

CARRERA: COMPUTACIÓN		ASIGNATURA: Programación Aplicada
NRO. PRÁCTICA:	2	TÍTULO PRÁCTICA: Clase Genéricas en Java
OBJETIVO ALCANZADO: <ul style="list-style-type: none">• Identificar los cambios importantes de Java• Diseñar e Implementar las nuevas técnicas de programación• Entender la cada uno de las características nuevas en Java		
ACTIVIDADES DESARROLLADAS		
<p>1. Revisar la teoría y conceptos de Java 8, 9 ,10, 11, 12 Dentro de los conceptos de las versiones de Java, podemos encontrar <i>Predicate</i>, la cual es una interface funcional que define una condición que un objeto determinado debe cumplir; también se revisó acerca de los <i>Streams</i>, que son una secuencia de elementos que soportan operaciones de agregación secuencial y paralela, un Stream es similar a una colección, permitiendo además el realizar operaciones directamente sobre el Stream</p>		
<p>2. Diseñar e implementar las características de Java para generar una abstracción que permita realizar un CRUD.</p>		

```

package ups.edu.ec.controlador;

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;
import java.util.function.Predicate;

/**
 *
 * @author Usuario
 */
public class ControladorAgendaTelefonica <Lis>{
    private List<Lis> lista;

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

    public void create(Lis obj){
        lista.add(obj);
    }

    public Optional<Lis> read(Predicate<Lis> pr){
        return lista.stream().filter(pr).findFirst();
    }

    public boolean update(Lis obj, Lis objetoNuevo){
        int posicion = lista.indexOf(obj);
        lista.set(posicion, objetoNuevo);
        return false;
    }

    public boolean delete(Lis obj){
        if(lista.contains(obj)){
            lista.remove(obj);
            return true;
        }
        return false;
    }

    public List<Lis> findAll(){
        return lista;
    }
}

```

3. Probar su funcionamiento y rendimiento dentro de los equipos de cómputo de programación genérica y ordenar una lista, buscar.

Usuarios

Cédula:

Nombre:

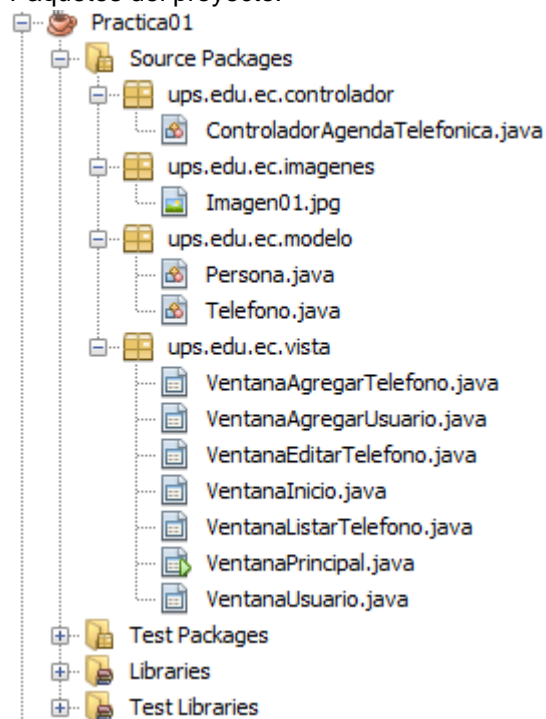
Apellido:

Dirección:

Cédula	Nombre	Apellido	Dirección
0105829246	Nelson	Ortega	Zaldumbide
0105829253	Carolina	Ortiz	Ciudadela
0105829287	Angel	Honorio	Rio Palora

4. Realizar práctica codificando los códigos de las nuevas características de Java y su uso dentro de una agenda telefónica.

