUsuario(idUsuario): Método para obtener todas las actividades registradas por un usuario específico.

**Asociaciones del Lado del Servidor:**

* Usuario tiene muchas AccionReciclaje (relación uno a muchos).
* Usuario tiene muchas RegistroActividad (relación uno a muchos).
* AccionReciclaje pertenece a Usuario (relación muchos a uno).
* Recompensa tiene muchas Usuario (relación muchos a muchos).
* CampanaEducativa tiene muchas Usuario (relación muchos a muchos).

## 6. Glosario del Proyecto.

Detalle la definición que tiene cada uno de los términos utilizados en los procesos del sistema y que son específicos de éste. Puede definir varias categorías.

NgModule: Definición: NgModule es una clase decoradora en Angular que se utiliza para definir un módulo, que agrupa componentes, directivas, servicios y otros constructores relacionados que forman un contexto de compilación y tiempo de ejecución para una aplicación de Angular.

Await: Definición: La palabra clave 'await' se utiliza en JavaScript para pausar la ejecución de una función asíncrona hasta que una promesa sea resuelta, devolviendo el valor resuelto de la promesa.

TestBed: Definición: TestBed es una utilidad de Angular para configurar y manipular entornos de prueba para componentes de Angular, servicios y otros constructores.

FormGroup: Definición: FormGroup es una clase en Angular que representa un grupo de controles de formulario, como FormControl, y proporciona métodos para trabajar con ellos como un conjunto.

Fixture: Definición: En el contexto de pruebas unitarias en Angular, Fixture es un objeto que proporciona acceso al componente bajo prueba y permite realizar acciones y aserciones en él durante las pruebas.

SubmitForm: Definición: SubmitForm es una función o método que se utiliza para enviar datos de un formulario HTML a un servidor o para realizar alguna acción específica en respuesta a la presentación del formulario.

OnInit: Definición: OnInit es una interfaz en Angular que define un método 'ngOnInit()', el cual se ejecuta una vez que Angular ha inicializado completamente un componente.

Private fb: Definición: 'private fb' probablemente se refiere a la convención de nomenclatura comúnmente utilizada en Angular para hacer referencia a un FormGroup, donde 'fb' representa 'FormBuilder', una utilidad de Angular para construir instancias de FormGroup.

ngOnInit: Definición: ngOnInit es un ciclo de vida de componente en Angular que se ejecuta después de que Angular ha inicializado completamente las propiedades del componente.

Navigate: Definición: Navigate es un término que se utiliza comúnmente en aplicaciones web para referirse al proceso de cambiar de una página a otra dentro de la misma aplicación, generalmente utilizando enlaces, botones o funciones de navegación programáticas.

forRoot: Definición: forRoot es un método estático utilizado en la definición de módulos de Angular para proporcionar configuraciones globales a un módulo y sus dependencias.

# Bibliografía

Angular - NgModule documentation (<https://angular.io/guide/architecture-modules>)

MDN Web Docs - await (<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/await>)

Angular - TestBed documentation (https://angular.io/api/core/testing/TestBed)

Angular - FormGroup documentation (<https://angular.io/api/forms/FormGroup>)

Angular - ComponentFixture documentation (<https://angular.io/api/core/testing/ComponentFixture>)

Angular - OnInit documentation (<https://angular.io/api/core/OnInit>)

Angular - FormBuilder documentation (https://angular.io/api/forms/FormBuilder)

Angular - ngOnInit documentation (<https://angular.io/api/core/OnInit>)

Angular - Configuring NgModule (<https://angular.io/guide/providers#creating-a-core-module-with-services>)