



**Pontificia Universidad Javeriana**  
**Departamento de Ingeniería de Sistemas**  
**Estructuras de Datos**  
**Proyecto del curso, 2018-10**

## **1 Descripción del problema**

Las empresas de mensajería y distribución de paquetes prestan un servicio básico a la sociedad: proporcionan toda la logística para facilitar la recolección y entrega de diferentes tipos de paquetes a lo largo de toda la geografía nacional e internacional. Aunque el surgimiento del correo electrónico hizo que disminuyera drásticamente la cantidad de cartas y documentos que se envían de un lado al otro, el nacimiento del comercio electrónico y las ventas por internet han permitido que estas empresas sigan manteniendo un alto volumen de entregas.



En general, una empresa de mensajería establecida, por ejemplo, a nivel nacional puede verse como una red interconectada de oficinas de reparto, en donde se procesan los paquetes y se envían a otra oficina o se entregan a su destinatario final. Cada oficina de reparto procesa sus paquetes agrupándolos por regiones o sectores similares de destino, facilitando así el reparto de cada grupo. Es decir, cada región o sector de reparto agrupa un conjunto de paquetes cuyo lugar de destino es igual o similar.

Para poder desarrollar este trabajo de forma eficiente, las empresas requieren contar con un sistema de información que les permita procesar rápidamente cada paquete al momento de recibirlo para así repartirlo y entregarlo en el menor tiempo posible. Para poder desarrollar tal sistema de información, es importante resaltar los datos disponibles dentro del proceso:

- Una oficina de reparto está identificada por un código individual, un nombre y la dirección (incluyendo ciudad, región, país) donde está ubicada. Esta oficina procesa únicamente paquetes que deben repartirse en su área de influencia.
- La región o sector de reparto dentro de la oficina se identifica por un nombre y un código o abreviatura. Cada oficina tiene un conjunto de regiones o sectores propios, que corresponden a su área de influencia y donde se programa el reparto directo.
- Cada paquete debe contener información completa que facilite su seguimiento y entrega de forma eficiente. Para esto, se requiere contar con los datos básicos del remitente y el destinatario. Adicionalmente, cada paquete tiene un peso, un tipo de contenido, y un número de guía (consecutivo) individual que lo identifica dentro de la empresa.
- Para manejar fácilmente los datos del remitente y destinatario de cada paquete, se cuenta con un directorio global de personas, donde para cada una se almacena el nombre completo, un número de identificación personal, la dirección de residencia (incluyendo ciudad, región, país) y un teléfono de contacto en caso de alguna eventualidad.

En la red interconectada de oficinas, que permite la adecuada distribución de paquetes, pueden identificarse dos niveles de conectividad:

- El primer nivel tiene una distribución jerárquica, en la cual existe una oficina general que cubre una ciudad o municipio y de la cual dependen otras oficinas que cubren partes de esa ciudad o municipio. Por ejemplo, podría existir una oficina general para la ciudad de Bogotá, de la cual dependen varias oficinas que cubren las 21 localidades de la ciudad, y de cada una de ellas pueden depender a su vez las oficinas de los barrios de la localidad. Esta jerarquía puede tener tantos niveles como sean necesarios para garantizar un adecuado reparto de los paquetes.
- El segundo nivel de conectividad es la interacción entre oficinas generales, las cuales están conectadas entre sí para facilitar el envío y recepción de paquetes. Por ejemplo, las oficinas generales de Bogotá, Medellín, Cali y otras ciudades principales estarían conectadas todas entre sí para facilitar el movimiento de los paquetes.

## 1.1 Descripción de la información de entrada

(ver documento del enunciado de la primera entrega)

# 2 Descripción del proyecto

El objetivo del presente proyecto es construir un sistema de información para la empresa de mensajería. El sistema se implementará como una aplicación que recibe comandos textuales, agrupados en componentes con funcionalidades específicas. A continuación se describen los componentes individuales que conforman el proyecto.

