



## Proyecto 2

### Objetivos Generales

- Que el estudiante aplique los conceptos vistos en clase respecto a estructuras dinámicas de datos lineales y no lineales.
- Que el estudiante aplique los conceptos y estructuras vistas en la resolución de problemas reales.

### Objetivos Específicos

- Que el estudiante genere estructuras genéricas y adapte la implementación de dichas estructuras a las necesidades del problema.
- Que el estudiante genere un nivel de abstracción alto.

### Descripción

En el mundo actual las compras personalizadas en lugares no físicos están tomando un mayor auge del esperado, incluso en nuestro país. De allí la importancia de aprender tecnologías Web. Pero, una de las grandes limitaciones es la interoperación entre diferentes tipos de desarrollos. Allí recae la importancia de los Web Services, que permiten la comunicación entre diferentes lenguajes de programación Web.

El proyecto consiste en el desarrollo de una tienda virtual de cualquier tipo de producto, desarrollado en java que se comunique vía Web Services codificando y decodificando la información que se use en la aplicación para su transporte en la red.

Esta aplicación debe ser capaz de representar las estructuras de memoria dinámica en forma visual (Graphviz), además de llevar una consola de acciones del servidor (log).

## Implementación

### Descripción de las estructuras

#### Usuarios

Esta estructura está del lado del servidor. Los usuarios estarán cargados en un árbol **AVL ordenado por Nickname**. Entre los datos con los que cuenta son:

- Nickname
- Contraseña
- Lista de direcciones, que es una lista enlazada con todas las direcciones a las que se les puede enviar sus compras y/o su factura. Los datos que lleva el nodo son:

- Dirección
- Envío ('0' si no se envía allí las compras, '1' si se envían).
- Facturación ('0' si no se envía allí la factura, '1' si se envían).
- Lista de productos por comprar (ver más adelante).
- Carrito de compra (ver más adelante).

Se debe implementar, también, una ventana para dar mantenimiento a los usuarios, es decir, crear nuevos usuarios, modificar usuarios (en cualquiera de sus datos) y eliminar. Esto es una opción que realiza un usuario al ingresar.

### Lista de Productos por comprar

Esta estructura es una cola, apuntada por la estructura de usuarios. Los datos que guarda son:

- Cantidad
- Apuntador al producto. Este producto está en la estructura de productos.

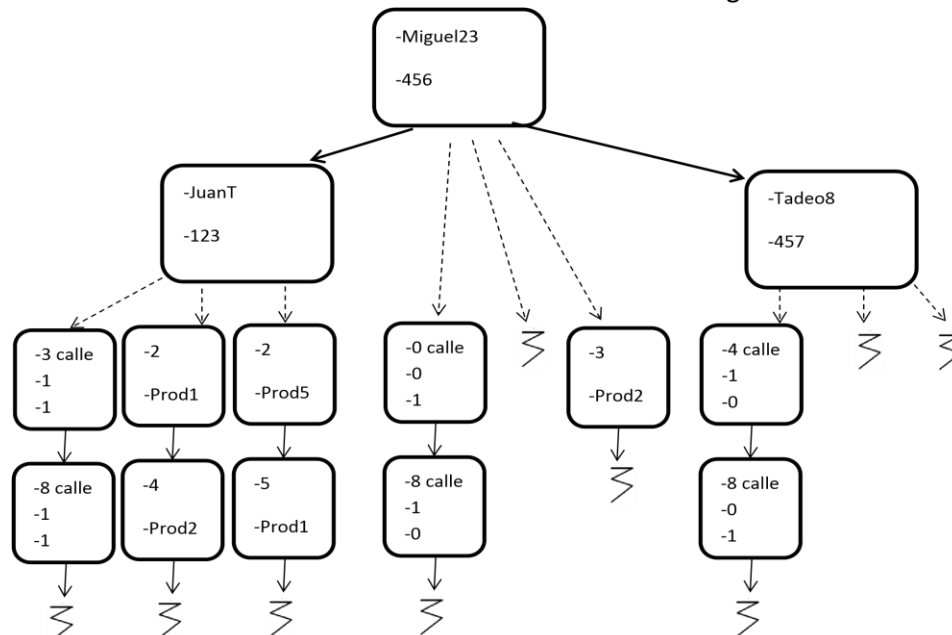
### Carrito de compras

Esta estructura es una cola, apuntada por la estructura de usuarios. Los datos que guarda son:

- Cantidad
- Apuntador al producto. Este producto está en la estructura de productos.


