**“GreenTrack” —Hábitos Ecológicos**

Una aplicación backend para registrar y seguir hábitos sostenibles de personas que desean reducir su huella ecológica. Usuarios que registran actividades como ir en bicicleta, reducir plásticos, plantar árboles, reciclar, etc.

**Concepto**

Los usuarios pueden registrar acciones ecológicas y ver su impacto verde estimado basándonos en una puntuación numérica básica.

Los administradores podrán dar de alta o eliminar nuevas acciones.

**Modelos**

**1. User (clase padre)**

* Atributos: id, name
* Herencia:
  + Regular**User** , Admin**User**

**2. EcoAction**

* Atributos: id, actionType, , date
* Relación ManyToOne con RegularUser

**3. ActionType**

* Atributos: id, name, greenImpact
* Enum: “BICICLETA”, “RECICLAJE”, “REUTILIZAR\_BOTELLA”, “PLANTAR\_ARBOL”
* Relación OneToMany con EcoAction

**Clase EcoAction**

La clase **EcoAction** representa una acción ecológica genérica realizada por un usuario. Esta extende a acciones específicas como en desplazarse en bicicleta, limpiar la costa marítima o plantar un árbol. Tiene una fecha donde se registra la acción ecológica, además, se vincula a un **usuario** que ha realizado la acción.

Atributos:

* **id** (Integer): Identificador único de la acción ecológica en la base de datos.

**date** (LocalDate): Fecha en la que se realiza la acción ecológica. Este atributo es importante para hacer un seguimiento cronológico de las acciones.

* **Description :** Nombre de la acción ecológica
* **greenImpact**: Valor numérico que indica el impacto verde de la eco acción.
* **user** (User): Relación de tipo **ManyToOne** con la clase User, lo que significa que una

**EcoAction** está asociada a un solo usuario, pero un usuario puede realizar varias acciones ecológicas.

**Especializaciones de EcoAction**

Cada acción específica es una especialización de la clase abstracta EcoAction. Este diseño permite especializar cada acción ecológica con sus cualidades y características propias de cada acción ecológica determinada.

**To Cycle**:

* **km** (Integer): Cantidad de kilómetros recorridos
* **origin (String):** Origen del desplazamiento.
* **Destiation(String):** Destino del desplazamiento

**CleanShoreLine:**

* **km** (Integer): Cantidad de kilómetros de costa limpiados
* **locate (**String**):** lugar de recogida de residuos
* **hours (**Integer**):** Horas invertidas en la recogida.

Diagrama

El contenido generado por IA puede ser incorrecto.Diagrama

El contenido generado por IA puede ser incorrecto.

