

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES</b>	
<b>CARRERA:</b> Computación		<b>ASIGNATURA:</b> Programación Aplicada	
<b>NRO. PRÁCTICA:</b>	1	<b>TÍTULO PRÁCTICA:</b> Clases Genéricas en Java	
<b>OBJETIVO ALCANZADO:</b> Identificar los cambios importantes de Java Diseñar e Implementar las nuevas técnicas de programación Entender la cada uno de las características nuevas en Java			
<b>ACTIVIDADES DESARROLLADAS</b>			
<a href="https://github.com/ddutan2000/Practica_1-Clases-Genericas.git">https://github.com/ddutan2000/Practica_1-Clases-Genericas.git</a>			
<b>1. Paquetes</b> Se nos ha pedido crear un registro de teléfonos en el cual se crearon 3 paquetes para poder utilizar la arquitectura MVC (Modelo, Vista, Controlador) <ul style="list-style-type: none"> <li>ec.edu.ups.controlador</li> <li>ec.edu.ups.vista</li> <li>ec.edu.ups.modelo</li> </ul>			
<b>2. ec.edu.ups.modelo</b> <ul style="list-style-type: none"> <li>en este paquete se crearon dos clases:</li> <li>Teléfono</li> <li>Usuario</li> </ul>			
<b>3. Clase Teléfono</b> <ul style="list-style-type: none"> <li>En esta clase se instanciaron todos los atributos de un teléfono. Se crearon sus métodos: getters, setters, hashCode, equals, y toString.</li> </ul> <p>Atributos</p> <ul style="list-style-type: none"> <li>int codigo;</li> <li>String numero;</li> <li>String tipo;</li> <li>String operadora;</li> </ul> <p>de igual manera se crearon dos constructores, un constructor de manera normal sin ninguna variable y otro constructor que tiene a todos los atributos del teléfono en sus parámetros.</p>			
<b>4. Clase Usuario</b> <ul style="list-style-type: none"> <li>En esta clase se instanciaron todos los atributos de un Usuario. Se crearon sus métodos: getters, setters, hashCode, equals, y toString.</li> </ul>			

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

#### Atributos

- String cedula;
- String nombre;
- String apellido;
- String correo;
- String contrasenia;
- List<Telefono> listaDeTelefonos;

de igual manera se crearon dos constructores, un constructor de manera normal en la cual solamente se instancia una lista y otro constructor que tiene a todos los atributos del usuario en sus parámetros.

#### Métodos:

- crearTelefono: este método recibe en sus parámetros un objeto de tipo Teléfono. Lo que hace este método es añadir ese teléfono a la lista.
- modificarTelefono: este método recibe en sus parámetros un objeto de tipo Teléfono. Lo que hace este método es que busca el índice del teléfono con el mismo código en la lista y lo que hace es remplazarlo con el nuevo teléfono que se encuentra en el parámetro del método.
- buscarTelefono: este método recibe un numero entero en su parámetro. Este método busca el teléfono con el mismo código en la lista y una vez que lo haya encontrado lo retorna.
- eliminarTelefono: este método recibe en sus parámetros un objeto de tipo Teléfono. Este método busca el índice del teléfono con el mismo código en la lista y lo que hace es eliminarlo de la lista
- ListarTelefono: este método solamente retorna toda la lista de teléfonos que puede tener el usuario.

### 5. Ec.edu.ups.controladores

En este paquete se crearon los controladores que van a interactuar con la vista.

- ConotroladorGeneral
- ControladorUsuario1
- ControladorGenerico

### 6.ControladorGenerico


En esta clase se instanciaro una lista y un objeto de tipo (T)

- List<T> lista;
- T t;

En el controlador solo se instancio la lista.

#### Métodos:

- Crear: este método recibe un objeto de tipo T en sus parámetros. Esta clase añade el objeto a la lista.
- Read: este método recibe un objeto de tipo T en sus parámetros. Mediante los métodos stream() filtramos la lista y encontramos el objeto. Una vez encontrada retornamos el objeto.
- Update: este método recibe un objeto de tipo T en sus parámetros. Recorremos la lista que existe y comparamos con el objeto en los parámetros, si es que encontramos el objeto los remplazamos en la lista
- Delete: este método recibe un objeto de tipo T en sus parámetros. Este método utiliza un iterator para recorrer la lista. Si es que encuentra el objeto que busca lo elimina de la lista.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

- findAll: este método retorna toda la lista
- encontrarObjeto: este método lo único que hace es retornar un objeto de tipo T;

### 7. ControladorGeneral

este controlador utiliza métodos de la clase abstracta (ControladorGenerico).

Atributos:

- private Telefono telefono;
- private List<Telefono> listaTelefonos;
- private int codigo;

este controlador instancia el código y también instancia la lista de teléfono utilizando la lista de la clase ControladorGenerico.

