Materia: Seminario de Lenguajes
Lic. En sistemas

Proyecto -FINAL Libre-

Sistema de Viviendas Recicladoras

Una cooperativa de trabajo, dedicada a la recolección de residuos reciclables, de la ciudad de Viedma quiere implementar un sistema. La misma desea tener un registro de las viviendas que estén dispuestas a separar residuos reciclables.

Requerimientos Funcionales

Los registros de las viviendas se podrán realizar cualquier día del año. Se deberá registrar fecha y hora de registro, datos del dueño (nombre, apellido, dni, etc.) y la ubicación determinada por la dirección, zona, barrio y latitud/longitud de la misma.

Por otra parte, la cooperativa desea tener un registro de su personal. De estos recolectores se desea contar con su nombre, apellido y DNI.

Los ciudadanos podrán informar a la cooperativa, a través del sistema, cuando cuenten con residuos en sus viviendas para ser retirados. Realizando un pedido de retiro el cual debe contar con los siguientes datos: fecha de emisión, residuos a retirar, si se requiere vehículo de carga pesada y una observación.

De esta manera se busca administrar las órdenes de retiros las cuales son generadas a partir de los pedidos. En las órdenes de retiro se establece que recolector retirará los residuos, la vivienda a visitar, el día y hora en la que se genera y un estado (PENDIENTE, EN_EJECUCION, CONCRETADO, CANCELADO).

Estas órdenes de retiro serán tomadas por los recolectores de la cooperativa, los cuales se encargarán de realizar las visitas. De cada visita se registra el dia y la hora, datos del retiro (cantidad de residuos en kg y tipo) y una observación. Tener en cuenta que cada orden puede contar con más de una visita, es decir, que se podría tener la siguiente información:

OrdenDeRetiro1 – Vivienda: 1. Recolector: Juan Perez. (Estado: PENDIENTE):
<sin visitas>

OrdenDeRetiro2 – Vivienda: 2. Recolector: Juan Perez (Estado: EN_EJECUCION):
Visita 1: Fecha: 04/06/2018 08:30 –
Obs.:No se encontraba en el domicilio.
Visita 2: Fecha: 05/06/2018 09:00–
Obs.: Se retiraron residuos pero resta completar el traslado.
Visita n: ...

Además, los residuos estarán discriminados por tipo, por ejemplo, plástico, papel y cartón, vidrio, metal, etc. Donde cada tipo contará con un puntaje por kg.

El municipio lanzó una campaña denominada "Municipio Club (Mc)" permitiendo al ciudadano(dueño) acumular puntos de acuerdo a los órdenes de retiro con estado CONCRETADO que se realice para este, estableciendo ventajas sobre el mismo, proporcionando premios como reducción de impuestos, reconocimiento al ciudadano, etc.
Se deberá llevar a cabo una funcionalidad en el sistema que permita conocer la cantidad de puntos que acumuló cada ciudadano y la posibilidad de canjear los puntos en un catálogo que brindará el sistema.

Por ejemplo el Catálogo se podrá definir de la siguiente manera:

Descripción del Premio	Puntos (mayor e igual a:)
Reducción de impuesto ABL- 15%	30
Descuento del 50% en Talleres brindado por el municipio	50
Boleto gratis ida/vuelta a El Cóndor.	20
Pedido de Poda y retiro de árboles gratis	80

Materia: Seminario de Lenguajes
Lic. En sistemas

Requerimientos No Funcionales

Hay que tener en cuenta lo siguiente:

- No acoplar el sistema a un tipo de Vista específica (Swing, Web o Mobile). Se tendrá la posibilidad en algún otro momento requerir la vista como sistema web. Seguir la arquitectura definida en otra sección del documento.
- No acoplar el sistema a un idioma específico. Podríamos implementarlo en países de habla inglesa, por ejemplo. Utilizar ResourceBundle.
- No acoplar el sistema a una configuración de base de datos (usuario/password, string de conexión, etc). Utilizar Properties.

Para poder realizar un buen diseño y cumplir con los puntos antes mencionados, implemente el sistema bajo los siguientes lineamientos:

1. **Back-end**: Implemente el modelo de dominio en objetos y una API para exponer los servicios del modelo. Cualquier tipo de servicio que la Vista requiera, deberá exponerse en esta API.

Ejemplos de la API:

```
api#nuevoVivienda(String identificador, String descripcion);
apis#traerVivienda(String identificador);
```

- Utilice interfaces donde sea necesario para independizar al modelo de dominio del mecanismo de acceso a la base de datos. Utilizaremos JDBC para conectarnos con la base de datos.
- Cada método de la API tiene que estar **documentado con Javadoc**. Y la documentación debe proveer ejemplos de cómo se utiliza la API.
- Validaciones sobre cualquier entrada que se haga. Lance excepciones con mensajes i18n.
- Utilice Excepciones propias.

