**PRÁCTICA 16 | LISTAS SIMPLES LIGADAS**

# Introducción

Para esta práctica tengo pensado crear varias clases de entidad que almacenarán los datos requeridos. Será de gran ayuda poder usar memoria dinámica para que no haya limitante a la hora de agregar aplicaciones a las computadoras. En cambio, para éstas mismas pienso que sería bueno usar un arreglo, ya que se da un número fijo de computadoras a poder ser agregadas.

# Desarrollo

La codificación de esta práctica en lugar de ser muy complicada fue algo extensa.

Comencé creando las clases de entidad que necesitaba: **Computadora** y **Aplicación** que fueron incluida en el paquete de **Catalogos**. Teniendo las anteriores clases comentadas y codificadas proseguí a crear la clase con la lógica de operaciones: **CentroComputo** colocada también en el mismo paquete que las dos anteriores.

Debido a que la práctica tenía muchos puntos, me límite a hacer pruebas del código hasta el final. Afortunadamente la mayoría de los métodos funcionaron como lo había planeado, cosa que superó mis expectativas (pensaba que necesitaría corregir más cosas).

Tuve algunos dilemas sobre cómo imprimir los datos de una computadora, ya que, en algunos puntos entendía que tenía que ser más detallada la descripción y en otros no tanto. Como producto, agregué un nuevo enum de nombre **TipoImpresion**, para acaparar una impresión detallada o una menos extensa.

Las listas ligadas simples fueron de gran ayuda para la realización de la práctica, las utilicé más de lo que tenía pensado, y a diferencia de un arreglo, como es evidente, no tenía el problema de definir un tamaño fijo en su inicialización.

A continuación, se indican los archivos creados junto con sus métodos:

## Computadora.java (paquete catalogos).

* public Computadora(int numero, int ram, String discoDuro, String procesador, String marca).
* public int getNumero().
* public void setNumero(int numero).
* public int getRam().
* public void setRam(int ram).
* public String getDiscoDuro().
* public void setDiscoDuro(String discoDuro).
* public String getProcesador().
* public void setProcesador(String procesador).
* public String getMarca().
* public void setMarca(String marca).
* public ListaLigada getAplicaciones().
* public boolean agregarAplicacion(Aplicacion nuevaApp).
* public Aplicacion eliminarAplicacion(String nombre).
* public String getDatos().
* public String toString().

## Aplicacion.java (paquete catalogos).

* public Aplicacion(String nombre, String logo, String version, int ramMinima).
* public String getNombre().
* public void setNombre(String nombre).
* public String getLogo().
* public void setLogo(String rutaLogo).
* public ListaLigada getAutores().
* public boolean agregarAutor(String autor).
* public Object eliminarAutor(String autor).
* public String getVersion().
* public void setVersion(String version).
* public int getRamMinima().
* public void setRamMinima(int ramMinima).
* public String getDatos().
* public String toString().

## TipoImpresion.java (paquete utilidades).

* DETALLADA(1, "Detallada").
* NO\_DETALLADA(2, "No Detallada").
* private TipoImpresion(int id, String tipo).
* public int getId().
* public String getTipo().

## CentroComputo.java (paquete catalogos).

* public CentroComputo(int tamano).
* public Computadora getComputadora(int numeroComputadora).
* public boolean agregarComputadora(Computadora nuevaComputadora).
* public Aplicacion buscarAplicacion(Computadora computadora, Aplicacion aplicacion).
* public Computadora eliminarComputadora(int numeroComputadora).
* public boolean agregarAplicacion(int numeroComputadora, Aplicacion nuevaApp).
* public Aplicacion eliminarAplicacion(int numeroComputadora, String nombreApp).
* public Arreglo getComputadorasPorApp(Aplicacion aplicacion).
* public ListaLigada getComputadorasCompatibles(Aplicacion aplicacion).
* public ListaLigada getComputadorasAppsRam(int ram).
* public ListaLigada getComputadorasConCantRam(int ram).
* public ListaLigada getComputadorasConApp(Aplicacion aplicacion).
* public void imprimirDatosTodasComputadoras().
* public void imprimirDatosComputadora(int numeroComputadora, TipoImpresion tipoImpresion).
* public void imprimirAplicacionesComputadora(Computadora computadora).
* public void imprimirDatosAplicacion(Aplicacion aplicacion).
* public void imprimirComputadorasCompatibles(Aplicacion aplicacion).
* public void imprimirComputadorasAppsRam(int ram).
* public void imprimirComputadorasConCantRam(int ram).
* public void imprimirComputadorasConApp(Aplicacion aplicacion).

