

UNIVERSIDAD MADERO PUEBLA



Seleccionadora y control de cortinas para un invernadero.

Alumno: Emigdio Espinosa Jasso.

Num:7979763883.

Matricula: 20086534.

Carrera: Ingeniería de Software.

Materia: Patrones de diseño.

Maestro: Nicolás Arrioja Landa Cosio.

**Introducción.**

Este es un proyecto de la materia de patrones de diseño, que consiste en crear un programa aplicando patrones de diseño de software.

El proyecto va enfocado en crear un programa para llevar un buen control de mi centro de alto rendimiento. Este programa nos ayudara a agregar más sucursales, así como para inscribir a nuevos niños que se integren al equipo. También nos ayudara a llevar un buen control de pagos.

**Antecedentes**

Para saber cómo podría funcionar el programa, se utilizaron los ejemplos de otros programas que hicimos con los mismos patrones de diseño, para que de esta manera entendiéramos como es que funcionaban y pudiéramos implementarlos de la mejor manera.

**Descripción del problema.**

Tengo un proyecto de centro de alto rendimiento (enfocado en el fútbol) en Lomas de Angelopolis.

En este a veces se presenta el problema de que es difícil llevar un control de todos los niños que pertenecen al equipo, así como el control de los pagos.

**Objetivo general.**

Crear un programa que me ayude a llevar un mejor control del centro de alto rendimiento, así como la posibilidad de agregar más centros de alto rendimiento. También nos ayudara a llevar el control de pagos.

**Objetivo específico.**

Para agregar más sucursales y niños al equipo, ocupamos el patrón de diseño Componente.

Para llevar el control de los pagos, ocupamos el patrón de diseño Bridge.

**Alcances.**

Podemos agregar de manera rápida otras sucursales, así como niños al equipo. Podemos ver de manera rápida los pagos correspondientes por niño, así como por sucursal.