2. **Front-end**: Implemente la interfaz gráfica del sistema utilizando Swing. La interfaz gráfica debe ser multi-lenguaje. Esto es, se podrá modificar el idioma sin modificar código java.

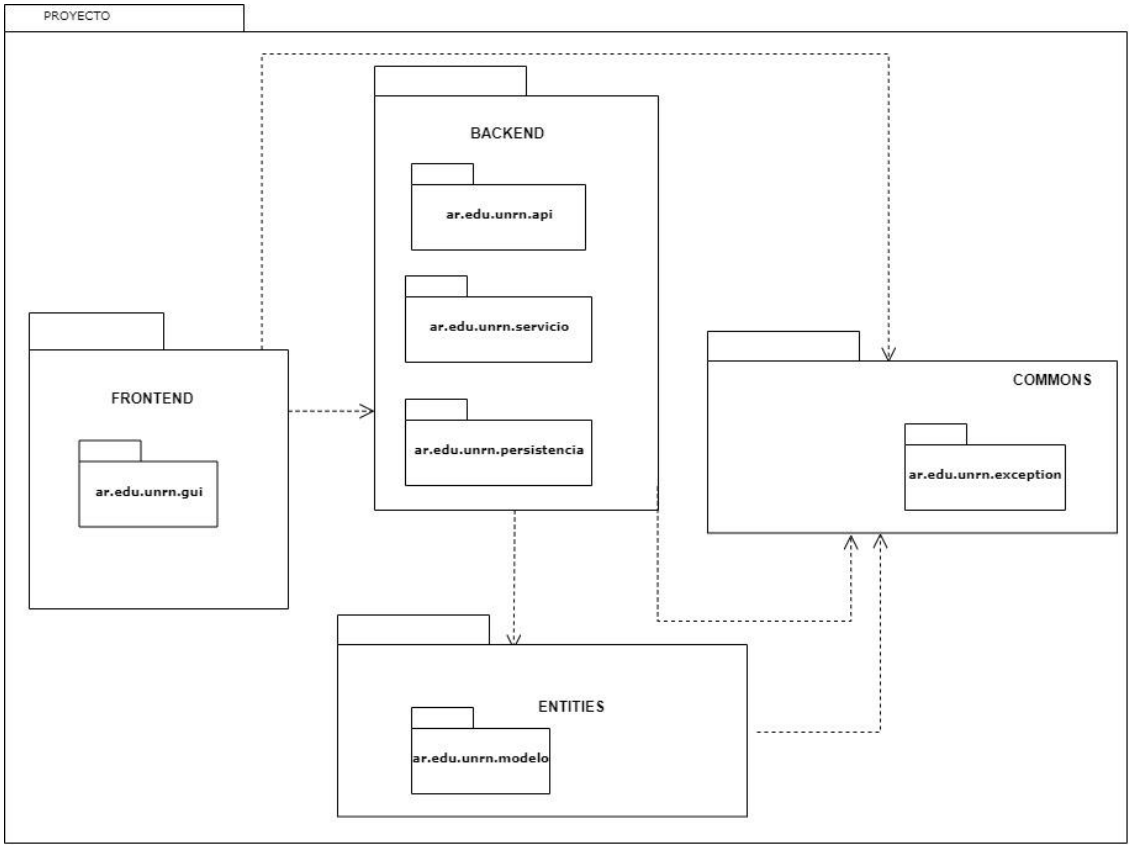
Ayuda técnica

- Utilizaremos ResourceBundles (ver tutorial en la plataforma) para generar Vistas y mensajes de error i18n.

- Utilizaremos System Properties para que el cliente puede configurar el back-end antes de usarlo.

Proyecto Base con la siguiente estructura de proyectos Java:

- proyecto **BACKEND**
- proyecto **COMMONS**
- proyecto **ENTITIES**
- proyecto **FRONTEND**



Materia: Seminario de Lenguajes
Lic. En sistemas

Se pide (justificando sus decisiones):

1. Identificar los objetos y las operaciones esenciales para la resolución de este trabajo. Representar cada clase utilizando UML.
2. Graficar el diagrama UML reflejando las dependencias entre las clases.
3. Identificar las funcionalidades del sistema, escribir los casos de usos, en un template dado.
4. Modelar e implementar en objetos (utilizando el lenguaje Java) los requerimientos expuestos anteriormente.
5. A partir del punto anterior, describir los métodos de la interface API, mencionando: nombre, parámetros, excepciones, etc (firma de método).
6. Crear la clase PersistenceAPI cuyos métodos implementen las funcionalidades.
7. Desarrollar las clases “DAO” para las entidades seleccionadas

- **Demostrar el uso del Framework de colecciones en la solución planteada. (Set, List Map, Stream)**
- **Crear excepciones propias, de acuerdo al dominio del problema y los eventos a notificar.**
- **Hacer uso en general de todos los contenidos dados en la materia: uso de modificadores de acceso, variables y métodos de clase, calificadores (final), , uso y manejo de fecha (Date, o LocalDate), formato de fecha, etc.**