

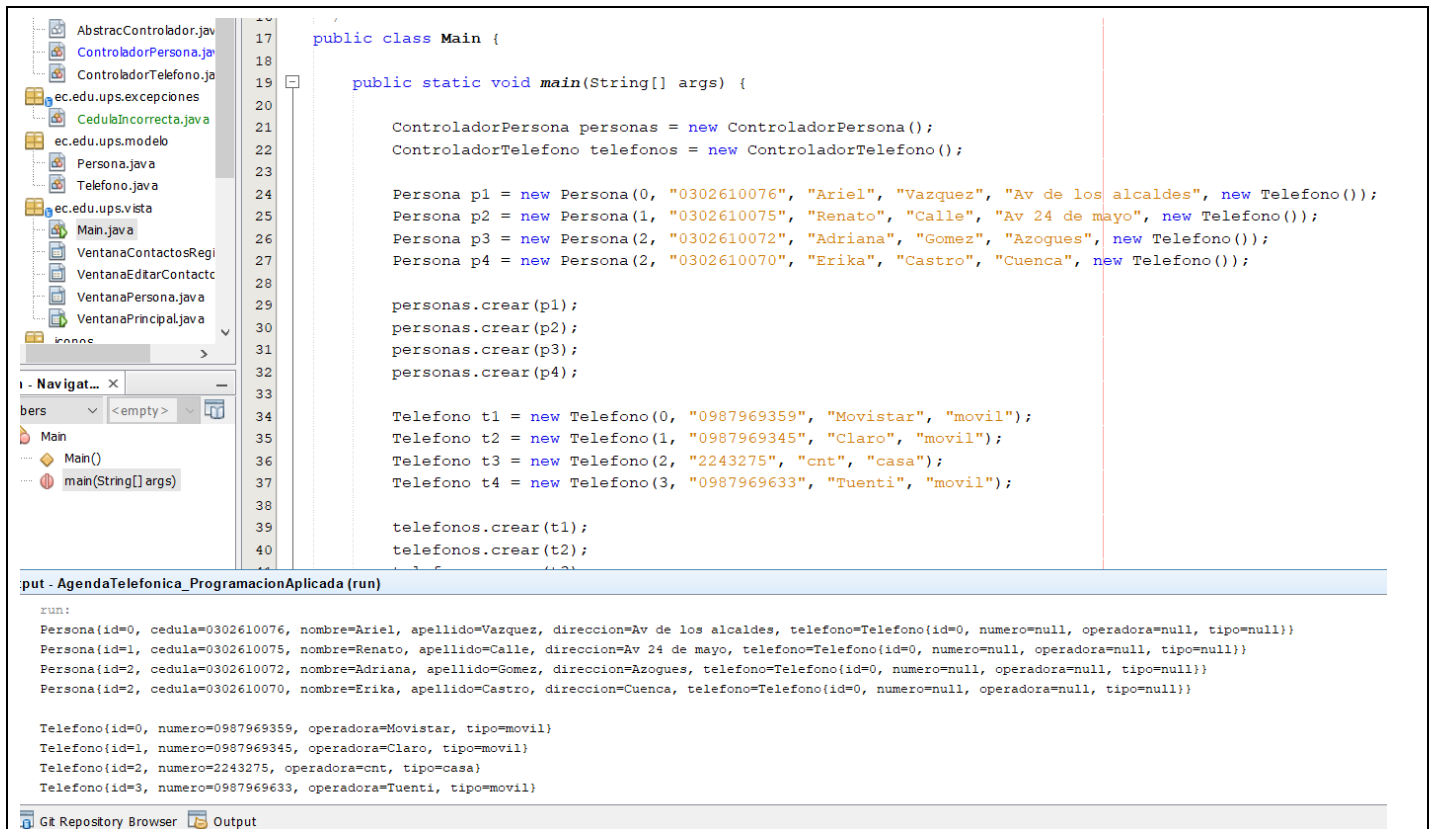
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		PRÁCTICA DE LABORATORIO	
CARRERA: COMPUTACIÓN		ASIGNATURA: Programación Aplicada	
NRO. PRÁCTICA:	3	TÍTULO PRÁCTICA: Reflexión en Java	
<b>OBJETIVO ALCANZADO:</b> <p>En esta Practica se pudo identificar y aplicar los nuevos cambios importantes en Java</p> <p>Se diseño e implemento las nuevas técnicas de programación</p> <p>Se implemento las nuevas características de Java</p>			
<b>ACTIVIDADES DESARROLLADAS</b>			
<p>1. Revisar la teoría y conceptos de Java 8, 9, 10, 11, 12, 13, 14, 15</p> <ul style="list-style-type: none"> <li>Todos estos cambios los hemos venido revisando y aplicando en todas las clases de Programación Aplicada, así como también en las anteriores prácticas.</li> </ul>			
<p>2. Diseñar e implementar las características de Java para generar la impresión de cualquier lista, de los modelos que tengan el campo id generar automáticamente</p> <ul style="list-style-type: none"> <li>Se continuo con la Practica anterior (Agenda Telefónica) en donde se implementó un nuevo método imprimirListadoEnConsola() para probar dicho método se crean una clase Main en donde se ejecutó dicho método.</li> </ul> <pre> public void imprimirListadoEnConsola() {     listado.forEach(System.out::println); }  personas.imprimirListadoEnConsola(); System.out.println(""); telefonos.imprimirListadoEnConsola(); System.out.println(""); </pre>			