### Estructura

En total, la estructura anteriormente descrita se entenderá como la siguiente:



**De esta estructura se debe implementar inserción, eliminación, actualización.**

## Productos

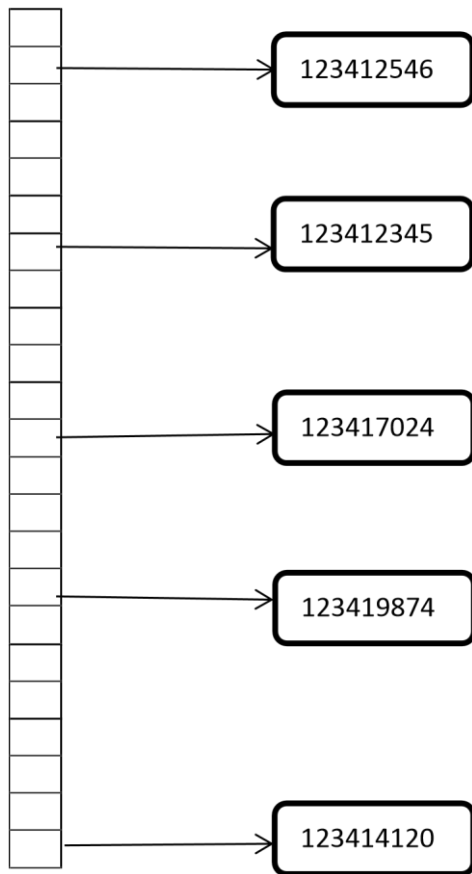
Los productos se guardarán en una tabla de dispersión (hash). La función hash es de **plegamiento** y la resolución de colisiones por exploración cuadrática. El tamaño inicial del vector de la tabla de dispersión es de 23. A estos nodos es que debe apuntar las dos listas de productos mencionadas anteriormente. 

Los datos que guardará son:

- Código de producto
- Nombre
- Marca
- Precio
- Imagen ruta)

**La tabla hash tendrá un tamaño inicial de 10, al llegar al 80% de utilización se deberá de hacer un rehashing en donde su nuevo tamaño seria,  $2*N$ , donde N es el tamaño actual. De esta estructura se debe implementar inserción, eliminación y actualización de datos.**

## Estructura



## Ventas

Las ventas estarán guardadas en un **árbol B de orden 5**, ordenado por número de factura. Los datos que guardará serán:

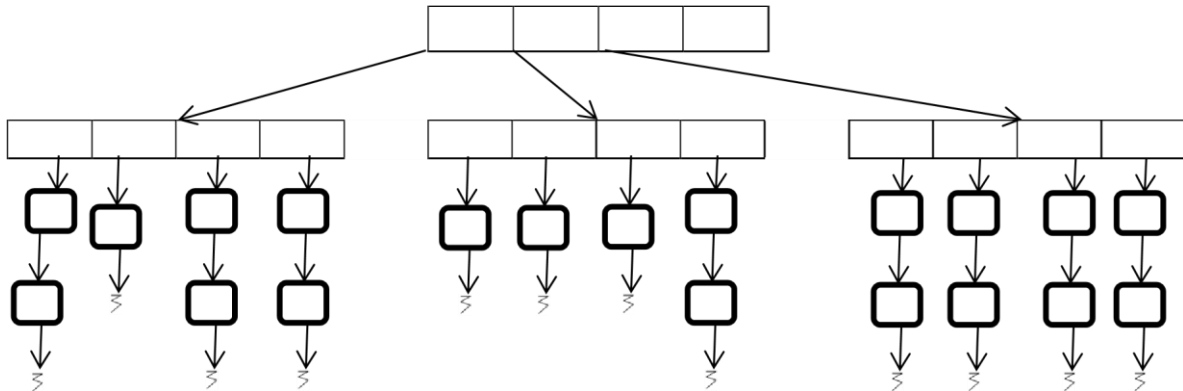
- Número Factura
- Fecha
- Total
- Apuntador al Usuario
- Apuntador al Detalle

## Detalle

Este detalle se guardará lo que conceptualmente se conoce como maestro/detalle en una base de datos. Será una cola y guardará los siguientes datos:

- Cantidad
- Precio (Se guarda este precio pues puede que el precio cambie a lo largo del tiempo, por lo tanto, es necesario que este precio se guarde).
- Apuntador Producto

## Estructura



## Funcionalidad

Se asumirá que la aplicación es cerrada y debe haber una previa suscripción (registro) a la página para poder ver el catálogo. Esto significa que la aplicación cliente debe tener una interfaz de entrada que valide los datos previamente cargados del lado del servidor (login).