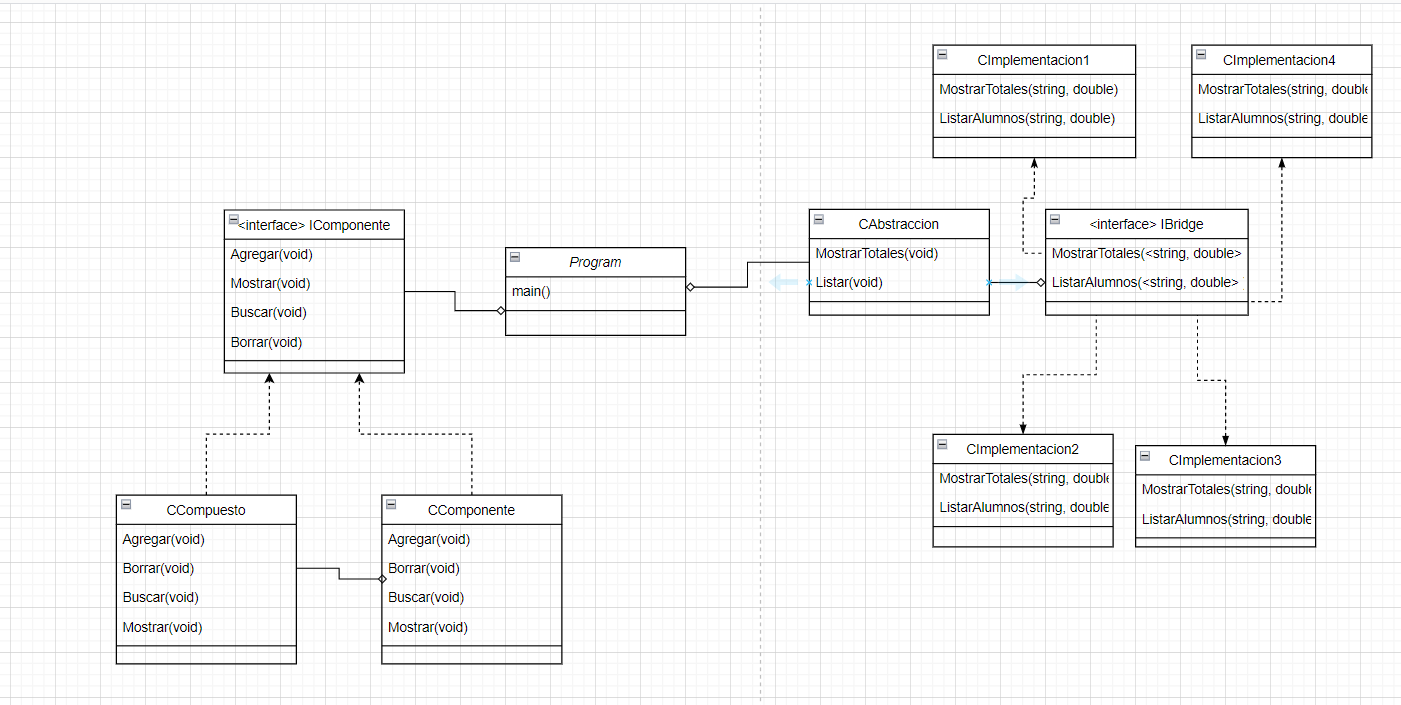
**Limitaciones.**

No puedes guardar todos los datos de los niños, como edad, sexo, categoría etc.

**Bibliografía.**

*Federación Mexicana de Fútbol. (s. f.). FMF. Recuperado 12 de octubre de 2022, de https://fmf.mx/SNC/Academia*

**Diagrama.**

****

**Código.**

using System.Collections.Generic;

namespace ProyectoPD02

{

///Clase CAbstraccion

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

internal class CAbstraccion

{

//Referencia a la implementacion

IBridge implementacion;

Dictionary<string, double> alumnos;

//Metodo 1, el mas usual

//Constructor, recibimos la implementacion concreta

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo 1. Constructor de la clase CAbstraccion

/// </summary>

/// <param name="pImp"></param>

/// <param name="pAlumn"></param>

public CAbstraccion(IBridge pImp, Dictionary<string, double> pAlum)

{

implementacion = pImp;

alumnos = pAlum;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo 2 nos sirve para llamar cada implementación

/// </summary>

/// <param name="pTipo"></param>

/// <param name="pAlumn"></param>

public CAbstraccion(int pTipo, Dictionary<string, double> pAlum)

{

if (pTipo == 1)

implementacion = new CImplementacion1();

if (pTipo == 2)

implementacion = new CImplementacion2();

if (pTipo == 3)

implementacion = new CImplementacion3();

if (pTipo == 4)

implementacion = new CImplementacion4();

alumnos = pAlum;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo de la interfaz que el usuario conoce que nos ayuda a mostrar los totales,

/// </summary>

public void MostrarTotales()

{

implementacion.MostrarTotales(alumnos);

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo de la interfaz que el usuario conoce, nos ayuda a litar los alumnos

/// </summary>

public void Listar()

{

implementacion.ListarAlumnos(alumnos);

}

}

}

using System.Collections.Generic;

namespace ProyectoPD02

{

///Clase CAbstraccion

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

internal class CAbstraccion

{

//Referencia a la implementacion

IBridge implementacion;

Dictionary<string, double> alumnos;

//Metodo 1, el mas usual

//Constructor, recibimos la implementacion concreta

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo 1. Constructor de la clase CAbstraccion

/// </summary>

/// <param name="pImp"></param>

/// <param name="pAlumn"></param>

public CAbstraccion(IBridge pImp, Dictionary<string, double> pAlum)

{

implementacion = pImp;

alumnos = pAlum;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo 2 nos sirve para llamar cada implementación

/// </summary>

/// <param name="pTipo"></param>

/// <param name="pAlumn"></param>

public CAbstraccion(int pTipo, Dictionary<string, double> pAlum)

{

if (pTipo == 1)

implementacion = new CImplementacion1();

if (pTipo == 2)

implementacion = new CImplementacion2();

if (pTipo == 3)

implementacion = new CImplementacion3();

if (pTipo == 4)

implementacion = new CImplementacion4();

alumnos = pAlum;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo de la interfaz que el usuario conoce que nos ayuda a mostrar los totales,

/// </summary>

public void MostrarTotales()

{

implementacion.MostrarTotales(alumnos);

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo de la interfaz que el usuario conoce, nos ayuda a litar los alumnos

/// </summary>

public void Listar()

{

implementacion.ListarAlumnos(alumnos);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProyectoPD02

{

///Clase CCompuesto

///Implementa IComponente

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

internal class CCompuesto<T> : IComponente<T>

{

//Lista donde guardamos los elementos que va a guardar

List<IComponente<T>> elementos;