Métodos:

- buscarTelefono: este método recibe una variable de tipo entero en sus parámetros. Este método recorre toda la lista de teléfonos y busca al teléfono con el mismo código, una vez que lo haya encontrado lo retorna.
- update: recibe un objeto de tipo teléfono en sus parámetros. Este método recorre toda la lista de teléfonos y una vez que haya encontrado el teléfono con el mismo código lo reemplaza con el nuevo teléfono que tiene es sus parámetros.
- obtenerCodigo: este método solamente consigue el siguiente código para los números de teléfonos.

### 8. ControladorUsuario1

Este controlador utiliza métodos de la clase abstracta (controladorGenerico)


Atributos:

- private Telefono telefono;
- private Usuario usuario;
- private ControladorGeneral controladorT;
- private List<Usuario> listaDeUsuarios;

este controlador instancia el controlador del teléfono y también instancia la lista de teléfono utilizando la lista de la clase ControladorGenerico.

Métodos:

- update: recibe un objeto de tipo usuario en sus parámetros. Este método recorre toda la lista de usuarios y una vez que haya encontrado el usuario con la misma cedula lo reemplaza con el nuevo usuario que tiene es sus parámetros.
- verUsuario: este método retorna un usuario.
- autenticar: este método recorre toda la lista de usuarios con un forEach y encuentra a ese usuario con esas credenciales y lo retorna.
- crearTelefono: Este método recibe un teléfono en sus parámetros. Primero crea un nuevo teléfono en el usuario y luego lo añade a ese usuario a la lista de teléfonos.
- modificartelefono: Este método recibe un teléfono en sus parámetros. Este método actualiza el teléfono en la clase usuario y luego llama al método de la clase padre y lo actualiza por este medio.
- eliminarTelefono: Este método recibe una variable de tipo entero en sus parámetros. Este método llama al método de búsqueda del controlador teléfono y los asigna a una variable de tipo teléfono. Luego llama

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

al método del usuario para que elimine ese teléfono. Una vez eliminado llama al método de update() de la clase padre y actualiza a ese usuario

- buscarTelefono: Este método recibe una variable de tipo entero en sus parámetros. Este método llama al método de búsqueda del usuario para encontrar un teléfono por el código. Una vez que haya encontrado a ese teléfono lo retorna.
- listarTelefonos: este método lo único que hace es retornar el listado de teléfonos que puede tener ese usuario

### 9. ecu.ups.edu.vista

En este paquete se crearon todas las interfaces para la interacción del usuario con el programa

- InicioSession
- ListarUsuarios
- MenuPrincipal
- MenuTelefono
- MenuUsuario
- RegistrarUsuario

### 10. InicioSession

Esta interfaz permite al usuario ingresar su correo y contraseña. Si es que el usuario registro sus datos el programa le dejara registrar teléfonos si es que no el programa le soltara una ventana de error diciéndole que no “existe el usuario”

Paquetes importados:

- import ec.edu.ups.controlador.ControladorUsuario;
- import javax.swing.JOptionPane;

atributos:

- private ControladorUsuario controladorU;
- private MenuPrincipal menuP;

en el constructor de esta interfaz solamente se le pasa la ventana principal y el controlador.

Métodos:

- limpiar: este método lo único que hace es que limpia las cajas de texto de cualquier tipo e texto que haya tenido anteriormente.


ActionPerformed

- btnIniciarSessionActionPerformed: esta acción primero revisa que todas las cajas de texto tengan un contenido. Una vez revisado que las cajas de texto tengan contenido procede a extraer esos datos y los pasa al controlador para que compruebe si es que existe este usuario dentro del programa. Si es que existe el usuario las demás ventanas se visualizaran y si es que no soltara una ventana de error.
- btnCerrarActionPerformed: esta acción lo único que hace es limpiar las cajas de texto.

### 11. ListarUsuarios

Esta interfaz permite que el usuario encuentre la información personal de un usuario al ingresar su numero de cedula. Esta interfaz también permite que listen todos los numero de teléfonos que existen en el programa

Paquetes importados:

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

- `import ec.edu.ups.controlador.*;`
- `import ec.edu.ups.modelo.Telefono;`
- `import ec.edu.ups.modelo.Usuario;`
- `import java.util.Collection;`
- `import java.util.List;`
- `import javax.swing.JOptionPane;`
- `import javax.swing.table.DefaultTableModel;`

atributos:

- `private ControladorGeneral controlIG;`
- `private ControladorUsuario controladorU;`

en el constructor de esta interfaz se pasa la ventana principal y el controlador.