1. Ingreso de un usuario. Se debe informar si existe un error o, en caso de estar correcto, ingresar al catálogo de ventas.
2. En este catálogo se deben mostrar todos los productos, junto con su imagen y precio. Debe existir paginación, es decir, que solo se pueden mostrar 20 productos por página y, si se desean ver otros 20, se debe ir a una página “siguiente”.
3. Se puede seleccionar uno o varios productos y se tienen dos opciones. Se puede agregar a la lista de **Productos por comprar** o al **Carrito de compra**.

4. Si fueron agregados a **Productos por comprar**, se pueden eliminar de esa lista o agregar al **Carrito de compra**. También, se puede ver la estructura dibujada por medio de Graphviz.
5. Si fueron agregados al **Carrito de compra** se pueden eliminar de la lista o comprar. Se deben comprar todos los productos a la vez. También, se puede ver la estructura dibujada por medio de Graphviz.
6. Al realizar la compra, se debe preguntar nuevamente la contraseña para validar si realmente es el correcto el que está comprando.
7. Al comprar, se asumirá que solamente con ingresar un valor correcto de pago se puede realizar la compra. No se habrán diferentes formas de pago, solamente se “pagará” ingresando el valor correcto. Además, se podrá escoger entre las direcciones apropiadas para enviar los productos y enviar la factura. Esto actualizará el árbol B de ventas y el detalle de ventas del lado del servidor. Se debe mostrar un reporte de la venta al usuario con la venta recién realizada.
8. Luego de realizar la compra y ver el reporte de la venta, se debe poder regresar al menú o ver las matrices utilizadas para la encriptación (se explicará más adelante).
9. Se puede cerrar sesión, por lo que solo lo que se ha comprado se guarda en el árbol B. Lo que quedó en el carrito o en los productos por comprar, se puede recuperar en el siguiente ingreso.
10. Cliente puede ver todas sus compras realizadas, puede descarga una factura en específico en formato pdf con todos los datos de la factura.

Del lado del servidor

1. Cargar usuarios
2. Cargar productos
3. Cargar ventas con su detalle
4. Dar los servicios Web necesarios (Ventas, validación de usuarios, proveer productos).
5. Guardar todas las estructuras en un archivo (Estructura del archivo se explicará más adelante).

## Reportes

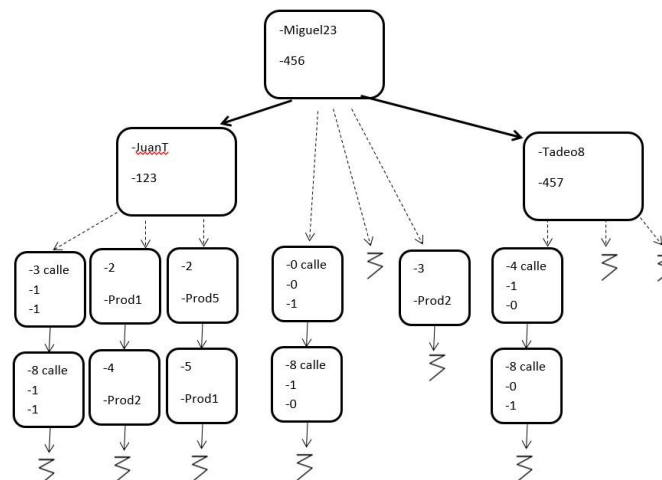
Del lado del cliente, se pueden realizar los siguientes reportes:

- Ver lista de productos por comprar
  - o Datos mínimos para mostrar:
    - Nombre del Producto
    - Código de Producto
    - Precio
    - Cantidad
- Ver carrito de compra

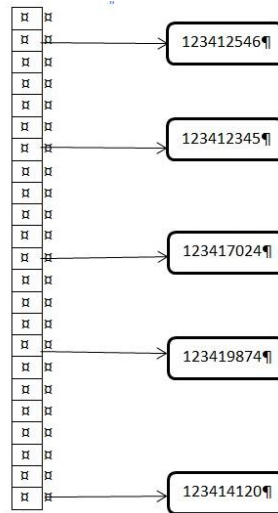
- o Datos mínimos para mostrar:
  - Nombre del Producto
  - Código de Producto
  - Precio
  - Cantidad

De lado del servidor.