//Identificador

public T Nombre { get; set; }

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo constructor de CCompuesto

/// </summary>

/// <param name="pNombre"></param>

public CCompuesto(T pNombre)

{

Nombre = pNombre;

//Instanciamos la lista

elementos = new List<IComponente<T>>();

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo que implementa IComponente, Nos sirve para adicionar elementos

/// </summary>

/// <param name="pElemento"></param>

public void Adicionar(IComponente<T> pElemento)

{

elementos.Add(pElemento);

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo que implementa IComponente, Nos sirve para borrar elementos

/// </summary>

/// <param name="pElemento"></param>

public IComponente<T> Borrar(T pElemento)

{

//Buscamos el elemento a borrar

IComponente<T> elemento = this.Buscar(pElemento);

//Si la encontramos, la eliminamos de la lista

if (elemento != null)

{

(this as CCompuesto<T>).elementos.Remove(elemento);

}

return this;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo que implementa IComponente, solo se pone para buscar elementos

/// </summary>

/// <param name="pElemento"></param>

public IComponente<T> Buscar(T pElemento)

{

//Si somos quien se busca nos regresamos

if (Nombre.Equals(pElemento))

return this;

IComponente<T> encontrado = null;

//Recorremos la lista y buscamos en nuestro elementos

foreach (IComponente<T> elemento in elementos)

{

encontrado = elemento.Buscar(pElemento);

if (encontrado != null)

break;

}

return encontrado;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo que implementa IComponente, Nos sirve para mostrar los elementos, junto con elementos que contenga

/// </summary>

/// <param name="pProfundidad"></param>

public string Mostrar(int pProfundidad)

{

//Construimos la cadena con una cantidad de - igual a la profundidad

StringBuilder infoElemento = new StringBuilder(new String('-', pProfundidad));

//Adicionamos la informacion del compuesto

infoElemento.Append( Nombre + " " + "\r\n");

//Adicionamos los elementos

foreach (IComponente<T> elemento in elementos)

infoElemento.Append(elemento.Mostrar(pProfundidad + 1));

return infoElemento.ToString();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProyectoPD02

{

///Clase CImplementacion1

///Implementa IBridge

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

internal class CImplementacion1 : IBridge

{

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo que nos muestra los totales de los alumnos

/// </summary>

/// <param name="pAlumnos"></param>

public void MostrarTotales(Dictionary<string, double> pAlumnos)

{

//Solo lo llamamos para que no muestre error el IBridge

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo que nos enlista los alumnos

/// </summary>

/// <param name="pAlumnos"></param>

public void ListarAlumnos(Dictionary<string, double> pAlumnos)

{

foreach (KeyValuePair<string, double> p in pAlumnos)

{

Console.WriteLine(p.Key);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProyectoPD02

{

///Clase CImplementacion2

///Implementa IBridge

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

internal class CImplementacion2 : IBridge

{

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo que nos muestra los totales de los alumnos

/// </summary>

/// <param name="pAlumnos"></param>

public void MostrarTotales(Dictionary<string, double> pAlumnos)

{

//Solo lo llamamos para que no muestre error el IBridge

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo que nos enlista los alumnos

/// </summary>

/// <param name="pAlumnos"></param>

public void ListarAlumnos(Dictionary<string, double> pAlumnos)

{

foreach (KeyValuePair<string, double> p in pAlumnos)

{

Console.ForegroundColor = ConsoleColor.Red;

if (p.Key[0] == 'L')

Console.WriteLine("{0}", p.Key);

}

foreach (KeyValuePair<string, double> p in pAlumnos)

{

Console.ForegroundColor = ConsoleColor.Blue;

if (p.Key[0] == 'M')

Console.WriteLine("{0}", p.Key, p.Value);

}

foreach (KeyValuePair<string, double> p in pAlumnos)

{

Console.ForegroundColor = ConsoleColor.Green;

if (p.Key[0] == 'A')

Console.WriteLine("{0}", p.Key);

}

Console.ForegroundColor = ConsoleColor.White;

Console.WriteLine("\r\n");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProyectoPD02

{

///Clase CImplementacion3

///Implementa IBridge

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

internal class CImplementacion3 : IBridge

{

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo que nos muestra los totales de los alumnos

/// </summary>

/// <param name="pAlumnos"></param>

public void MostrarTotales(Dictionary<string, double> pAlumnos)

{

double total = 0;

double totalm = 0;

double totall = 0;

double totala = 0;

int cantidad = 0;

foreach (KeyValuePair<string, double> p in pAlumnos)