Métodos:

- `actualizarVista`: este método recibe una lista en sus parámetros. Lo que hace este método es actualizar la tabla con los números de teléfonos que pueda tener un solo usuario.
- `TodosLosTelefonos`: este método recibe una colección en sus parámetros. Lo que hace este método es actualizar la tabla con todos los números de teléfonos que existen dentro del programa
- `Limpiar`: este método lo único que hace es que limpia las cajas de texto.

ActionPerformed

- `btnListarActionPerformed`: esta acción lo único que hace es conseguir la colección de teléfonos del controlador y lo manda a que se visualice en la tabla
- `btnBuscarActionPerformed`: esta acción primero revisa que la caja de texto tenga algún contenido, una vez que se haya comprobado que, si se ha llenado la caja de texto, llama al controlador y busca al usuario por la cedula. Si no encuentra al usuario suelta una ventana de error, pero si es que si lo encuentra llama a otro método del controlador y encuentra todos los números de teléfonos que pueda tener ese único usuario.

## 12.MenuPrincipal

Esta interfaz permitirá que el usuario pueda visualizar todas las demás interfaces.


Paquetes importados:

- `private InicioSession inicioSession;`
- `private ListarUsuarios listarUsuario;`
- `private MenuTelefono menuTelefono;`
- `private MenuUsuario menuUsuario;`
- `private RegistrarUsuario registrarUsuario;`

atributos:

- `private ControladorGeneral controladorGeneral;`
- `private ControladorUsuario controladorUsuario;`

en el constructor de esta interfaz se instancian todos los atributos y todas las demás interfaces. De igual manera se le añade a esta interfaz principal todas las demás interfaces secundarias.

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

#### Métodos:

- `getTabMenu`: este método retorna el `JMenu (tabMenu)`
- `getIniciarSession`: este método retorna el `JMenuItem(tabIniciarSession)`
- `getregistrarUsuario`: este método retorna el `JMenuItem(tabRegistrarU)`
- `getCerrarSession`: este método retorna el `JMenuItem(tabCerrarSession)`

#### Action Performed

- `exitMenuItemActionPerformed`: Esta acción termina de correr el programa
- `tabRegistrarUActionPerformed`: Esta acción visualiza la venta de registrarUsuario
- `tabCerrarSessionActionPerformed`: esta accion oculta el `JMenu(TabMenu)` y oculta el `JMenuItem(tabCerrarSession)`. Por último, visualiza las `JMenuItem (tabIniciarSession, tabRegistrarU)`
- `tabIniciarSessionActionPerformed`: Esta acción visualiza la interfaz (`inicioSession`)
- `tabListarActionPerformed`: Esta acción visualiza la interfaz (`listarUsuario`)
- `tabMenuUsuarioActionPerformed`: Esta acción visualiza la interfaz (`menuUsuario`)
- `tabMenuTelefonoActionPerformed`: Esta acción visualiza la interfaz (`menuTelefono`)

### 13. MenuTelefono

Esta interfaz permitirá que el usuario pueda registrar todos los teléfonos que desee, de la misma manera se le permite al usuario buscar, modificar y eliminar números de teléfonos que se haya creado.

#### Paquetes importados

- `import ec.edu.ups.controlador.ControladorGeneral;`
- `import ec.edu.ups.controlador.ControladorUsuario;`
- `import ec.edu.ups.modelo.Telefono;`
- `import javax.swing.JOptionPane;`
- `import javax.swing.table.DefaultTableModel;`

#### atributos:

- `private ControladorGeneral controladorG;`
- `private ControladorUsuario controladorU;`

En el constructor de esta interfaz se instancian los controladores y un método llamado (`activarBotones`)

#### Métodos:

- `actualizarVista`: este método lo único que hace es actualizar los datos de la tabla cada vez que el usuario interactúe con el programa.
- `cargarSiguieteCodigo`: este método consigue el siguiente código para el teléfono
- `limpiar`: este método limpia las cajas de texto en el área de registrar teléfonos.
- `limpiarModificacion`: Este método limpia las cajas de texto en el área de modificación del teléfono
- `activarBotones`: este método activa ciertos botones
- `desactivarBotones`: este método desactiva ciertos botones.

#### Action Performed

- `btnRegistrarActionPerformed`: esta acción lo que hace es recolectar todos los datos de las cajas de texto en el área de registrar teléfono. Después llama a controlador y crea un teléfono

- `cmbxTipoActionPerformed`: Esta acción formatea la manera en la cual se debe ingresar los datos en el cajón de `JFormattedTextField`.
- `btnBuscarActionPerformed`: Esta acción obtiene el código ingresado por el usuario en el cajón de búsqueda y señala en la tabla de datos donde está el teléfono con ese código.
- `btnEliminarActionPerformed`: Esta acción obtiene el código del cajón de texto de búsqueda y llama al controlador para que lo elimine. Una vez eliminado el teléfono actualiza la vista de la tabla de datos.
- `btnCancelarActionPerformed`: esta acción lo único que hace es desactivar ciertos botones y activar el botón de registrar.
- `btnActualizarActionPerformed`: Esta acción obtiene todos los datos de los cajones de texto del área de modificar y llama al controlador a que actualice esos datos. Por último, actualiza la vista de la tabla de datos.
- `cmbxModificarTipoActionPerformed`: Esta acción formatea la manera en la cual se debe ingresar los datos en el cajón de `JFormattedTextField`.

