

TRABAJO PRÁCTICO INTEGRADOR ANUAL

*TEMA: Monitoreo de Estado de Servicios de Transporte Público y
de Establecimientos*

Grupo N°12

Apellido, Nombre	Legajo
Condori, Joselin Magali	203.520-0
de los Ríos, Carolina	213.242-4
Kassem, Agustin	213.248-5
Piccinnino, Franco Ezequiel	203.958-8
Vazquez, Juan Martin	204.144-3

Fecha Presentación: 06/06/2023

Profesores: Ezequiel Escobar

Gaston Prieto

Ayudante: Uriel Soifer

Clases

Entidad

- Representa tanto a los servicios de transporte como a las organizaciones. Esto se determina desde su tipo de entidad (tipoEntidad).
- Esta clase contendrá una lista de los establecimientos que pertenecen a dicha entidad.
- Además, posee una localización, un estado y un nombre (String).

TipoEntidad

- Es la categoría a la que pertenece un entidad, por ejemplo: tren, subte, bancario, restaurante, etc.
- Tendrá instancias ya parametrizadas y otros tipos se podrán ir agregando en tiempo de ejecución, es decir, es extensible.

Establecimiento

- Representa tanto a las estaciones (en el caso de los servicios de transporte) como a las sucursales (en el caso de las organizaciones).
- Esta clase contendrá una lista de los servicios que se encuentran en dicho establecimiento.
- Además, posee una dirección, un estado y un nombre (String).

Servicio

- Tiene un nombre (String), una descripción (String) y un estado.

Estado

- Es el estado asociado que tienen las entidades, los establecimientos y los servicios.
- No es extensible al ser una enumeración, pero los requerimientos actuales no requieren que lo sea.

Usuario

- Tiene un atributo booleano denominado “esValido” que se determina desde el Validador utilizando el usuario y la contraseña.
- Cuando un usuario desee unirse a una comunidad, se creará una instancia de Miembro que tendrá asociado al usuario en cuestión.
- Contiene todos sus intereses en su atributo “interés”.
- Tiene un rol que le da permisos en el sistema.
- Además, posee una lista de roles, un nombre (String) y un apellido (String).

Miembro

- Contiene un atributo comunidad que apunta a Comunidad de la que es miembro.
- Tiene un rol que le da permisos de la comunidad a la que pertenece.

Rol

- Representa a un conjunto de permisos que se relacionan entre sí.
- Además, posee una descripción para comprender su función.

Permiso

- Es la credencial que se necesita para realizar una acción determinada.
- Además, posee una descripción para comprender su función.

Interés

- Representa las localizaciones, los servicios y las entidades que le son de atención para un usuario.
- Decidimos hacerlo en una clase aparte ya que puede tener varias entidades y servicios de interés y así le damos la posibilidad de aumentar la expresividad y la cohesión.
- Posee la localización que tendrá de interés

Comunidad

- Una comunidad está conformada por una lista de Miembros y una lista de Administradores, los segundos poseen un rol con los permisos necesarios para administrar la comunidad.
- Además, posee un nombre (String).

EntidadPrestadora

- Representa a las empresas encargadas de la prestación de servicios en una o en más entidades.
- Tiene un representante (Usuario) encargado de recibir las problemáticas que se presentan en las entidades que administra.
- Además, tiene una lista de entidades y un nombre (String) que la identifica.

Organismo de Control

- Representa a las organizaciones encargadas de controlar a las entidades prestadoras.
- Tiene un representante (Usuario) encargado de recibir las problemáticas que se presentan en las entidades que administra.
- Además, tiene una lista de entidades prestadoras y un nombre (String) que la identifica.

Validador de Contraseña

- Con esta clase vamos a validar todas las contraseñas de las cuentas de los usuarios en el momento de registrarse.
- Cumple el rol de activador en el patrón Command.

Validador

- Es una interfaz que se encarga de implementar cada una de las validaciones de las clases ValidadorLongitudContrasenia, Validador10kContrasenia, ValidadorCaracter, ValidadorContraseniaNoContieneUsuario con un método esValida(). Utilizamos un patron command porque todas las validaciones que agregamos son acciones que deben activarse al comprobar un usuario y una contraseña.
- Es una clase extensible porque podemos crear más subclases que hereden los atributos y métodos con usos diferentes.
- Utilizando este patrón logramos que se puedan llegar a reutilizar los módulos ante un posible cambio. También pensamos que esto lo hace un sistema mantenible y extensible porque si llegamos a modificar alguno de estos módulos no vamos a tener que hacer tantos cambios en los demás.
- Cumple el rol Command en el patrón homónimo.

ValidadorLongitudContrasenia

- Implementa la clase Validador
- Su método esValida() verifica que la contraseña tenga más de 8 caracteres.
- Cumple el rol de CommandConcreto en el patrón Command

Validador10kContrasenia

- Implementa la clase Validador.
- Su método esValida() verifica que no se encuentre dentro de la lista de 10 mil contraseñas comunes.

ValidadorCaracteres

- Implementa la clase Validador.
- Su método esValida() controla si los caracteres de la contraseña son validos.
- Cumple el rol de CommandConcreto en el patrón Command.

ValidadorContraseniaNoContieneUsuario

- Implementa la clase Validador
- Su método esValida() controla que la contraseña no contenga al usuario.
- Cumple el rol de CommandConcreto en el patrón Command.

Localización

- Es una interfaz que define el método mostrarNombre()

- Provincia: clase que implementa la interfaz localización
- Municipio: clase que implementa la interfaz localización
- Departamento: clase que implementa la interfaz localización
- Dirección: Es una dirección precisa con una calle y una altura, ubicadas en una provincia, departamento y un municipio específico.

GeorefService

- Es una interfaz que define las operaciones HTTP.
- Cada método representa una posible llamada a la API.
- El valor de retorno de cada método es Call(<Clase>), donde Clase es la clase molde.

Carga de datos CSV

Formato:

1	NOMBRE_PRESTADOR	NOMBRE_USUARIO	APELLIDO_USUARIO	MAIL_USUARIO	USER_NAME	CONTRASENIA	TIPO_PRESTADOR
2	Santander	juan	hola	juan@mail.com	juan	asjfkajsflk123134234	ENTIDAD_PRESTADORA
3	LineaB	joselin	chau	joselin@mail.com	joselin	josjos1234	ENTIDAD_PRESTADORA
4	CNRT	armando	hola	armando@mail.com	armando	juanjuan1234	ORGANISMO_CONTROL

La primera columna nos dará el nombre de la entidad prestadora u organismo de control, y la última columna nos dirá cual de las dos clases debemos instanciar (organismo de control o entidad prestadora). Mientras que las columnas del medio las utilizaremos para instanciar la clase usuario que representará el representante de la clase que instanciemos.