{

total += p.Value;

if (p.Key[0] == 'L')

totall += p.Value;

if (p.Key[0] == 'M')

totalm += p.Value;

if (p.Key[0] == 'A')

totala += p.Value;

cantidad++;

}

Console.WriteLine("El total del centro deportivo Lomas es: ${0}", totall);

Console.WriteLine("El total del centro deportivo Momoxpan es: ${0}", totalm);

Console.WriteLine("El total del centro deportivo Aquixtla es: ${0}", totala);

Console.WriteLine("El total de las tres sucursales es: ${1}", cantidad, total);

Console.WriteLine("\r\n");

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo que nos enlista los alumnos

/// </summary>

/// <param name="pAlumnos"></param>

public void ListarAlumnos(Dictionary<string, double> pAlumnos)

{

foreach (KeyValuePair<string, double> p in pAlumnos)

{

Console.ForegroundColor = ConsoleColor.Red;

if (p.Key[0] == 'L')

Console.WriteLine("{0} - {1}", p.Key, p.Value);

}

foreach (KeyValuePair<string, double> p in pAlumnos)

{

Console.ForegroundColor = ConsoleColor.Blue;

if (p.Key[0] == 'M')

Console.WriteLine("{0} - {1}", p.Key, p.Value);

}

foreach (KeyValuePair<string, double> p in pAlumnos)

{

Console.ForegroundColor = ConsoleColor.Green;

if (p.Key[0] == 'A')

Console.WriteLine("{0} - {1}", p.Key, p.Value);

}

Console.ForegroundColor = ConsoleColor.White;

Console.WriteLine("\r\n");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProyectoPD02

{

///Clase CImplementacion4

///Implementa IBridge

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

internal class CImplementacion4 : IBridge

{

double descuento = 0;

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo que nos enlista los alumnos

/// </summary>

/// <param name="pAlumnos"></param>

public void ListarAlumnos(Dictionary<string, double> pAlumnos)

{

string leer = "";

Console.WriteLine("Cuál es el descuento a aplicar?");

leer = Console.ReadLine();

descuento = Convert.ToDouble(leer);

foreach (KeyValuePair<string, double> p in pAlumnos)

{

Console.ForegroundColor = ConsoleColor.Red;

if (p.Key[0] == 'L')

Console.WriteLine("{0} - {1}", p.Key, p.Value - (p.Value \* descuento / 100));

}

foreach (KeyValuePair<string, double> p in pAlumnos)

{

Console.ForegroundColor = ConsoleColor.Blue;

if (p.Key[0] == 'M')

Console.WriteLine("{0} - {1}", p.Key, p.Value - (p.Value \* descuento / 100));

}

foreach (KeyValuePair<string, double> p in pAlumnos)

{

Console.ForegroundColor = ConsoleColor.Green;

if (p.Key[0] == 'A')

Console.WriteLine("{0} - {1}", p.Key, p.Value - (p.Value \* descuento / 100));

}

Console.ForegroundColor = ConsoleColor.White;

Console.WriteLine("\r\n");

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 11-10-2022

///Versión: 1.0

/// <summary>

/// Metodo que nos muestra los totales de los alumnos

/// </summary>

/// <param name="pAlumnos"></param>

public void MostrarTotales(Dictionary<string, double> pAlumnos)

{

double total = 0;

double totalm = 0;

double totall = 0;

double totala = 0;

int cantidad = 0;

foreach (KeyValuePair<string, double> p in pAlumnos)

{

total += p.Value;

if (p.Key[0] == 'L')

totall += p.Value - (p.Value \* descuento / 100);

if (p.Key[0] == 'M')

totalm += p.Value - (p.Value \* descuento / 100);

if (p.Key[0] == 'A')

totala += p.Value - (p.Value \* descuento / 100);

cantidad++;

total = totall + totala + totalm;

}

Console.WriteLine("El total del centro deportivo Lomas con descuento es: ${0}", totall);

Console.WriteLine("El total del centro deportivo Momoxpan es: ${0}", totalm);

Console.WriteLine("El total del centro deportivo Aquixtla es: ${0}", totala);

Console.WriteLine("El total de las tres sucursales es: ${0}",total);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProyectoPD02