## 2.1 Componente 1: gestión de la oficina de reparto.

**Objetivo:** Los algoritmos implementados en este componente servirán para gestionar la información básica de una oficina de reparto y de los paquetes procesados. Este componente ya debería estar completamente implementado y funcional con las siguientes operaciones (ver detalles en el documento del enunciado de la primera entrega):

- **comando:** `cargarPersonas nombre_archivo`
- **comando:** `cargarPaquetes nombre_archivo`
- **comando:** `registrarPersona`
- **comando:** `registrarPaquete`
- **comando:** `conteoPaquetes`
- **comando:** `salir`

## 2.2 Componente 2: jerarquía de oficinas.

**Objetivo:** Los algoritmos implementados en este componente servirán para gestionar el reparto de paquetes en el primer nivel de conectividad de la red de distribución. Este componente se implementará con las siguientes funciones, redefinidas a partir del componente anterior:

- **comando:** `cargarOficinas nombre_archivo`  
**posibles salidas en pantalla:**  
(Archivo vacío o incompleto) El archivo `nombre_archivo` no contiene información válida.  
(Archivo no existe) El archivo `nombre_archivo` no existe o es ilegible.  
(Resultado exitoso) Desde el archivo `nombre_archivo`, se han cargado exitosamente  $x$  registros; mientras que  $y$  registros presentaron problemas.  
**descripción:** Carga en memoria la información contenida en el archivo identificado por `nombre_archivo`, la cual debe corresponder a la información básica de un conjunto de oficinas. Se debe informar la cantidad de registros que pudieron procesarse correctamente, así como aquellos que no pudieron ser almacenados por alguna inconsistencia en la información.

- comando:** `cargarRegiones nombre_archivo`  
**posibles salidas en pantalla:**  
 (Archivo vacío o incompleto) El archivo *nombre\_archivo* no contiene informacion valida.  
 (Archivo no existe) El archivo *nombre\_archivo* no existe o es ilegible.  
 (Resultado exitoso) Desde el archivo *nombre\_archivo*, se han cargado exitosamente *x* registros; mientras que *y* registros presentaron problemas.  
**descripción:** Carga en memoria la información contenida en el archivo identificado por *nombre\_archivo*, la cual debe corresponder a la información básica de un conjunto de regiones. Dado que las regiones están relacionadas con las oficinas, el comando debe generar la interconexión entre oficinas que genere la jerarquía del sistema. Se debe informar la cantidad de registros que pudieron procesarse correctamente, así como aquellos que no pudieron ser almacenados por alguna inconsistencia en la información.
- comando:** `cargarPersonas nombre_archivo`  
**posibles salidas en pantalla:**  
 (Archivo vacío o incompleto) El archivo *nombre\_archivo* no contiene informacion valida.  
 (Archivo no existe) El archivo *nombre\_archivo* no existe o es ilegible.  
 (Resultado exitoso) Desde el archivo *nombre\_archivo*, se han cargado exitosamente *x* registros; mientras que *y* registros presentaron problemas.  
**descripción:** Carga en memoria la información contenida en el archivo identificado por *nombre\_archivo*, la cual debe corresponder a la información básica de un grupo de personas. Se debe informar la cantidad de registros que pudieron procesarse correctamente, así como aquellos que no pudieron ser almacenados por alguna inconsistencia en la información.
- comando:** `cargarPaquetes nombre_archivo`  
**posibles salidas en pantalla:**  
 (Archivo vacío o incompleto) El archivo *nombre\_archivo* no contiene informacion valida.  
 (Archivo no existe) El archivo *nombre\_archivo* no existe o es ilegible.  
 (Resultado exitoso) Desde el archivo *nombre\_archivo*, se han cargado exitosamente *x* registros; mientras que *y* registros presentaron problemas.  
**descripción:** Carga en memoria la información contenida en el archivo identificado por *nombre\_archivo*, la cual debe corresponder a la información básica de un grupo de paquetes. Se debe informar la cantidad de registros que pudieron procesarse correctamente, así como aquellos que no pudieron ser almacenados por alguna inconsistencia en la información.
- comando:** `registrarOficina`  
**posibles salidas en pantalla:**  
 (Oficina ya existe) La oficina con identificacion *ofi\_id* ya se encuentra registrada en el sistema.  
 (Resultado exitoso) La oficina con identificacion *ofi\_id* ha sido registrada exitosamente.  
**descripción:** Permite insertar en el sistema la información asociada de una oficina. La información básica de la oficina (código de identificación, nombre, dirección, ciudad) debe solicitarse al usuario por teclado y validarse adecuadamente.
- comando:** `registrarRegion`  
**posibles salidas en pantalla:**  
 (Región ya existe) La region con codigo *reg\_id* ya se encuentra registrada en el sistema.  
 (Resultado exitoso) La region con codigo *reg\_id* ha sido registrada exitosamente.  
**descripción:** Permite insertar en el sistema la información asociada de una región, e interconectarla con la oficina respectiva. La información básica de la región (nombre, código, identificación de la oficina a la que pertenece) debe solicitarse al usuario por teclado y validarse adecuadamente.
- comando:** `registrarPersona`  
**posibles salidas en pantalla:**  
 (Persona ya existe) La persona con identificacion *num\_id* ya se encuentra registrada en el sistema.  
 (Resultado exitoso) La persona con identificacion *num\_id* ha sido registrada exitosamente.

