



Nombre: Gina Belen Molina Gutiérrez

Fecha: 1 Semana

Tema: Programa con clases abstractas, interfaces y polimorfismo.

Clases abstractas, interfaces y polimorfismo.

```
EloteMx.java  Principal.java  Doriesquites.java  AbstractElote.java X
1 package com.sinsingleton;
2
3 public abstract class AbstractElote implements EloteMx {
4     public void bienvenida() {
5         System.out.println("Bienvenido a Elotes Gutiérrez!");
6     }
7 }
8
```

Definición de la clase abstracta

Definimos una clase abstracta que implementa la interfaz EloteMx y proporciona algunas implementaciones comunes.

Implementación de la clase concreta

```
EloteMx.java X  Principal.java  Doriesquites.java
1 package com.sinsingleton;
2
3 public interface EloteMx {
4     void servidoEn();
5     void acompañadoDe();
6     void precio();
7 }
```

Definición de la interfaz

Descripción:

Se define una interfaz EloteMx que declara tres métodos: servidoEn(), acompañadoDe(), y precio(). Las clases que implementen esta interfaz deberán proporcionar implementaciones para estos métodos.





Clases abstractas, interfaces y polimorfismo.

```
EloteMx.java  Principal.java X  Doriesquites.java
1 package com.sinsingleton;
2
3 public class Principal {
4     public static void main(String[] args) {
5         // Creación de objetos
6         EloteMx doriesquites = new Doriesquites(); // Referencia de tipo de la interfaz
7
8         // Llamada a los métodos
9         doriesquites.servidoEn();
10        doriesquites.acompañadoDe();
11        doriesquites.precio();
12    }
13 }
14
```

Clase principal

Descripción:

La clase Principal contiene el método main, que es el punto de entrada del programa.

Se crea un objeto Doriesquites usando una referencia de tipo EloteMx.

Se llaman los métodos servidoEn(), acompañadoDe(), y precio() del objeto Doriesquites.

```
EloteMx.java  Principal.java  Doriesquites.java X
1 package com.sinsingleton;
2
3 public class Doriesquites implements EloteMx {
4     @Override
5     public void servidoEn() {
6         System.out.println("Doritos preparados con Esquite");
7     }
8     @Override
9     public void acompañadoDe() {
10        System.out.println("Elote preparado con todo");
11    }
12    @Override
13    public void precio() {
14        System.out.println("Tu pedido tiene un costo de $65");
15    }
16 }
```

Resumen:

1. Clase abstracta (AbstractElote): Proporciona un lugar para métodos comunes que todas las clases concretas pueden heredar.
2. Clase concreta (Doriesquites): Extiende la clase abstracta y proporciona implementaciones específicas de los métodos definidos en la interfaz.
3. Clase principal (Principal): Crea objetos y llama a los métodos, aprovechando cualquier funcionalidad común proporcionada por la clase abstracta.