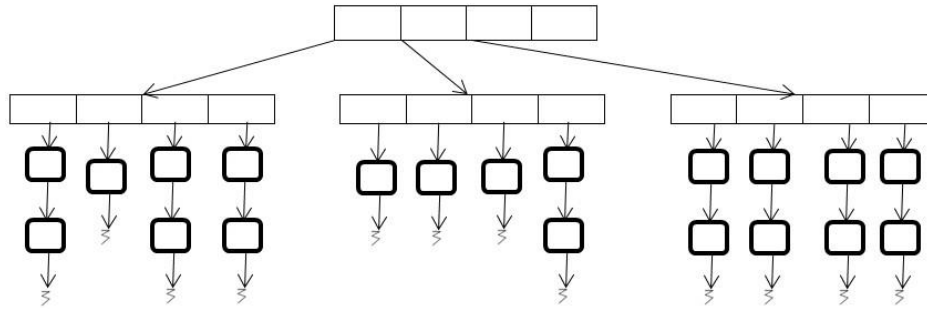
- Ver la estructura de usuarios.
  - o Datos mínimos para mostrar:
    - Nickname
    - Contraseña
    - Lista de Direcciones
      - Dirección
      - Envío
      - Facturación
    - Carrito de Compra
      - Nombre del Producto
      - Código de Producto
      - Precio
      - Cantidad
    - Lista de productos por Comprar
      - Nombre del Producto
      - Código de Producto
      - Precio



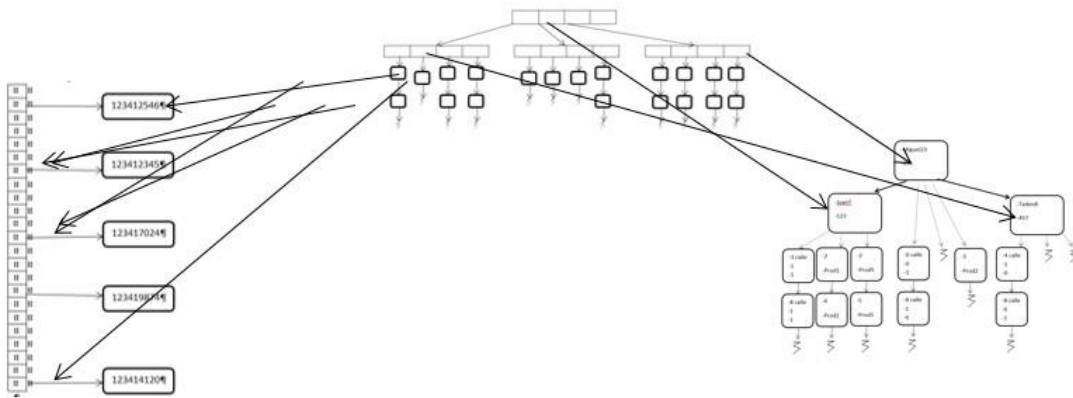
- Ver estructura de productos.
  - Datos mínimos para mostrar:
    - Código de Producto
    - Nombre del Producto
    - Precio
    - Marca



- Ver Estructura de Ventas
  - Datos mínimos para mostrar:
    - Numero de Factura
    - Fecha
    - Usuario que realizo la compra
    - Lista de Productos
      - Nombre del Producto
      - Código del Producto
      - Precio
      - Cantidad



- Realizar una búsqueda de ventas por rangos de fecha y/o valor y/o cliente.
- Mostrar los productos más vendidos por rango de fecha.
- Mostrar los usuarios que más compras realizan por rangos de fecha.
- Mostrar la estructura global y sus relaciones (mostrando las EDD con sus datos mínimos).



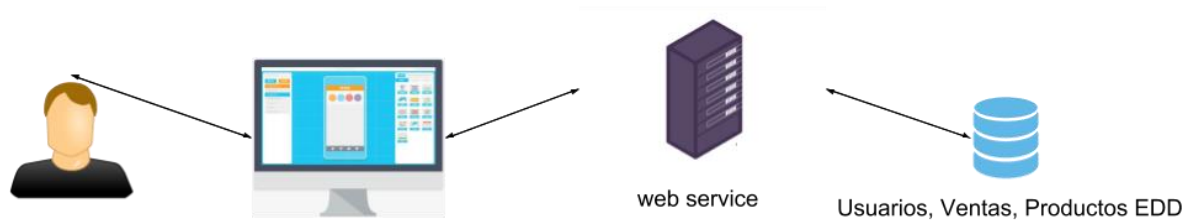
## Consola de reportes

Se generará un archivo de log en el cual se indicará los siguientes mensajes con su tipo específico:

- INFO ○ Cada inserción satisfactoria realizada a cualquier estructura.
- WARN
  - Si ocurre una colisión, explicando qué datos colisionaron y en qué estructura.
- ERROR
  - Si un contacto ya existe y no se puede agregar.
  - Si un usuario ya existe y no se puede agregar.
  - Si el valor de la venta no era el correcto. ○ Si el producto ya existe y no se puede agregar.