**descripción:** Permite insertar en el sistema la información asociada de una persona, para usarla posteriormente como remitente o destinatario de un paquete a enviar. La información básica de la persona (nombres, identificación, dirección, ciudad, teléfono) debe solicitarse al usuario por teclado y validarse adecuadamente.

- **comando:** registrarPaquete

**posibles salidas en pantalla:**

(Paquete ya existe) El paquete con numero de guia *num\_guia* ya se encuentra registrado en el sistema.

(Resultado exitoso) El paquete con numero de guia *num\_guia* ha sido registrado exitosamente.

**descripción:** Permite insertar en el sistema un paquete a entregar con toda su información asociada. La información básica del paquete (identificación remitente, identificación destinatario, peso, tipo de contenido, número de guía, código oficina, código región) debe solicitarse al usuario por teclado y validarse adecuadamente.

- **comando:** conteoPaquetes

**posibles salidas en pantalla:**

(Paquetes no registrados) No existe informacion de paquetes registrada en el sistema.

(Resultado exitoso) Se encuentran en el sistema *cant\_paquetes* pendientes por entregar. Estan distribuidos asi:

*cant\_paquetes\_r1* en la oficina *cod\_ofi1*, region de reparto *nombre\_r1*

*cant\_paquetes\_r2* en la oficina *cod\_ofi2*, region de reparto *nombre\_r2*

...

**descripción:** De acuerdo a la información almacenada, realiza el conteo de los paquetes registrados que se encuentran pendientes por entregar. Se debe informar el total de paquetes pendientes (valor *cant\_paquetes*) y el conteo parcial por cada una de las oficinas y las regiones de reparto.

- **comando:** repartirPaquetes *codigo\_oficina*

**posibles salidas en pantalla:**

(Regiones de reparto no definidas) La oficina *codigo\_oficina* no tiene regiones de reparto asociadas.

(Resultado exitoso) Se han repartido exitosamente *x* paquetes en *y* oficinas de las regiones de reparto de la oficina *codigo\_oficina*.

**descripción:** Utilizando la información de la jerarquía de oficinas, el comando distribuye los paquetes almacenados en las regiones de reparto de la oficina dada y los ubica en las oficinas respectivas, distribuyéndolos de forma equitativa en las regiones de reparto de las oficinas dependientes.

## 2.3 Componente 3: red de distribución.

**Objetivo:** Los algoritmos implementados en este componente servirán para incorporar el segundo nivel de conectividad de la red y así gestionar completamente el recibo y entrega de paquetes. Las funciones de este componente se informarán posteriormente.

## 2.4 Interacción con el sistema

La interfaz de la aplicación a construir debe ser una consola interactiva donde los comandos correspondientes a los componentes serán tecleados por el usuario, de la misma forma que se trabaja en la terminal o consola del sistema operativo. El indicador de línea de comando debe ser el caracter '\$'. Se debe incluir el comando 'ayuda' para indicar una lista de los comandos disponibles en el momento. Así mismo, para cada comando se debe incluir una ayuda de uso que indique la forma correcta de hacer el llamado, es decir, el comando 'ayuda *comando*' debe existir. Cada comando debe presentar en pantalla los mensajes de resultado especificados antes, además de otros mensajes que se consideren necesarios. Los comandos de los diferentes componentes deben ensamblarse en un único sistema (es decir, funcionan todos dentro de un mismo programa, no como programas independientes por componentes).