Paquetes del proyecto:



ControladorAgendaTelefónica:

```

package ups.edu.ec.controlador;

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;
import java.util.function.Predicate;

/**
 * @author Usuario
 */
public class ControladorAgendaTelefonica <Lis>{
    private List<Lis> lista;

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

    public void create(Lis obj){
        lista.add(obj);
    }

    public Optional<Lis> read(Predicate<Lis> pr){
        return lista.stream().filter(pr).findFirst();
    }

    public boolean update(Lis obj, Lis objetoNuevo){
        int posicion = lista.indexOf(obj);
        lista.set(posicion, objetoNuevo);
        return false;
    }

    public boolean delete(Lis obj){
        if(lista.contains(obj)){
            lista.remove(obj);
            return true;
        }
        return false;
    }

    public List<Lis> findAll(){
        return lista;
    }
}

```

Ups.edu.ec.modelo

Persona:

```
package ups.edu.ec.modelo;

import java.util.Objects;

/**
 *
 * @author Usuario
 */
public class Persona {

    private String cedula;
    private String nombre;
    private String apellido;
    private String direccion;

    public Persona() {
        this.cedula="";
        this.nombre="";
        this.apellido="";
        this.direccion="";
    }

    public Persona(String cedula, String nombre, String apellido,
        String direccion) {
        this.cedula = cedula;
        this.nombre = nombre;
        this.apellido = apellido;
        this.direccion = direccion;
    }
}
```

```
public String getCedula() {  
    return cedula;  
}  
  
public void setCedula(String cedula) {  
    this.cedula = cedula;  
}  
  
public String getNombre() {  
    return nombre;  
}  
  
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}  
  
public String getApellido() {  
    return apellido;  
}  
  
public void setApellido(String apellido) {  
    this.apellido = apellido;  
}  
  
public String getDireccion() {  
    return direccion;  
}  
  
public void setDireccion(String direccion) {  
    this.direccion = direccion;  
}
```

```

@Override
public int hashCode() {
    int hash = 3;
    hash = 37 * hash + Objects.hashCode(this.cedula);
    return hash;
}

@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Persona other = (Persona) obj;
    if (!Objects.equals(this.cedula, other.cedula)) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Persona{" + "cedula=" + cedula + ", nombre=" +
        nombre + ", apellido=" + apellido +
        ", direccion=" + direccion + '}';
}

```

Telefono:

```
package ups.edu.ec.modelo;

import java.util.Objects;

/**
 * @author Usuario
 */
public class Telefono {

    private String codigo;
    private String numero;
    private String tipo;
    private String operadora;

    public Telefono() {
        this.codigo="";
        this.numero="";
        this.tipo="";
        this.operadora="";
    }

    public Telefono(String codigo, String numero,
        String tipo, String operadora) {
        this.codigo = codigo;
        this.numero = numero;
        this.tipo = tipo;
        this.operadora = operadora;
    }

    public String getCodigo() {
        return codigo;
    }
}
```



```
public void setCodigo(String codigo) {  
    this.codigo = codigo;  
}  
  
public String getNumero() {  
    return numero;  
}  
  
public void setNumero(String numero) {  
    this.numero = numero;  
}  
  
public String getTipo() {  
    return tipo;  
}  
  
public void setTipo(String tipo) {  
    this.tipo = tipo;  
}  
  
public String getOperadora() {  
    return operadora;  
}  
  
public void setOperadora(String operadora) {  
    this.operadora = operadora;  
}
```

```

@Override
public int hashCode() {
    int hash = 3;
    hash = 72 * hash + Objects.hashCode(this.codigo);
    return hash;
}

@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Telefono other = (Telefono) obj;
    if (!Objects.equals(this.codigo, other.codigo)) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Telefono{" + "codigo=" + codigo + ", numero=" + numero +
        ", tipo=" + tipo + ", operadora=" + operadora + '}';
}

```

ups.edu.ec.vista

VentanaAgregarTelefono:

Agregar Teléfono

Cédula/ID:

Número:

Tipo:

Operadora:

