



Consignas

Proyecto

Deberá nombrar el proyecto con la siguiente estructura:

- PB2-Final20220726-ApellidosNombres.

Enunciado

Central de Alarmas

Un centro de seguridad que posee muchas alarmas, permite ingresar usuarios los cuáles serán configuradores o activadores.

Los usuarios poseen:

- DNI.
- Nombre.

Cada alarma está definida por:

- ID de alarma (numérico).
- Código de activación/desactivación.
- Código de configuración.
- Nombre.
- Lista de usuarios válidos para operar sobre la misma.
- Lista de acciones realizadas. Una acción está formada por:
 - Identificador de acción.
 - La alarma sobre la que se realiza la acción.
 - El usuario que realiza la acción.
 - Fecha en la que se realiza.
 - Tipo de operación: CONFUGRACION, ACTIVACION, DESACTIVACION.
- Lista de sensores. Cada sensor posee:
 - Identificador numérico.
 - Estado (boolean). Ejemplo: estará en verdadero cuando se active.

Cada alarma deberá registrar las acciones realizadas.

Desarrollar una interfaz “Configurable” que se implemente por los usuarios configuradores y una interfaz “Activable” implementada por los usuarios activadores.



El usuario administrador podrá realizar las siguientes operaciones sobre la central de alarmas:

- Registrar una alarma en la central.
 - `public boolean agregarAlarma(Alarma alarma).`
- Registrar un nuevo usuario configurador o activador en la central.
 - `public boolean agregarUsuario(Usuario usuario).`
- Agregar un usuario a la lista de usuarios válidos de una alarma.
 - `public void agregarUsuarioAUnaAlarma(int dniUsuarioAAgregar, int idAlarma, String codigoConfiguracionAlarma).`
- Agregar un sensor a una alarma.
 - `public boolean agregarSensorAAlarma(int idAlarma, String codigoConfiguracionAlarma, Sensor sensor, idUsuarioConfigurador)`
- Activar un sensor de alarma.
 - `public boolean activarSensorDeAlarma(int idSensor, int idAlarma, String codigoActivacionAlarma).`
- Activar/desactivar una alarma. Importante: La alarma se activará sólo cuando todos los sensores se encuentren activados.
 - `public boolean activarDesactivarAlarma(int idAlarma, String codigoActivacionAlarma, Configurador usuario).`

Los usuarios configuradores podrán:

- Agregar un usuario a la lista de usuarios válidos de una alarma determinada. Para lograr esta operación de deberá pasar por parámetro:
 - DNI del usuario a agregar.
 - Identificador de la alarma.
 - Código de configuración de la alarma.
 - Este método debe lanzar una excepción `CodigoAlarmaIncorrectoException` en caso de no coincidir el código de configuración ingresado con el de la alarma.
- Agregar sensor a una alarma indicando:
 - Identificador de la alarma.
 - Código de configuración de la alarma.
 - Sensor que se agrega a la lista de sensores.
 - Este método debe lanzar una excepción `SensorDuplicadoException` en caso de ya existir un sensor con el mismo identificador.
- Activar un sensor de alarma indicando:
 - Identificador del sensor.
 - Identificador de la alarma .
 - Código de configuración de la alarma.

Cada operación debe registrar la acción realizada sobre la alarma con tipo "CONFIGURACION".



Los usuarios activadores podrán:

- Activar o desactivar una alarma:
 - `public boolean activarDesactivarAlarma(Alarma alarma, String codigoActivacionAlarma).`

Cada operación debe registrar la acción realizada sobre la alarma con tipo "ACTIVACION" o "DESACTIVACION" según corresponda.

Tests a desarrollar

1. `queSePuedaRegistrarUnaAlarmaEnLaCentral().`
2. `queSePuedaAgregarUnUsuarioConfiguradorAUnaAlarma().`
3. `alAgregarUnUsuarioAUnaAlarmaConCodigoDeConfiguracionDeAlarmaInvalidoSeLanceCodigoAlarmaIncorrectoException().`
4. `alAgregarUnSensorDuplicadoEnUnaAlarmaSeLanceUnaSensorDuplicadoException().`
5. `queNoSePuedaActivarUnaAlarmaSiHayAlMenosUnSensorDesactivado().`
6. `queParaUnaAlarmaDeterminadaSePuedaObtenerUnaColeccionOrdenadaDeAccionesDeTipoConfiguracionOrdenadasPorIdDeAccion().`

Entrega

Se deberá comprimir el proyecto y entregarlo en la plataforma MleL, en la práctica determinada para tal fin.