{

///Interface IBridge

///Autor: Emigdio Espinosa Jasso

///Fecha: 12-10-2022

///Versión: 1.0

internal interface IBridge

{

void MostrarTotales(Dictionary<string, double> pAlumnos);

void ListarAlumnos(Dictionary<string, double> pAlumnos);

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProyectoPD02

{

///Interface IComponente

///Autor: Emigdio Espinosa Jasso

///Fecha: 12-10-2022

///Versión: 1.0

internal interface IComponente <T>

{

T Nombre { get; set; }

void Adicionar(IComponente<T> pElemento);

IComponente<T> Borrar(T pElemento);

IComponente<T> Buscar(T pElemento);

string Mostrar(int pProfundidad);

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProyectoPD02

{

///Se encarga de correr todo el programa

///Autor: Emigdio Espinosa Jasso

///Fecha: 12-10-2022

///Versión: 1.0

internal class Program

{

static void Main(string[] args)

{

string opc = "";

while(opc != "3")

{

Console.WriteLine("--------------------------------LOMITOS F.C----------------------------------------");

Console.WriteLine("Bienvenido al programa de control de los centros de alto rendimiento de LOMITOS FC ");

Console.WriteLine("Qué le gustaria ver? 1.Control de Centros de Alto rendimiento, 2.Control de alumnos, 3.Salir");

opc = Console.ReadLine();

//Componente

//1.Control de centros de alto rendimiento

if (opc == "1")

{

IComponente<string> arbol = new CCompuesto<string>("Control Centros de Rendimiento");

IComponente<string> trabajo = arbol;

string opcion = "";

string dato = "";

while (opcion != "6")

{

Console.WriteLine("Estoy en {0}", trabajo.Nombre);

Console.WriteLine("1. Adicionar centro de rendimiento, 2.Adicionar jugador 3. Borrar, 4.Buscar, 5.Mostrar, 6.Salir");

opcion = Console.ReadLine();

Console.WriteLine("----------------");

if (opcion == "1")

{

Console.WriteLine("Dime la ubicacion del centro de rendimiento: ");

dato = Console.ReadLine();

IComponente<string> c = new CCompuesto<string>(dato);

trabajo.Adicionar(c);

trabajo = c;

}

if (opcion == "2")

{

Console.WriteLine("Dime el nombre del jugador ");

dato = Console.ReadLine();

trabajo.Adicionar(new CCompuesto<string>(dato));

}

if (opcion == "3")

{

Console.WriteLine("Dame el elemento a borrar: ");

dato = Console.ReadLine();

trabajo = trabajo.Borrar(dato);

}

if (opcion == "4")

{

Console.WriteLine("Dame el elemento a encontrar: ");

dato = Console.ReadLine();

trabajo = arbol.Buscar(dato);

}

if (opcion == "5")

{

Console.WriteLine(arbol.Mostrar(0));

}

}

}

//Bridge

//Control de pagos

if(opc == "2")

{

while (opc != "5")

{

Console.WriteLine("¿Qué le gustaria ver? 1.Alumnos, 2.Alumnos ordenados por Sucursal. 3.Pago correspondiente. 4.Pago con descuento 5.Salir");

opc = Console.ReadLine();

// Main representa al cliente

Dictionary<string, double> productos = new Dictionary<string, double>();

//Estos son los productos a usar

productos.Add("L-Mateo", 800);

productos.Add("M-Eduardo", 600);

productos.Add("A-Jorge", 500);

productos.Add("A-Luis", 500);

productos.Add("M-Manuel", 600);

productos.Add("L-Victor", 800);

productos.Add("A-Cesar", 500);

productos.Add("M-Sebastian", 600);

productos.Add("A-Esteban", 500);

productos.Add("L-Emilio", 800);

if(opc == "1")

{

//Creamos el bridge

CAbstraccion bridge = new CAbstraccion(1, productos);

bridge.MostrarTotales();

bridge.Listar();

}

if (opc == "2")

{

//Creamos el bridge

CAbstraccion bridge = new CAbstraccion(2, productos);

bridge.MostrarTotales();

bridge.Listar();

}

if (opc == "3")

{

//Creamos el bridge

CAbstraccion bridge = new CAbstraccion(3, productos);

bridge.Listar();

bridge.MostrarTotales();

}

if(opc == "4")

{

//Creamos el bridge

CAbstraccion bridge = new CAbstraccion(4, productos);

bridge.Listar();

bridge.MostrarTotales();

}

}

}

}

}

}

}

**Corridas.**