## Diagrama de la solución



## Documentación

Se debe presentar documentación técnica que sea bien realizada. Esta información debe incluir:

- Diagrama de clases
- Diagrama de paquetes
- Diagrama de colaboración
- Análisis de variables (su uso y su envío).

## Estructura de archivos de carga

- **Usuario (Nickname, Contraseña):**
  - Nickname: alfanumérico
  - Contraseña: alfanumérico
- **Dirección(Nickname, Direccion, Envío, Facturacion):**
  - Nickname: alfanumérico
  - Dirección: alfanumérico
  - Envío: numérico (0 o 1)
  - Facturación: numérico (0 o 1)
- **PorComprar(Nickname, Cantidad, Producto):**
  - Nickname: alfanumérico
  - Cantidad: numérico
  - Producto: numérico
- **Carrito (Nickname, Cantidad, Producto):**
  - Nickname: alfanumérico
  - Cantidad: numérico
  - Producto: numérico
- **Producto (Código, Nombre, Marca, Precio, Imagen):**
  - Código: numérico
  - Nombre: alfanumérico
  - Marca: alfanumérico
  - Precio: numérico
  - Imagen: alfanumérico (Este describe el nombre de la imagen)

- **Venta (Número\_Factura, Fecha, Total, Usuario):**
  - Número\_Factura: numérico
  - Fecha: alfanumérico. Formato dd/mm/aaaa
  - Total: numérico
  - Usuario: alfanumérico (Nickname)
- **Detalle (Número\_Factura, Cantidad, Precio, Producto):**
  - Número\_Factura: numérico
  - Cantidad: numérico
  - Precio: numérico
  - Producto: numérico (Código de producto)

Todos los archivos son csv y van en el orden especificado.

## Entregables

- Código Fuente
- Ejecutables
- Manual Técnico
- Manual de Usuario
- Documento de Especificación de Tareas por persona, a entregar **Martes 25 de Diciembre antes de medianoche.**

## Observaciones

- Lenguaje de programación a utilizar: Java
- Herramientas de comunicación: Web Services Java
- Sistema Operativo: Libre
- IDE: Netbeans
- Al generar reportes debe abrirse solo, no ir a buscarlo entre carpetas
- Se prohíbe el uso de vectores o matrices para el almacenamiento de la información
- Null point exception tendrán automáticamente una nota de 0 en la estructura y funcionalidad que lo provoque.
- **Reportes de Estructuras que no muestren los datos mínimos no se calificaran.**
- **Copias tendrán nota de 0 puntos y serán reportados al catedrático y a la Escuela de Sistemas.**
- **Para este proyecto, se puede trabajar en parejas y solo las parejas que envíen el documento requerido van a poder realizar el proyecto, el documento se entrega martes 25 de diciembre del 2018 antes de medianoche.**
- **Durante la calificación, para quienes realizaron el proyecto en parejas, se les harán preguntas a ambos integrantes sobre la realización del proyecto, de no poder responder alguna de esas preguntas, el proyecto se les será anulado.**

## Requerimientos mínimos para tener derecho a calificación

- Todas las estructuras deben de ser realizada por el estudiante, sin el uso de librerías específicas de ninguno IDE o Framework.

- La aplicación debe permitir generar reportes en graphviz en cualquier momento.
- La aplicación debe permitir jugar con comunicación en tiempo real entre el cliente y el servidor.
- Estructuras solas con sus funciones no se calificarán, por lo menos se deberá de tener alguna funcionalidad del proyecto.
- Tiene que cargar los archivos de entrada.

## Fecha y modo de entrega

- Domingo 6 de enero del 2019 antes de las 23:59 vía Classroom.
- De entregarse con retraso no se calificará.
- Formato: [EDD]Proyecto2\_<carne>.rar o [EDD]Proyecto2\_<carne1>\_<carne2>.rar para Parejas.