

Proyecto de Gestión Logística

El proyecto de este semestre estará basado en la gestión de un correcto funcionamiento de la **logística** de una empresa portuaria. En nuestro desarrollo deberemos gestionar las **rutas**, **empaquetado** y **facturación** de la mercancía, toda ella transportada en contenedores.

El proyecto está pensado, por su complejidad, para realizarse en **grupos de 3 personas**. Pero puede realizarse en **grupos de 2** o hasta de forma individual si es conveniente por vuestra situación personal/trabajo formar un grupo.

Durante el curso ireis viendo la teoría necesaria, algunos aspectos serán de repaso de asignaturas anteriores (diagramas de clase y casos de uso) o bases de datos, y otras nuevas como puede ser la parte gráfica.

No hace falta que la primera entrega del proyecto incluya todos los aspectos. El proyecto está pensado para hacerse de forma **incremental**. Recomendando primero hacer un **esquema general** y después ir abordando cada una de las funcionalidades. También podéis dejar la parte gráfica para el final (una vez vista la teoría) y concentrarse primero en las funcionalidades que se pueden realizar con los conocimientos que tienen de otras asignaturas. La **separación por capas** se puede hacer inicialmente o hacia el final.

Funcionalidad

Tendremos **5 áreas funcionales o grandes bloques**: Registro al sistema, operaciones, rutas, empaquetado y facturación. Si entramos en detalle con una breve descripción:

- **Registro al sistema**: Se deberá gestionar acciones CRUD de usuarios de acceso al sistema, esta gestión deberá estar asociada a unos roles que nos permitan tener una serie de permisos concretos.
- **Operaciones**: Se definirá las operaciones como elemento base, cada operación tendrá relacionada toda la información de ruta, empaquetado y facturación.
- **Rutas (routing)**: En este bloque el usuario definirá el tipo de ruta (terrestre, marítima o aérea) cada una con un esquema de datos diferentes. Ejemplo: puertos origen y destino, nombre del expedidor, exportador o cargador (shipper), nombre del destinatario o receptor de la mercancía (consignee)...
- **Empaquetado (packing)**: Esta sección se encargará de definir todo aquello relacionado con la mercancía. Ejemplo: número de contenedores relacionados, peso y tamaño del producto a transportar...
- **Facturación (billing)**: En este apartado se llevará la lógica de la facturación a seguir, validaremos datos para poder generar facturas correctamente a partir de líneas pendientes a facturar.

A continuación, se va a detallar cada módulo comentado a grandes rasgos, el alumno deberá entrar en más detalle creando una solución funcionalmente correcta. Es interesante buscar vocabulario dentro del ámbito del transporte de mercancías para así, hablar con propiedad en la terminología empleada de campos.

En el **registro del sistema**, dar de alta a usuarios será una operación sólo disponible para usuarios Administradores. Los cuales tendrán un nombre/identificador (caracteres, mín. 1, máx. 10), clave (caracteres, mín. 4, máx. 8), tipo de usuario según roles, teléfono, y email.

Las **operaciones** son el elemento base, se crea una operación a realizar, se trata de movimiento de producto para gestionar su logística, de ella se relacionará toda la lógica de datos de rutas, empaquetado y facturación. La operación se puede considerar una unidad funcional cerrada.

Al crear una operación se debe asignar el tipo de ruta, a partir de este punto, la lógica y esquema de datos será diferente, igual que las diferentes interfaces a mostrar.

Las **rutas** describirán la trazabilidad de la mercancía entre 2 puntos del planeta, se gestionará tanto las fechas como lugares físicos origen destino. También existirá la posibilidad de que un contenedor cambie de medio de transporte en su trayectoria de entrega. Todo esto tiene que estar bien definido. Tener en cuenta posibles imprevistos como retrasos.

El **empaquetado** contendrá toda la información relativa al producto para su correcta gestión en todo el transporte.

El módulo de **facturación** es el encargado de saber precio final de la operación, cada producto se puede relacionar con un concepto, que este a su vez tiene un coste definido. Se deberá abordar el IVA que irá relacionado al concepto a facturar. Ejemplo: concepto "freight" (sin impuestos).

Especificaciones

- Realizar un diagrama de clases del proyecto.
- Diagrama de casos de uso teniendo en cuenta lógica.
- Acciones según diferentes roles, (administrador, jefe de sección, operario...)
- Guardar mediante logs trazabilidad de operaciones.
- Valorar hacer integraciones con "sistema B".
- Control de excepciones.

Infraestructura

El repositorio de código debe ser privado para terceros y editable para todos los miembros del grupo. El docente tendrá permisos de lectura, el cual será asociado en la plataforma elegida (Github, Bitbucket, etc).

Base de datos

Para dar persistencia a los datos, se usarán una base de datos en la nube. Se recomienda usar AWS. La cuenta estudiante otorga hasta \$50 de crédito gratuito en la modalidad de servidores pequeños, con acceso a bases de datos como MaríaDB o PostgreSQL. En caso de usar otro servicio, éste debe permitir el acceso remoto.

Interfaz gráfica

El programa debe contar con una interfaz gráfica, que puede ser implementada con Java AWT o equivalente.

Capas

La arquitectura del producto debe estar organizada en capas. Se pide una adecuada separación de funcionalidades en presentación, negocio/lógica, y acceso a datos/persistencia.

Entregas

Informe en PDF:

- Portada (asignatura, miembros del grupo, número de entrega [1-4])
- División de tareas: Descripción, tiempo de la entrega, persona responsable
- Datos de acceso al servicio de base de datos en la nube
- Diagramas de clases, casos de uso, esquema de bases de datos
- Descripciones adicionales

Código fuente

Sólo incluir código fuente relevante y correspondiente a cada entrega.

Notas

Antes de empezar a programar hacer un esquema de cómo piensan solucionar el problema y las posibles tareas. Pueden mandármelo por email como entrega 0, para tener feedback.

Hay varias maneras de realizar el proyecto, cualquiera que cumpla con lo que se solicita a nivel funcional y que siga las pautas de documentación y estructura, es válida, siempre que se use Java.

Un punto donde podéis elegir es en el tipo de estructuras de datos (tablas, vectores, listas, etc) que serán necesarios para guardar todos los parámetros del sistema que se solicitan. En principio, cualquier aproximación que se haga es aceptable. No tendrán más o menos puntos por usar una en particular. Elegir la más sencilla.

AWS ofrece hasta 20GB de almacenamiento para la capa gratuita del servicio de bases de datos relacionales (RDS), el cual se puede ampliar aunque con impacto en los costos.

Las dudas pueden ser enviadas a: hector.simarro@campusviu.es