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



```

17 public class Main {
18
19     public static void main(String[] args) {
20
21         ControladorPersona personas = new ControladorPersona();
22         ControladorTelefono telefonos = new ControladorTelefono();
23
24         Persona p1 = new Persona(0, "0302610076", "Ariel", "Vazquez", "Av de los alcaldes", new Telefono());
25         Persona p2 = new Persona(1, "0302610075", "Renato", "Calle", "Av 24 de mayo", new Telefono());
26         Persona p3 = new Persona(2, "0302610072", "Adriana", "Gomez", "Azogues", new Telefono());
27         Persona p4 = new Persona(2, "0302610070", "Erika", "Castro", "Cuenca", new Telefono());
28
29         personas.crear(p1);
30         personas.crear(p2);
31         personas.crear(p3);
32         personas.crear(p4);
33
34         Telefono t1 = new Telefono(0, "0987969359", "Movistar", "movil");
35         Telefono t2 = new Telefono(1, "0987969345", "Claro", "movil");
36         Telefono t3 = new Telefono(2, "2243275", "cnt", "casa");
37         Telefono t4 = new Telefono(3, "0987969633", "Tuenti", "movil");
38
39         telefonos.crear(t1);
40         telefonos.crear(t2);
41     }
42 }

```

put - AgendaTelefonica\_ProgramacionAplicada (run)

```

RUN:
Persona(id=0, cedula=0302610076, nombre=Ariel, apellido=Vazquez, direccion=Av de los alcaldes, telefono=Telefono(id=0, numero=null, operadora=null, tipo=null))
Persona(id=1, cedula=0302610075, nombre=Renato, apellido=Calle, direccion=Av 24 de mayo, telefono=Telefono(id=0, numero=null, operadora=null, tipo=null))
Persona(id=2, cedula=0302610072, nombre=Adriana, apellido=Gomez, direccion=Azogues, telefono=Telefono(id=0, numero=null, operadora=null, tipo=null))
Persona(id=2, cedula=0302610070, nombre=Erika, apellido=Castro, direccion=Cuenca, telefono=Telefono(id=0, numero=null, operadora=null, tipo=null))

Telefono(id=0, numero=0987969359, operadora=Movistar, tipo=movil)
Telefono(id=1, numero=0987969345, operadora=Claro, tipo=movil)
Telefono(id=2, numero=2243275, operadora=cnt, tipo=casa)
Telefono(id=3, numero=0987969633, operadora=Tuenti, tipo=movil)