### 3 Evaluación

Las entregas se harán en la correspondiente actividad de UVirtual, hasta la media noche del día anterior al indicado para la entrega. Se debe entregar un archivo comprimido (únicos formatos aceptados: .zip, .tar, .tar.gz, .tar.bz2, .tgz) que contenga dentro de un mismo directorio (**sin estructura de carpetas interna**) los documentos (único formato aceptado: .pdf) y el código fuente (.h, .hxx, .hpp, .c, .cxx, .cpp). Si la entrega contiene archivos en cualquier otro formato, será descartada y no será evaluada, es decir, la nota definitiva de la entrega será de **0 (cero) sobre 5 (cinco)**.

#### 3.1 Entrega 1 (jueves 1 y viernes 2 de marzo de 2018)

Componente 1 completo y funcional. Esta entrega tendrá una sustentación durante la correspondiente sesión de clase. Esta entrega se compone de:

- (30%) Documento de diseño. El documento de diseño debe seguir las pautas de ingeniería que usted ya conoce: descripción de entradas, salidas y condiciones para el procedimiento principal y las operaciones auxiliares. Para la descripción de los TADs utilizados, debe seguirse la plantilla definida en clase. Además, se exigirán esquemáticos (diagramas, gráficos, dibujos) que describan el funcionamiento general de las operaciones principales.
- (55%) Código fuente compilable en el compilador gnu-g++ (versión 4.0.0, como mínimo). Este porcentaje de la entrega será un promedio de la evaluación de cada comando.
- (15%) Sustentación con participación de todos los miembros del grupo.

#### 3.2 Entrega 2 (jueves 19 y viernes 20 de abril de 2018)

Componentes 1 y 2 completos y funcionales. Esta entrega tendrá una sustentación durante la correspondiente sesión de clase. Esta entrega se compone de:

- (15%) Completar la funcionalidad que aún no haya sido desarrollada de la primera entrega. Se debe generar un acta de evaluación de la entrega anterior que detalle los elementos faltantes y cómo se completaron para la segunda entrega.
- (25%) Documento de diseño. El documento de diseño debe seguir las pautas de ingeniería que usted ya conoce: descripción de entradas, salidas y condiciones para el procedimiento principal y las operaciones auxiliares. Para la descripción de los TADs utilizados, debe seguirse la plantilla definida en clase. Además, se exigirán esquemáticos (diagramas, gráficos, dibujos) que describan el funcionamiento general de las operaciones principales.
- (45%) Código fuente compilable en el compilador gnu-g++ (versión 4.0.0, como mínimo). Este porcentaje de la entrega será un promedio de la evaluación de cada comando.
- (15%) Sustentación con participación de todos los miembros del grupo.

#### 3.3 Entrega 3 (jueves 31 de mayo y viernes 1 de junio de 2018)

Componentes 1, 2 y 3 completos y funcionales. Esta entrega tendrá una sustentación (del proyecto completo) durante el jueves 31 de mayo o el viernes 1 de junio de 2018 entre las 8am y las 6pm. Esta entrega se compone de:

- (15%) Completar la funcionalidad que aún no haya sido desarrollada de la primera y segunda entregas. Se debe generar un acta de evaluación de la entrega anterior que detalle los elementos faltantes y cómo se completaron para la tercera entrega.
- (25%) Documento de diseño. El documento de diseño debe seguir las pautas de ingeniería que usted ya conoce: descripción de entradas, salidas y condiciones para el procedimiento principal y las operaciones auxiliares. Para la descripción de los TADs utilizados, debe seguirse la plantilla definida en clase. Además, se exigirán esquemáticos (diagramas, gráficos, dibujos) que describan el funcionamiento general de las operaciones principales.

- (45%) Código fuente compilable en el compilador `gnu-g++` (versión 4.0.0, como mínimo). Este porcentaje de la entrega será un promedio de la evaluación de cada comando.
- (15%) Sustentación con participación de todos los miembros del grupo.