Aceptar

Cancelar

```

package ups.edu.ec.vista;

import java.util.List;
import javax.swing.JOptionPane;
import ups.edu.ec.controlador.ControladorAgendaTelefonica;
import ups.edu.ec.modelo.Telefono;

/**
 *
 * @author Usuario
 */
public class VentanaAgregarTelefono extends javax.swing.JFrame {

    ControladorAgendaTelefonica<Telefono> ctrAgendaTelefono;

    public VentanaAgregarTelefono(ControladorAgendaTelefonica controladorAgenda, String codigo) {
        initComponents();
        this.setLocationRelativeTo(null);

        ctrAgendaTelefono = controladorAgenda;
        jTextField1.setText(codigo);
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String codigo = jTextField1.getText();
        String numero = jTextField2.getText();
        String tipo = jTextField3.getText();
        String operadora = jTextField4.getText();

        if(codigo.isEmpty() || numero.isEmpty() || tipo.isEmpty() || operadora.isEmpty()){
            JOptionPane.showMessageDialog(this, "Complete los campos");
        }else{
            ctrAgendaTelefono.create(new Telefono(codigo, numero, tipo, operadora));
            JOptionPane.showMessageDialog(this, "Registro completo");
            vaciar();
            this.dispose();
        }

        List lista = ctrAgendaTelefono.findAll();
        for (Object object : lista) {
            System.out.println(object);
        }
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        vaciar();
        this.dispose();
    }

    public void vaciar(){
        jTextField1.setText("");
        jTextField2.setText("");
        jTextField3.setText("");
        jTextField4.setText("");
    }
}

```

VentanaAgregarUsuario:

Agregar Usuario

Cedula:

Nombre:

Apellido:

Dirección:

```
package ups.edu.ec.vista;

import javax.swing.JOptionPane;
import ups.edu.ec.controlador.ControladorAgendaTelefonica;
import ups.edu.ec.modelo.Persona;

/**
 *
 * @author Usuario
 */
public class VentanaAgregarUsuario extends javax.swing.JFrame {

    /**
     * Creates new form PantallaAgregarUsuario
     */

    ControladorAgendaTelefonica<Persona> ctrAgendaPersona;

    public VentanaAgregarUsuario(ControladorAgendaTelefonica controladorAgendaPersona) {
        initComponents();
        this.setLocationRelativeTo(null);

        this.ctrAgendaPersona = controladorAgendaPersona;
    }
}
```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String cedula = jTextField1.getText();
    String nombre = jTextField2.getText();
    String apellido = jTextField3.getText();
    String direccion = jTextField4.getText();

    if(cedula.isEmpty() || nombre.isEmpty() || apellido.isEmpty() || direccion.isEmpty()){
        JOptionPane.showMessageDialog(this, "Complete los campos");
    }else{
        ctrAgendaPersona.create(new Persona(cedula, nombre, apellido, direccion));
        JOptionPane.showMessageDialog(this, "Registro completo");
        vaciar();
        this.dispose();
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    vaciar();
    this.dispose();
}

public void vaciar() {
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
}

```

VentanaEditarTelefono:

Editar Teléfono

Cédula/ID:

Número:

Tipo:

Operadora:

```

package ups.edu.ec.vista;

import javax.swing.JOptionPane;
import ups.edu.ec.controlador.ControladorAgendaTelefonica;
import ups.edu.ec.modelo.Telefono;

/**
 *
 * @author Usuario
 */
public class VentanaEditarTelefono extends javax.swing.JFrame {

    ControladorAgendaTelefonica<Telefono> ctrAgendaTelefono;
    Telefono aux;
    public VentanaEditarTelefono(ControladorAgendaTelefonica controladorAgenda,
        Telefono telefono, String codigo) {
        initComponents();
        this.setLocationRelativeTo(null);

        ctrAgendaTelefono = controladorAgenda;
        aux = telefono ;
        jTextField1.setText(codigo);
    }

    private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String codigo = jTextField1.getText();
        String numero = jTextField2.getText();
        String tipo = jTextField3.getText();
        String operadora = jTextField4.getText();


        if(codigo.isEmpty() || numero.isEmpty() || tipo.isEmpty() || operadora.isEmpty()){
            JOptionPane.showMessageDialog(this, "Complete los campos");
        }else{
            ctrAgendaTelefono.update(aux ,new Telefono(codigo, numero, tipo, operadora));
            JOptionPane.showMessageDialog(this, "Actualización exitosa");
            vaciar();
            this.dispose();
        }
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        vaciar();
        this.dispose();
    }

    public void vaciar(){
        jTextField1.setText("");
        jTextField2.setText("");
        jTextField3.setText("");
        jTextField4.setText("");
    }
}

```

VentanaInicio:

— □ ×

Menú

Buscar Usuario:

Buscar

Nombre:

Apellido:

Cédula:

Dirección:

Teléfonos:

Listar

Limpiar

Número	Tipo	Operadora
--------	------	-----------

Agregar

Editar

Actualizar

Eliminar

