

Requerimientos funcionales:

Nombre:	R. #1 Registrar un cliente
Resumen:	El programa recibe los datos del cliente y lo registra en el árbol binario
Entradas:	
Resultados:	Cliente registrado con éxito

Nombre:	R. #2. Eliminar cliente
Resumen:	El programa recibe el código del cliente que se desea eliminar y procede a eliminarlo del árbol binario
Entradas:	
Resultados:	Cliente eliminado con éxito

Nombre:	R. #3. Buscar cliente
Resumen:	El programa recibe el código del cliente que se desea buscar y procede a hacer la búsqueda. Si lo encuentra muestra la información de este cliente.
Entradas:	
Resultados:	Cliente buscado con éxito, información mostrada en pantalla
	Cliente no encontrado

Nombre:	R. #4. Cargar información de los productos de un archivo plano
Resumen:	El programa carga la información de los productos desde un archivo plano
Entradas:	
Resultados:	Información cargada con éxito

Nombre:	R. #5. Exportar información de los productos en un archivo plano
Resumen:	El programa exporta la información de los productos en un archivo plano
Entradas:	
Resultados:	La esfera rebota con éxito

Nombre:	R. #6. Insertar producto nuevo
Resumen:	El programa recibe la información del producto nuevo y lo agrega a una lista doblemente enlazada
Entradas:	
Resultados:	Producto insertado con éxito

Nombre:	R. #7. Eliminar producto
Resumen:	El programa recibe la información del producto que se desea eliminar y procede a quitarlo de la lista doblemente enlazada
Entradas:	
Resultados:	Producto eliminado con éxito

Nombre:	R. #8. Buscar producto
Resumen:	El programa recibe el código del producto que se desea buscar y procede a realizar la búsqueda en la lista doblemente enlazada
Entradas:	
Resultados:	Producto encontrado con éxito
	El producto no existe

Nombre:	R. #9. Modificar nombre producto
Resumen:	El programa recibe el código del producto cuyo nombre se desea modificar y procede a cambiarlo con el nombre nuevo que ingresa el usuario
Entradas:	
Resultados:	Nombre cambiado con éxito
	Producto no encontrado

Nombre:	R. #10. Modificar código producto
Resumen:	El programa recibe el código del producto cuyo código se desea modificar y procede a cambiarlo con el código nuevo que ingresa el usuario
Entradas:	
Resultados:	Código cambiado con éxito

	Producto no encontrado
--	------------------------

Nombre:	R. #11. Modificar precio del producto
Resumen:	El programa recibe el código del producto cuyo precio se desea modificar y procede a cambiarlo con el precio nuevo que ingresa el usuario
Entradas:	
Resultados:	Precio cambiado con éxito
	Producto no encontrado

Nombre:	R. #12. Modificar ubicación del producto
Resumen:	El programa recibe el código del producto cuya ubicación se desea modificar y procede a cambiarlo con la nueva ubicación que ingresa el usuario
Entradas:	
Resultados:	Ubicación cambiada con éxito

	Producto no encontrado
--	------------------------

Nombre:	R. #13. Serializar información
Resumen:	El programa persiste la información por medio de serialización en un archivo txt
Entradas:	
Resultados:	Serialización realizada con éxito

Nombre:	R. #13. Deserialización información
Resumen:	El programa carga la información por medio de deserialización de un archivo txt
Entradas:	
Resultados:	Deserialización realizada con éxito

Nombre:	R. #14. Contratar empleado
Resumen:	El programa recibe la información de un empleado que se desea agregar a un cargo en específico
Entradas:	Información del empleado a agregar
Resultados:	Empleado contratado con éxito

Nombre:	R. #15. Buscar empleado
Resumen:	El programa recibe el código del empleado que desea ser buscado
Entradas:	Código del empleado a buscar
Resultados:	Empleado encontrado con éxito

--	--

Nombre:	R. #16. Despedir empleado
Resumen:	El programa recibe el código del empleado que desea ser despedido
Entradas:	Código del empleado a despedir
Resultados:	Empleado despedido con éxito

Nombre:	R. #17. Agregar cargo
Resumen:	El programa recibe la información del cargo que se desea agregar
Entradas:	Nombre del cargo a agregar con sus respectivas características.
Resultados:	Cargo agregado con éxito

Nombre:	R. #18. Eliminar cargo
Resumen:	El programa recibe la información del cargo que se desea eliminar
Entradas:	Nombre del cargo a eliminar
Resultados:	Cargo eliminado con éxito

Requerimientos No Funcionales

RNF1: En el proceso de búsqueda de los elementos, las operaciones a realizar deben ser realizadas por los métodos implementados por la aplicación.

RNF2: La aplicación debe seguir los principios de usabilidad y diseño, por lo tanto, debe ser intuitivo y agradable con el usuario.

RNF3: La aplicación debe ser inicializada inmediatamente apenas se ejecute.

RNF4: Todas las excepciones, ya sean de tipo de Runtime o errores propios del programa serán manejados por la propia aplicación, permitiendo de esta manera que el proceso de ejecución no termine por estos errores.

RNF5: La aplicación debe ser persistente con la información que el usuario ingrese; esto quiere decir, que una vez ingresados ciertos datos, no importa si el programa es cerrado, los datos van a seguir en la aplicación.