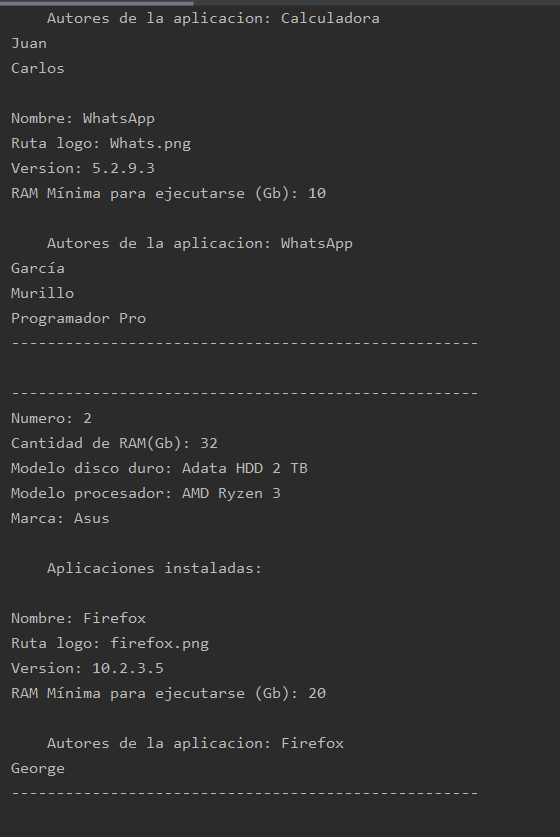
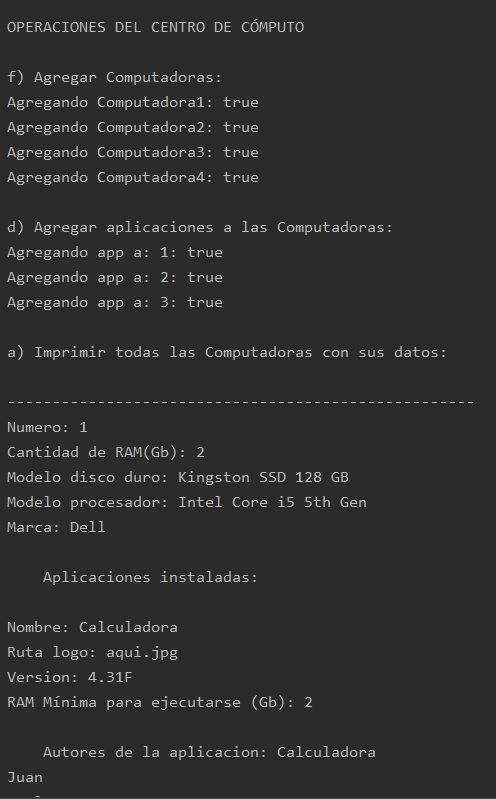
## PruebaCentroComputo.java (paquete interfaces).

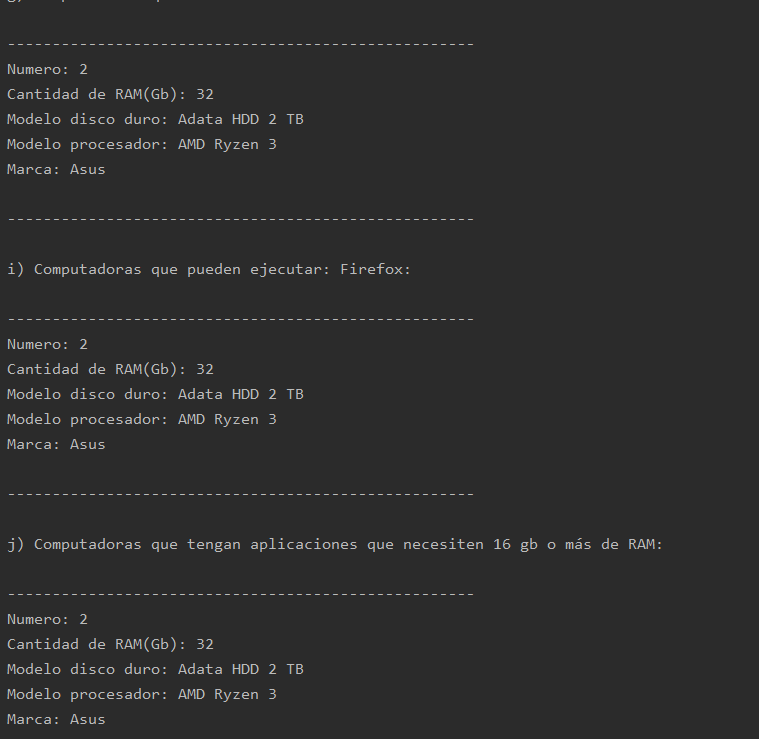
La clase anterior es donde se encuentran las pruebas de la práctica.

**Para mayor información sobre los métodos y la clases creadas vaya a las siguientes rutas dentro del proyecto:**

* **./docs/catalogos/Computadora.html**
* **./docs/catalogos/Aplicacion.html**
* **./docs/catalogos/CentroComputo.html**
* **.**/**docs/utilidades/TipoImpresion.html**

# Capturas del programa funcionando





# Conclusiones

Las listas ligadas tienen una gran utilidad en este tipo de programas de almacenado indefinido de datos. Ya había hecho uso de la clase “ArrayList” de java anteriormente en otras materias, y según mi opinión, funciona bajo este concepto. Fue interesante para esta ocasión volver a hacer uso de este concepto de clase, pero ahora analizado, reflexionado y programado con ayuda del profesor en clase.