#### 14.MenuUsuario

Esta interfaz permite que el usuario visualice sus datos personales y permite que modifique sus datos si es necesario.

Paquetes importados:

- `import ec.edu.ups.controlador.ControladorUsuario;`
- `import javax.swing.JOptionPane;`
- `import javax.swing.table.DefaultTableModel;`

Atributos:

- `private ControladorUsuario controladorU;`

el constructor de esta interfaz solamente instancia el controlador

métodos:

- `cargarDatos`: este método lo que hace es que carga los datos del usuario y los carga en las cajas de texto.
- `actualizarVista`: este método lo único que hace es actualizar la vista de la tabla de datos

Action Performed:

- `btnActualizarActionPerformed`: esta acción obtiene todos los datos de las cajas de texto y llama al controlador para que actualice al usuario.

#### 15.RegistrarUsuario

Esta interfaz permite que el usuario se registre en el programa permitiéndolo ingresar con sus datos.


Paquetes importados:

- `import ec.edu.ups.controlador.ControladorUsuario;`
- `import javax.swing.JOptionPane;`

el constructor de esta interfaz solamente instancia el controlador.

Métodos:

- `limpiar`: este método lo único que hace es limpiar todo texto existente en las cajas de texto de la interfaz

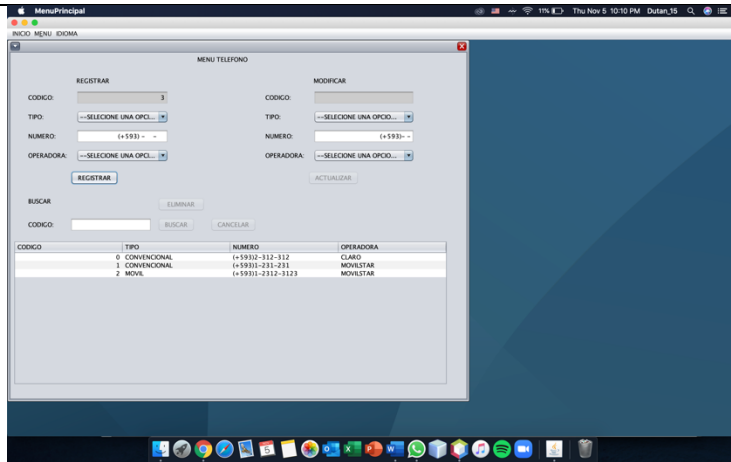
	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Action Performed:

- btnRegistrarActionPerformed: esta acción recolecta todos los datos del text box y llama al controlador para que cree un nuevo usuario.
- btnCancelarActionPerformed: esta acción solamente elimina cualquier contenido de los text box y los deja limpios.

**RESULTADO(S) OBTENIDO(S):**





CODIGO	TIPO	NUMERO	OPERADORA
0	CONVENCIONAL	0-5932-312-312	CLARO
1	CONVENCIONAL	0-5931-231-231	MOVISTAR
2	MOVIL	0-5931-232-312	MOVISTAR

```

public abstract class ControladorGenerico<T> {
    private List<T> lista;
    private T t;

    public ControladorGenerico() {
        lista=new ArrayList<>();
    }

    public void crear(T objeto){
        lista.add(objeto);
    }

    public T read(Object obj){
        return lista.stream().filter(tipo-> tipo.equals(obj)).findFirst().get();
    }

    public void update(T objeto){
        for (int i = 0; i < lista.size(); i++) {
            T nuevo=lista.get(i);
            if (nuevo.equals(objeto)){
                lista.add(i, objeto);
            }
        }
    }


    public void delete(T objeto){
        Iterator<T> it =lista.iterator();
        while (it.hasNext()) {
            T u = it.next();
            if(u.equals(objeto)){
                it.remove();
                break;
            }
        }
    }
}

```

```

55
56 public List<T> findAll(){
57     return lista;
58 }
59
60 public T encontrarObjeto(){
61     return t;
62 }
63 }
64

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

**CONCLUSIONES:** en conclusión, se debe entender como manejar exitosamente las clases genéricas y como aplicarlas con interfaces.

**RECOMENDACIONES:**

**Nombre de estudiante:** \_\_\_\_\_ Denys Dutan \_\_\_\_\_

**Firma de estudiante:** \_\_\_\_\_  \_\_\_\_\_