```

package ups.edu.ec.vista;

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import ups.edu.ec.controlador.ControladorAgendaTelefonica;
import ups.edu.ec.modelo.Persona;
import ups.edu.ec.modelo.Telefono;

/**
 *
 * @author Usuario
 */
public class VentanaInicio extends javax.swing.JFrame {

    /**
     * Creates new form PantallaAgenda
     */

    ControladorAgendaTelefonica<Persona> ctrAgendaPersona;
    ControladorAgendaTelefonica<Telefono> ctrAgendaTelefono;

    VentanaUsuario ventanaUsuario;
    VentanaAgregarTelefono ventanaAgregarTelefono;
    VentanaEditarTelefono ventanaEditarTelefono;
    VentanaListarTelefono ventanaListarTelefonos;

    private String codigo;
    private Telefono aux;

```



```

public class VentanaInicio extends javax.swing.JFrame {

    /**
     * Creates new form PantallaAgenda
     */

    ControladorAgendaTelefonica<Persona> ctrAgendaPersona;
    ControladorAgendaTelefonica<Telefono> ctrAgendaTelefono;

    VentanaUsuario ventanaUsuario;
    VentanaAgregarTelefono ventanaAgregarTelefono;
    VentanaEditarTelefono ventanaEditarTelefono;
    VentanaListarTelefono ventanaListarTelefonos;

    private String codigo;
    private Telefono aux;

    public VentanaInicio(ControladorAgendaTelefonica controladorAgendaPersona,
        ControladorAgendaTelefonica controladorAgendaTelefono) {
        initComponents();
        this.setLocationRelativeTo(null);

        this.ctrAgendaPersona = controladorAgendaPersona;
        this.ctrAgendaTelefono = controladorAgendaTelefono;

        ventanaUsuario = new VentanaUsuario(this.ctrAgendaPersona);
        ventanaListarTelefonos = new VentanaListarTelefono(controladorAgendaTelefono);

        jTextField1.setToolTipText("Ingrese una cedula");

        deshabilitarBotones();
    }

```

```

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    ventanaUsuario.setVisible(true);
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String cedula = jTextField1.getText();

    Optional<Persona> persona = ctrAgendaPersona.read(p -> cedula.equals(p.getCedula()));

    if(persona.isPresent()){
        Persona pers = persona.get();
        jTextField2.setText(pers.getNombre());
        jTextField3.setText(pers.getApellido());
        jTextField4.setText(pers.getCedula());
        jTextField5.setText(pers.getDireccion());
        limpiarTabla();
        llenarTabla();
        activarParcialBotones();
    }else{
        JOptionPane.showMessageDialog(this, "Cedula no encontrada");
    }
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
    limpiarTabla();
    deshabilitarBotones();
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    codigo = jTextField4.getText();
    ventanaAgregarTelefono = new VentanaAgregarTelefono(this.ctrAgendaTelefono,
        codigo);
    ventanaAgregarTelefono.setVisible(true);
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    codigo = jTextField4.getText();
    ventanaEditarTelefono = new VentanaEditarTelefono(this.ctrAgendaTelefono,
        aux, codigo);
    ventanaEditarTelefono.setVisible(true);
    deshabilitarBotonesEditar();
    activarParcialBotones();
    llenarTabla();
    jTable1.clearSelection();
}

```

```

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    llenarTabla();
}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    int pos = jTable1.getSelectedRow();

    String numero = (String) jTable1.getValueAt(pos, 0);
    String tipo = (String) jTable1.getValueAt(pos, 1);
    String operadora = (String) jTable1.getValueAt(pos, 2);

    codigo = jTextField4.getText();
    aux = new Telefono(codigo, numero, tipo, operadora);

    habilitarBotonesEditar();
    deshabilitarParcialBotones();
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    boolean eliminar = ctrAgendaTelefono.delete(aux);

    if(eliminar){
        JOptionPane.showMessageDialog(this, "Se ha eliminado con éxito");
    }else{
        JOptionPane.showMessageDialog(this, "No ha sido posible eliminar");
    }
    deshabilitarBotonesEditar();
    activarParcialBotones();
    llenarTabla();
    jTable1.clearSelection();
}

```