```


- Se creo un método generarID() dentro del AbstracControlador el cual nos permitirá generar el id automáticamente para las clase que contengan el mismo como atributo

```

public int generarId() {
    if(listado.size() > 0) {
        return listado.indexOf(listado.get(listado.size() - 1 )) + 1;
    }
    return 1;
}

```

- Probar y modificar el método validar para que nos permita utilizar excepciones, además de modificar el buscar para controlar el nullpointerexception

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

- Se creó un nuevo paquete (ec.edu.ups.excepciones) en el cual se creó la Clase CedulaIncorrecta que hereda de Exception

```

package ec.edu.ups.excepciones;

/**
 *
 * @author ariel
 */
public class CedulaIncorrecta extends Exception{

    public CedulaIncorrecta() {}

    public CedulaIncorrecta(String mensaje) {
        super(mensaje);
    }
}

```

- Se modificó el método validar() en ControladorPersona

```

@Override
public boolean validar(Persona persona) throws Exception {
    if (persona.getCedula().length() != 10) {
        throw new CedulaIncorrecta("El tamaño de la cedula es incorrecto");
    } else {
        return true;
    }
}
}


```

- Se modificó también el método crear()

```

public boolean crear(T objeto) {
    try {
        if (validar(objeto)) {
            return listado.add(objeto);
        }
        return false;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

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

- Presentando así la Excepción(si la cedula no tiene 10 caracteres) en ejecución



```

ec.edu.ups.excepciones.CedulaIncorrecta: El tamaño de la cedula es incorrecto
    at ec.edu.ups.controlador.ControladorPersona.validar(ControladorPersona.java:32)
    at ec.edu.ups.controlador.ControladorPersona.validar(ControladorPersona.java:17)
    at ec.edu.ups.controlador.AbstracControlador.crear(AbstracControlador.java:27)
    at ec.edu.ups.vista.Main.main(Main.java:30)
Persona(id=0, cedula=0302610076, nombre=Ariel, apellido=Vazquez, direccion=Av de los alcaldes, telefono=Telefono(id=0, numero=null, operadora=null, tipo=null))
Persona(id=2, cedula=0302610072, nombre=Adriana, apellido=Gomez, direccion=Azogues, telefono=Telefono(id=0, numero=null, operadora=null, tipo=null))
Persona(id=2, cedula=0302610070, nombre=Erika, apellido=Castro, direccion=Cuenca, telefono=Telefono(id=0, numero=null, operadora=null, tipo=null))

```

- Se modifico el metodo buscar de la clase abstracta para controlar el nullpointerexception.

```

public T buscar(T objetoBuscado) {
    return listado.stream().filter(objeto -> objeto.equals(objetoBuscado)).findFirst().orElse(null);
}

```

- Realizar práctica codificando los códigos de las nuevas características de Java y su uso dentro de una agenda telefónica.
- En esta práctica se continuó agregando, modificando e implementando nuevas funciones dentro de la agenda telefónica que veníamos programando

[https://github.com/VazquezAriel/AgendaTelefonica\\_ProgramacionAplicada/commits/master](https://github.com/VazquezAriel/AgendaTelefonica_ProgramacionAplicada/commits/master)

#### RESULTADO(S) OBTENIDO(S):

Realizar procesos de investigación sobre los cambios importantes de Java

Entender las aplicaciones de codificación de las nuevas características en base a la programación genérica

Entender las funcionalidades adicionales de Java.


#### CONCLUSIONES:

En esta práctica se continuó probando las nuevas funcionalidades de las nuevas versiones de Java.

Se trabajo dentro de plazos de tiempo establecidos, manejando el lenguaje de programación de Java.

#### RECOMENDACIONES:

Realizar el trabajo dentro del tiempo establecido.

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

**Nombre de estudiante:** Ariel Vazquez

**Firma de estudiante:**

