PRUEBA 1 MÓDULO 3
Herramientas Aplicadas para el desarrollo de Java.

C O N T E X T O

Después de una intensa etapa de desarrollo de software, es necesario probar el software creado aplicándole pruebas unitarias y adhiriendo el proyecto a un repositorio de control de versiones.

1. Considere la clase **ColeccionVentas** la cual tiene 4 métodos:
   - ✓ boolean **Agregar**(Venta v): Agrega un objeto del tipo Dato a la colección.
   - ✓ boolean E**liminar**(String codigo): Ubica mediante el código y elimina un objeto de la colección.
   - ✓ ArrayList<Venta> **Listar**(): Retorna una lista con todos los objetos de la colección.
   - ✓ boolean **Modificar**(Venta v): Modifica un objeto de la colección.

   La definición de la clase **Venta** es la siguiente:

```
public class Venta {
   private String codigo;
   private int precio;
   private int cantidad;

   public Venta() {
   }

   public Venta(String codigo, int precio, int cantidad) {
      this.codigo = codigo;
      this.precio = precio;
      this.cantidad = cantidad;
   }

   public String getCodigo() {
      return codigo;
   }
   public void setCodigo(String codigo) {
      this.codigo = codigo;
   }
   public int getPrecio() {
      return precio;
   }
   public void setPrecio(int precio) {
      this.precio = precio;
   }
   public int getCantidad() {
      return cantidad;
   }
   public void setCantidad(int cantidad) {
      this.cantidad = cantidad;
   }
}
```

La definición de la clase **ColeccionVentas** es la siguiente:

```
public class ColeccionVentas {
   private ArrayList<Venta> base;

   public ColeccionVentas() {
      this.base = new ArrayList<Venta>();
   }

   public ArrayList<Venta> Listar() {
      return this.base;
   }
   public boolean Agregar(Venta venta) {
      return this.base.add(venta);
   }
   public boolean modificar(Venta venta) {
      for (int i = 0; i < this.base.size(); i++) {
         if (this.base.get(i).getCodigo().equals(venta.getCodigo())) {
            this.base.get(i).setCodigo(venta.getCodigo());
            this.base.get(i).setPrecio(venta.getPrecio());
            this.base.get(i).setCantidad(venta.getCantidad());
            return true;
         }
      }
      return false;
   }
   public boolean Eliminar(String codigo) {
      for (int i = 0; i < this.base.size(); i++) {
         if (this.base.get(i).getCodigo().equals(codigo)) {
            this.base.remove(i);
            return true;
         }
      }
      return false;
   }
}
```

TESTS UNITARIOS

- ✓ Construya un **proyecto aplicación java** llamado **P01Mod3Ventas**.
- ✓ Incluya en el proyecto las clases antes descritas (Venta y ColeccionVentas)
- ✓ En la clase **ColeccionVentas, agregar un método llamado "subTotal"** que nos entregue el subtotal de una venta (precio*cantidad)
- ✓ Crear una clase para los test unitarios que se describen a continuación:
  - Listar
  - Agregar
  - Modificar
  - Eliminar
  - subTotal

(25 puntos)

GIT

- ✓ Cree un **nuevo repositorio público llamado P01Mod3Ventas** en su cuenta GitHub y realice lo siguiente:
  - Crear un nuevo repositorio local.
  - Agregar el proyecto completo al repositorio local.
  - Realizar commit de todos los archivos al repositorio local con un mensaje distintivo.
  - Realizar un push hacia el repositorio remoto
  - Verificar que todos los archivos hayan quedado disponibles en el repositorio remoto de GitHub.

(25 puntos)

2. Del siguiente repositorio GitHub https://github.com/ropoba/Clase20190927.git, descarge el proyecto java **EmpleadosP01Mod3** el cual implementa un CRUD sobre de una tabla empleados

CLASE EMPLEADO

- ✓ Crear un método llamado **"montoBono"**, este método debe calcular que cifra corresponde a cada empleado que tenga más de 3 años de antigüedad y el bono se calcula de la siguiente forma: antigüedad*50000

TESTS UNITARIOS

- ✓ Antes se realizar los test unitarios lee y siga las instrucciones de apartado GIT
- ✓ Realice los tests unitarios necesarios para asegurar el correcto funcionamiento del repositorio.
- ✓ Construya los tests unitarios para la clase **"BussEmpleado"**.
- ✓ Es necesario que realice pruebas para los métodos: **Crear, Buscar, Modificar, Eliminar y montoBono**.
- ✓ Implemente **una clase de tests** unitarios **por cada método** mencionado.
- ✓ Agrupe todos los tests unitarios en un @.

(25 puntos)

GIT

- ✓ Cree un **nuevo repositorio público llamado EmpleadosP01Mod3** en su cuenta GitHub y realice lo siguiente:

  Primero:
  - Crear un nuevo repositorio local.
  - Agregar el proyecto completo al repositorio local.
  - Crear una rama llamada **Version1**
  - Realizar commit de todos los archivos al repositorio local con un mensaje "Versión original".
  - Realizar un push hacia el repositorio remoto con el comando

  Segundo:
  - Crear una rama llamada **VersionFinal**
  - Realizar commit de todos los archivos al repositorio local con un mensaje "Versión final".
  - Realizar un push hacia el repositorio remoto con el comando
  - Verificar que todos los archivos hayan quedado disponibles en el repositorio remoto de GitHub.

(25 puntos)

Puntaje total: 100 puntos