```

public void limpiarTabla(){
    DefaultTableModel modelo = (DefaultTableModel) jTable1.getModel();
    modelo.setRowCount(0);
    for (int i = 0; i < modelo.getRowCount(); i++) {
        modelo.removeRow(i);
    }
}

public void llenarTabla(){
    DefaultTableModel modelo = (DefaultTableModel) jTable1.getModel();
    modelo.setRowCount(0);
    codigo = jTextField4.getText();

    List<Telefono> listaTelefono = ctrAgendaTelefono.findAll();
    List<Telefono> lista= new ArrayList<>();

    for (Telefono telefono : listaTelefono) {
        if(codigo.equals(telefono.getCodigo()))
            lista.add(telefono);
    }

    if(lista.isEmpty()){
        JOptionPane.showMessageDialog(this, "Este usuario no presenta teléfonos");
    }else{
        for (Telefono telefono : lista) {
            Object[] objeto = {telefono.getNumero(),
                                telefono.getTipo(), telefono.getOperadora()};
            modelo.addRow(objeto);
        }
    }
}
}

```

VentanaListarTelefono:

```

package ups.edu.ec.vista;

import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import ups.edu.ec.controlador.ControladorAgendaTelefonica;
import ups.edu.ec.modelo.Telefono;

/**
 *
 * @author Usuario
 */
public class VentanaListarTelefono extends javax.swing.JFrame {

    /**
     * Creates new form PantallaListarTelefonos
     */
    ControladorAgendaTelefonica<Telefono> ctrAgenda;
    public VentanaListarTelefono(ControladorAgendaTelefonica controladorAgenda) {
        initComponents();
        this.setLocationRelativeTo(null);
        this.ctrAgenda = controladorAgenda;
    }
}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    List<Telefono> lista = ctrAgenda.findAll();

    DefaultTableModel modelo = (DefaultTableModel) jTable1.getModel();
    modelo.setRowCount(0);

    if(!lista.isEmpty()){
        for (Telefono telefono : lista) {
            Object[] objeto = {telefono.getCodigo(), telefono.getNumero(),
                                telefono.getTipo(), telefono.getOperadora()};
            modelo.addRow(objeto);
        }
    }else{
        JOptionPane.showMessageDialog(this, "No existe telefonos agregados");
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

```

VentanaPrincipal:



```

package ups.edu.ec.vista;

import ups.edu.ec.controlador.ControladorAgendaTelefonica;
import ups.edu.ec.modelo.Persona;
import ups.edu.ec.modelo.Telefono;

/**
 *
 * @author Usuario
 */
public class VentanaPrincipal extends javax.swing.JFrame {

    /**
     * Creates new form VentanaPrincipal
     */
    ControladorAgendaTelefonica<Persona> ctrAgendaPersona;
    ControladorAgendaTelefonica<Telefono> ctrAgendaTelefono;
    VentanaInicio pantallaAgenda;

    public VentanaPrincipal() {
        initComponents();
        this.setLocationRelativeTo(null);

        ctrAgendaPersona = new ControladorAgendaTelefonica<>();
        ctrAgendaTelefono = new ControladorAgendaTelefonica<>();

        pantallaAgenda = new VentanaInicio(ctrAgendaPersona, ctrAgendaTelefono);

    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)
        //</editor-fold>
        //</editor-fold>
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new VentanaPrincipal().setVisible(true);
            }
        });
    }
}

```

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:

- Aprenden a trabajar dentro de plazos de tiempo establecidos, manejando el lenguaje de programación de Java.
- Se pone en práctica lo revisado acerca de las actualizaciones de Java

RECOMENDACIONES:

- Realizar el trabajo dentro del tiempo establecido.
- Siempre revisar la página oficial de Java para el lanzamiento de nuevas versiones y corrección de errores de la versión

Nombre de estudiante: Nelson Paul Ortega Segarra

A handwritten signature in black ink, appearing to be 'NP', enclosed within a circular loop. The signature is written on a light green, textured background.

Firma de estudiante: