

**Universidad ORT Uruguay**  
**Facultad de Ingeniería**  
**Escuela de Tecnología**

**OBLIGATORIO 1 PROGRAMACIÓN 2**  
**DOCUMENTACION**



**Maverick López - 335912**

**M2D**

**Docente: Joaquín Rodríguez**

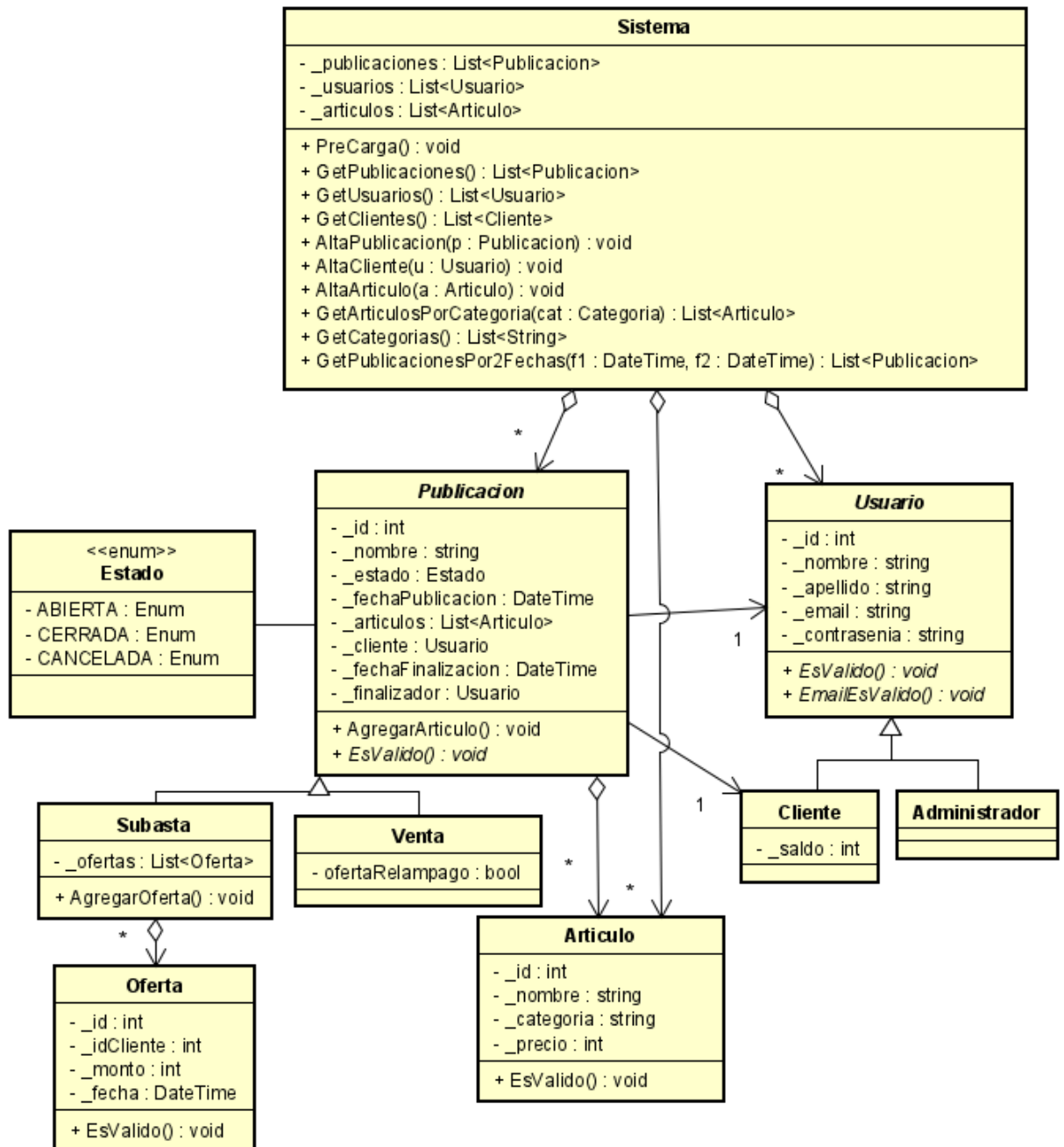
**Analista en Tecnologías de la Información**

**10-10-2024**

# Índice

1. Diagrama de Clases	3
2. Información Precargada	4
3. Código Fuente Comentado	8
4. Evidencia de Consultas Realizadas (IA)	29

# 1. Diagrama de Clases



## 2. Información Precargada

### INFORMACIÓN PRECARGADA CLIENTES:

---

NOMBRE	APELLIDO	EMAIL	CONTRASEÑA	SALDO
Juan	Perez	juanperez@gmail.com	123456	5000
Ana	Gomez	anagomez@yahoo.com	abcdef	3000
Luis	Martinez	luismartinez@hotmail.com	qwerty	7000
Sofia	Lopez	sofia.lopez@gmail.com	pass123	4500
Carlos	Diaz	carlosdiaz@outlook.com	mypassword	2500
Maria	Rodriguez	mrodriguez@gmail.com	1234abcd	6000
Fernando	Hernandez	ferhernandez@gmail.com	ferrari	8000
Lucia	Cruz	lucia.cruz@yahoo.com	letmein	5500
Jorge	Torres	jorge.torres@hotmail.com	password1	3500
Valeria	Morales	valeria.morales@gmail.com	iloveyou	4000

### INFORMACIÓN PRECARGADA ADMINISTRADORES:

---

NOMBRE	APELLIDO	EMAIL	CONTRASEÑA
Luis	Fernández	luis.fernandez@gmail.com	password123
Ana	González	ana.gonzalez@gmail.com	qwerty

### INFORMACIÓN PRECARGADA ARTICULOS:

---

NOMBRE	CATEGORIA	PRECIO
Camiseta	Ropa	500
Pantalones	Ropa	800
Zapatos	Calzado	1200
Chaqueta	Ropa	1500
Gorra	Accesorios	300
Vestido	Ropa	1000
Botas	Calzado	1800
Bufanda	Accesorios	400
Cinturon	Accesorios	600
Camisa	Ropa	700
Sudadera	Ropa	900
Shorts	Ropa	500
Zapatillas	Calzado	1300
Abrigo	Ropa	2000
Leggins	Ropa	750
Sandalias	Calzado	800
Sombrero	Accesorios	350

Chaleco	Ropa	950
Pañuelo	Accesorios	200
Traje	Ropa	2500
Mochila	Accesorios	700
Camiseta de deporte	Ropa	600
Botines	Calzado	1600
Jersey	Ropa	900
Tennis	Calzado	1100
Saco	Ropa	1700
Pantalones cortos	Ropa	400
Calzado deportivo	Calzado	1400
Camison	Ropa	850
Chandal	Ropa	950
Chaqueta de cuero	Ropa	2200
Foulard	Accesorios	300
Bailarinas	Calzado	700
Sudadera con capucha	Ropa	850
Chaqueta impermeable	Ropa	1300
Pantalones de pijama	Ropa	400
Alpargatas	Calzado	600
Corbata	Accesorios	450
Camisa de cuadros	Ropa	800
Zapatos de vestir	Calzado	1500
Camiseta sin mangas	Ropa	550
Cazadora	Ropa	1800
Chanclas	Calzado	300
Bolso de mano	Accesorios	1200
Chaqueta de mezclilla	Ropa	1100
Pantalones de chandal	Ropa	800
Corte clasico	Ropa	900
Sandalias de playa	Calzado	350
Gafas de sol	Accesorios	500
Guantes	Accesorios	400

**INFORMACIÓN PRECARGADA VENTAS:**

NOMBRE	ESTADO	FECHA DE PUBLICACION	CLIENTE	FECHA DE FINALIZACION	ID FINALIZADOR	OFERTA RELAMP
Verano en la playa	ABIERTA	Hoy	null	null	null	True
Escapada de invierno	ABIERTA	Hoy - 5 dias	null	null	null	False
Aventura en la montaña	ABIERTA	Hoy + 2 dias	null	null	null	True
Curso de cocina	ABIERTA	Hoy	null	null	null	True
Concierto de rock	ABIERTA	Hoy - 1 dia	null	null	null	True
Exposicion de arte	ABIERTA	Hoy - 10 dias	null	null	null	False
Taller de fotografia	ABIERTA	Hoy + 3 dias	null	null	null	True
Maraton de peliculas	ABIERTA	Hoy	null	null	null	True
Feria de libros	ABIERTA	Hoy - 15 dias	null	null	null	False
Festival de musica	ABIERTA	Hoy + 1 dia	null	null	null	True

**INFORMACIÓN PRECARGADA OFERTAS:**

Cliente	Monto	Fecha
Usuario1	150	Hoy
Usuario9	2000	Hoy – 1 dia
Usuario8	500	Hoy + 5 dias
Usuario2	300	Hoy – 8 dias
Usuario8	100	Hoy + 3 dias

**INFORMACIÓN PRECARGADA SUBASTAS:**

---

NOMBRE	ESTADO	FECHA DE PUBLICACION	CLIENTE	FECHA DE FINALIZACION	ID FINALIZADOR
Festival de musica	ABIERTA	Hoy + 1 día	null	null	null
Concierto de Rock	ABIERTA	Hoy + 3 días	null	null	null
Exposicion de Arte	ABIERTA	Hoy + 5 días	null	null	null
Torneo de Videojuegos	ABIERTA	Hoy + 2 días	null	null	null
Feria Gastronomica	ABIERTA	Hoy + 4 días	null	null	null
Maraton de Cine	ABIERTA	Hoy + 1 día	null	null	null
Conferencia de Tecnologia	ABIERTA	Hoy + 6 días	null	null	null
Festival de Danza	ABIERTA	Hoy + 2 días	null	null	null
Competencia de Cocina	ABIERTA	Hoy + 3 días	null	null	null
Feria de Libros	ABIERTA	Hoy + 5 días	null	null	null

**INFORMACIÓN PRECARGADA EXTRA:**

---

La publicacion 1 cuenta con 2 articulos: Artículo 1 y 50  
La publicacion 11 cuenta con 2 articulos: Artículo 10 y 15  
La publicacion 13 cuenta con 2 articulos: Artículo 29 y 14  
La publicacion 19 cuenta con 2 articulos: Artículo 43 y 38  
La publicacion 11 cuenta con 2 ofertas: Oferta 1 y 5  
La publicacion 13 cuenta con 2 ofertas: Oferta 3 y 2  
La publicacion 19 cuenta con 1 oferta: Oferta 4

### 3. Código Fuente Comentado

#### Program:

```
using Dominio;

namespace Obligatorio
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Sistema s = new Sistema();

            // --MENU COMPLETO--
            int op = -1;

            while (op != 0)
            {
                Console.Clear();
                Console.WriteLine("1 - Listar Clientes");
                Console.WriteLine("2 - Listar Articulos Filtrados con Categoria");
                Console.WriteLine("3 - Alta Articulo");
                Console.WriteLine("4 - Listar publicaciones entre x y x fecha");
                Console.WriteLine("0 - Salir");

                op = int.Parse(Console.ReadLine());

                // --LISTAR TODOS LOS CLIENTES--
                if (op == 1)
                {
                    // --Valida que no este vacia la lista de clientes y muestra mensaje--
                    if (s.GetClientes().Count == 0)
                    {
                        Console.WriteLine("No hay Clientes Registrados");
                    }
                    else
                    {
                        foreach (Cliente c in s.GetClientes())
                        {
                            Console.WriteLine("Nombre: " + c.Nombre + " - Apellido: " +
                                c.Apellido + " - Email: " + c.Email + " - Saldo: " + c.Saldo);
                        }
                    }
                }
                // --LISTAR ARTICULOS POR CATEGORIA INGRESADA--
            }
        }
    }
}
```



```

else if (op == 2)
{
    // --Muestra todas las categorias disponibles--
    Console.WriteLine("Categorias Disponibles: ");
    foreach (string item in s.GetCategorias())
    {
        Console.WriteLine(item);
    }

    // --Solicita categoria--
    Console.WriteLine("Ingrese categoria");
    string cat = Console.ReadLine();

    // --Valida la categoria y muestra mensaje--
    if (!s.GetCategorias().Contains(cat))
    {
        Console.WriteLine("La Categoria ingresada no existe");
    }
    else
    {
        foreach (Articulo a in s.GetArticulosPorCategoria(cat))
        {
            Console.WriteLine("Nombre: " + a.Nombre + " - Categoria: " +
a.Categoria + " - Precio: " + a.Precio);
        }
    }
}
// --ALTA DE ARTICULO--
else if (op == 3)
{
    // --Solicita datos, valida y muestra mensaje--
    try
    {
        Console.WriteLine("Nombre:");
        string nom = Console.ReadLine();

        Console.WriteLine("Categoria:");
        string cat = Console.ReadLine();

        Console.WriteLine("Precio:");
        int pre = int.Parse(Console.ReadLine());

        Articulo a = new Articulo(nom, cat, pre);

        s.AltaArticulo(a);
        Console.WriteLine("Alta correcta");
    }
}

```

```

        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }
    }
    // --LISTAR PUBLICACIONES DADAS 2 FECHAS--
    else if (op == 4)
    {
        // --Solicita 2 fechas, valida y muestra mensaje--
        try
        {
            Console.WriteLine("Fecha 1:");
            DateTime f1 = DateTime.Parse(Console.ReadLine());

            Console.WriteLine("Fecha 2:");
            DateTime f2 = DateTime.Parse(Console.ReadLine());

            // --Valida que la lista no este vacia y muestra mensaje--
            if (s.GetPublicacionesPor2Fechas(f1, f2).Count == 0)
            {
                Console.WriteLine("No hay publicaciones publicadas entre esas
fechas");
            }
            else
            {
                foreach (Publicacion p in s.GetPublicacionesPor2Fechas(f1, f2))
                {
                    Console.WriteLine("ID:" + p.Id + ", Nombre: " + p.Nombre + ",
Estado: " + p.Estado + ", Fecha de Publicacion: " + p.FechaPublicacion);
                }
            }
        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }
    }

    Console.ReadKey();
}
}
}
}

```

## Sistema:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Dominio
{
    public class Sistema
    {
        // --Listas--
        List<Usuario> _usuarios = new List<Usuario>();
        List<Publicacion> _publicaciones = new List<Publicacion>();
        List<Articulo> _articulos = new List<Articulo>();

        public Sistema()
        {
            Precarga();
        }

        // --Precarga de Datos--
        public void Precarga()
        {
            // --Creacion de Clientes--
            Usuario usuario1 = new Cliente("Juan", "Perez", "juanperez@gmail.com", "123456", 5000);
            Usuario usuario2 = new Cliente("Ana", "Gomez", "anagomez@yahoo.com", "abcdef", 3000);
            Usuario usuario3 = new Cliente("Luis", "Martinez", "luismartinez@hotmail.com", "qwerty", 7000);
            Usuario usuario4 = new Cliente("Sofia", "Lopez", "sofia.lopez@gmail.com", "pass123", 4500);
            Usuario usuario5 = new Cliente("Carlos", "Diaz", "carlosdiaz@outlook.com", "mypassword", 2500);
            Usuario usuario6 = new Cliente("Maria", "Rodriguez", "mrodriguez@gmail.com", "1234abcd", 6000);
            Usuario usuario7 = new Cliente("Fernando", "Hernandez", "ferhernandez@gmail.com", "ferrari", 8000);
            Usuario usuario8 = new Cliente("Lucia", "Cruz", "lucia.cruz@yahoo.com", "letmein", 5500);
            Usuario usuario9 = new Cliente("Jorge", "Torres", "jorge.torres@hotmail.com", "password1", 3500);
            Usuario usuario10 = new Cliente("Valeria", "Morales", "valeria.morales@gmail.com", "iloveyou", 4000);

            // --Alta de los Clientes--
            AltaUsuario(usuario1);
            AltaUsuario(usuario2);
            AltaUsuario(usuario3);
            AltaUsuario(usuario4);
            AltaUsuario(usuario5);
        }
    }
}
```

```

AltaUsuario(usuario6);
AltaUsuario(usuario7);
AltaUsuario(usuario8);
AltaUsuario(usuario9);
AltaUsuario(usuario10);

// --Creacion de los Administradores
Usuario administrador1 = new Administrador("Luis", "Fernández",
"luis.fernandez@gmail.com", "password123");
Usuario administrador2 = new Administrador("Ana", "González",
"ana.gonzalez@gmail.com", "qwerty");

// --Alta de los Administradores
AltaUsuario(administrador1);
AltaUsuario(administrador2);

// --Creacion de los Articulos--
Articulo art1 = new Articulo("Camiseta", "Ropa", 500);
Articulo art2 = new Articulo("Pantalones", "Ropa", 800);
Articulo art3 = new Articulo("Zapatos", "Calzado", 1200);
Articulo art4 = new Articulo("Chaqueta", "Ropa", 1500);
Articulo art5 = new Articulo("Gorra", "Accesorios", 300);
Articulo art6 = new Articulo("Vestido", "Ropa", 1000);
Articulo art7 = new Articulo("Botas", "Calzado", 1800);
Articulo art8 = new Articulo("Bufanda", "Accesorios", 400);
Articulo art9 = new Articulo("Cinturón", "Accesorios", 600);
Articulo art10 = new Articulo("Camisa", "Ropa", 700);
Articulo art11 = new Articulo("Sudadera", "Ropa", 900);
Articulo art12 = new Articulo("Shorts", "Ropa", 500);
Articulo art13 = new Articulo("Zapatillas", "Calzado", 1300);
Articulo art14 = new Articulo("Abrigo", "Ropa", 2000);
Articulo art15 = new Articulo("Leggings", "Ropa", 750);
Articulo art16 = new Articulo("Sandalias", "Calzado", 800);
Articulo art17 = new Articulo("Sombrero", "Accesorios", 350);
Articulo art18 = new Articulo("Chaleco", "Ropa", 950);
Articulo art19 = new Articulo("Pañuelo", "Accesorios", 200);
Articulo art20 = new Articulo("Traje", "Ropa", 2500);
Articulo art21 = new Articulo("Mochila", "Accesorios", 700);
Articulo art22 = new Articulo("Camiseta de deporte", "Ropa", 600);
Articulo art23 = new Articulo("Botines", "Calzado", 1600);
Articulo art24 = new Articulo("Jersey", "Ropa", 900);
Articulo art25 = new Articulo("Tennis", "Calzado", 1100);
Articulo art26 = new Articulo("Saco", "Ropa", 1700);
Articulo art27 = new Articulo("Pantalones cortos", "Ropa", 400);
Articulo art28 = new Articulo("Calzado deportivo", "Calzado", 1400);
Articulo art29 = new Articulo("Camisón", "Ropa", 850);
Articulo art30 = new Articulo("Chándal", "Ropa", 950);
Articulo art31 = new Articulo("Chaqueta de cuero", "Ropa", 2200);
Articulo art32 = new Articulo("Foulard", "Accesorios", 300);
Articulo art33 = new Articulo("Bailarinas", "Calzado", 700);
Articulo art34 = new Articulo("Sudadera con capucha", "Ropa", 850);
Articulo art35 = new Articulo("Chaqueta impermeable", "Ropa", 1300);

```

```

Articulo art36 = new Articulo("Pantalones de pijama", "Ropa", 400);
Articulo art37 = new Articulo("Alpargatas", "Calzado", 600);
Articulo art38 = new Articulo("Corbata", "Accesorios", 450);
Articulo art39 = new Articulo("Camisa de cuadros", "Ropa", 800);
Articulo art40 = new Articulo("Zapatos de vestir", "Calzado", 1500);
Articulo art41 = new Articulo("Camiseta sin mangas", "Ropa", 550);
Articulo art42 = new Articulo("Cazadora", "Ropa", 1800);
Articulo art43 = new Articulo("Chancletas", "Calzado", 300);
Articulo art44 = new Articulo("Bolso de mano", "Accesorios", 1200);
Articulo art45 = new Articulo("Chaqueta de mezclilla", "Ropa", 1100);
Articulo art46 = new Articulo("Pantalones de chándal", "Ropa", 800);
Articulo art47 = new Articulo("Corte clásico", "Ropa", 900);
Articulo art48 = new Articulo("Sandalias de playa", "Calzado", 350);
Articulo art49 = new Articulo("Gafas de sol", "Accesorios", 500);
Articulo art50 = new Articulo("Guantes", "Accesorios", 400);

```

```
// --Alta de los Articulos--
```

```

AltaArticulo(art1);
AltaArticulo(art2);
AltaArticulo(art3);
AltaArticulo(art4);
AltaArticulo(art5);
AltaArticulo(art6);
AltaArticulo(art7);
AltaArticulo(art8);
AltaArticulo(art9);
AltaArticulo(art10);
AltaArticulo(art11);
AltaArticulo(art12);
AltaArticulo(art13);
AltaArticulo(art14);
AltaArticulo(art15);
AltaArticulo(art16);
AltaArticulo(art17);
AltaArticulo(art18);
AltaArticulo(art19);
AltaArticulo(art20);
AltaArticulo(art21);
AltaArticulo(art22);
AltaArticulo(art23);
AltaArticulo(art24);
AltaArticulo(art25);
AltaArticulo(art26);
AltaArticulo(art27);
AltaArticulo(art28);
AltaArticulo(art29);
AltaArticulo(art30);
AltaArticulo(art31);
AltaArticulo(art32);
AltaArticulo(art33);
AltaArticulo(art34);
AltaArticulo(art35);

```

```

AltaArticulo(art36);
AltaArticulo(art37);
AltaArticulo(art38);
AltaArticulo(art39);
AltaArticulo(art40);
AltaArticulo(art41);
AltaArticulo(art42);
AltaArticulo(art43);
AltaArticulo(art44);
AltaArticulo(art45);
AltaArticulo(art46);
AltaArticulo(art47);
AltaArticulo(art48);
AltaArticulo(art49);
AltaArticulo(art50);

// --Creacion de las Ventas--
Publicacion publicacion1 = new Venta("Verano en la playa", Estado.ABIERTA,
DateTime.Now, null, null, null, true);
Publicacion publicacion2 = new Venta("Escapada de invierno", Estado.ABIERTA,
DateTime.Now.AddDays(-5), null, null, null, false);
Publicacion publicacion3 = new Venta("Aventura en la montaña", Estado.ABIERTA,
DateTime.Now.AddDays(2), null, null, null, true);
Publicacion publicacion4 = new Venta("Curso de cocina", Estado.ABIERTA,
DateTime.Now, null, null, null, true);
Publicacion publicacion5 = new Venta("Concierto de rock", Estado.ABIERTA,
DateTime.Now.AddDays(-1), null, null, null, true);
Publicacion publicacion6 = new Venta("Exposición de arte", Estado.ABIERTA,
DateTime.Now.AddDays(-10), null, null, null, false);
Publicacion publicacion7 = new Venta("Taller de fotografía", Estado.ABIERTA,
DateTime.Now.AddDays(3), null, null, null, true);
Publicacion publicacion8 = new Venta("Maratón de películas", Estado.ABIERTA,
DateTime.Now, null, null, null, true);
Publicacion publicacion9 = new Venta("Feria de libros", Estado.ABIERTA,
DateTime.Now.AddDays(-15), null, null, null, false);
Publicacion publicacion10 = new Venta("Festival de música", Estado.ABIERTA,
DateTime.Now.AddDays(1), null, null, null, true);

// --Creacion de las Ofertas--
Oferta oferta1 = new Oferta(usuario1, 150, DateTime.Now);
Oferta oferta2 = new Oferta(usuario9, 2000, DateTime.Now.AddDays(-1));
Oferta oferta3 = new Oferta(usuario8, 500, DateTime.Now.AddDays(5));
Oferta oferta4 = new Oferta(usuario2, 300, DateTime.Now.AddDays(-8));
Oferta oferta5 = new Oferta(usuario8, 100, DateTime.Now.AddDays(3));

// --Creacion de las Subastas--
Subasta publicacion11 = new Subasta("Festival de música", Estado.ABIERTA,
DateTime.Now.AddDays(1), null, null, null);
Subasta publicacion12 = new Subasta("Concierto de Rock", Estado.ABIERTA,
DateTime.Now.AddDays(3), null, null, null);
Subasta publicacion13 = new Subasta("Exposición de Arte", Estado.ABIERTA,
DateTime.Now.AddDays(5), null, null, null);

```

```

Subasta publicacion14 = new Subasta("Torneo de Videojuegos", Estado.ABIERTA,
DateTime.Now.AddDays(2), null, null, null);
Subasta publicacion15 = new Subasta("Feria Gastronómica", Estado.ABIERTA,
DateTime.Now.AddDays(4), null, null, null);
Subasta publicacion16 = new Subasta("Maratón de Cine", Estado.ABIERTA,
DateTime.Now.AddDays(1), null, null, null);
Subasta publicacion17 = new Subasta("Conferencia de Tecnología", Estado.ABIERTA,
DateTime.Now.AddDays(6), null, null, null);
Subasta publicacion18 = new Subasta("Festival de Danza", Estado.ABIERTA,
DateTime.Now.AddDays(2), null, null, null);
Subasta publicacion19 = new Subasta("Competencia de Cocina", Estado.ABIERTA,
DateTime.Now.AddDays(3), null, null, null);
Subasta publicacion20 = new Subasta("Feria de Libros", Estado.ABIERTA,
DateTime.Now.AddDays(5), null, null, null);

```

```

// --Agregar Articulos a las Publicaciones--

```

```

publicacion1.AgregarArticulo(art1);
publicacion1.AgregarArticulo(art50);
publicacion11.AgregarArticulo(art10);
publicacion11.AgregarArticulo(art15);
publicacion13.AgregarArticulo(art29);
publicacion13.AgregarArticulo(art14);
publicacion19.AgregarArticulo(art43);
publicacion19.AgregarArticulo(art38);

```

```

// --Agregar Ofertas a las Subastas--

```

```

publicacion11.AgregarOferta(oferta1);
publicacion11.AgregarOferta(oferta5);
publicacion13.AgregarOferta(oferta3);
publicacion13.AgregarOferta(oferta2);
publicacion19.AgregarOferta(oferta4);

```

```

// --Alta de las Publicaciones--

```

```

AltaPublicacion(publicacion1);
AltaPublicacion(publicacion2);
AltaPublicacion(publicacion3);
AltaPublicacion(publicacion4);
AltaPublicacion(publicacion5);
AltaPublicacion(publicacion6);
AltaPublicacion(publicacion7);
AltaPublicacion(publicacion8);
AltaPublicacion(publicacion9);
AltaPublicacion(publicacion10);
AltaPublicacion(publicacion11);
AltaPublicacion(publicacion12);
AltaPublicacion(publicacion13);
AltaPublicacion(publicacion14);
AltaPublicacion(publicacion15);
AltaPublicacion(publicacion16);
AltaPublicacion(publicacion17);
AltaPublicacion(publicacion18);
AltaPublicacion(publicacion19);

```

```

AltaPublicacion(publicacion20);
}

// --Get de todas las Publicaciones--
public List<Publicacion> GetPublicaciones()
{
    return _publicaciones;
}

// --Get de todos los Usuarios--
public List<Usuario> GetUsuarios()
{
    return _usuarios;
}

// --Get de todos los Clientes--
public List<Cliente> GetClientes()
{
    List<Cliente> _clientes = new List<Cliente>();

    foreach (Usuario u in _usuarios)
    {
        if (u is Cliente c)
        {
            _clientes.Add(c);
        }
    }
    return _clientes;
}

// --Alta de la Publicacion--
public void AltaPublicacion(Publicacion p)
{
    p.EsValido();
    _publicaciones.Add(p);
}

// --Alta del Usuario--
public void AltaUsuario(Usuario u)
{
    u.EsValido();
    _usuarios.Add(u);
}

// --Alta del Artículo--
public void AltaArticulo(Articulo a)
{
    try
    {
        a.EsValido();
        _articulos.Add(a);
    }
}

```



```

catch (Exception e)
{
throw e;
}

}

// --Get de todas las Categorías--
public List<string> GetCategorias()
{
List<string> ret = new List<string>();
foreach (Articulo a in _articulos)
{
if (!ret.Contains(a.Categoria))
{
ret.Add(a.Categoria);
}
}
return ret;
}

// --Get de los Artículos filtrado por Categoría ingresada--
public List<Articulo> GetArticulosPorCategoria(string cat)
{
List<Articulo> ret = new List<Articulo>();

foreach (Articulo a in _articulos)
{
if (a.Categoria == cat)
{
ret.Add(a);
}
}
return ret;
}

// --Get de las Publicaciones por 2 Fechas ingresadas--
public List<Publicacion> GetPublicacionesPor2Fechas(DateTime f1, DateTime f2)
{
List<Publicacion> ret = new List<Publicacion>();

foreach (Publicacion p in GetPublicaciones())
{
if (p.FechaPublicacion >= f1 && p.FechaPublicacion <= f2)
{
ret.Add(p);
}
}
return ret;
}
}
}
}

```

## Usuario:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Dominio
{
    public abstract class Usuario
    {
        private static int _ultimoId = 1;
        public int Id { get; set; }
        public string Nombre { get; set; }
        public string Apellido { get; set; }
        public string Email { get; set; }
        public string Contraseña { get; set; }

        protected Usuario(string nombre, string apellido, string email, string contraseña)
        {
            Id = _ultimoId++;
            Nombre = nombre;
            Apellido = apellido;
            Email = email;
            Contraseña = contraseña;
        }
        protected Usuario()
        {
            Id = _ultimoId++;
        }

        // --Validacion
        public abstract void EsValido();

        // --Validacion del Email
        protected virtual bool EmailEsValido()
        {
            bool cumple = true;

            int posArroba = Email.IndexOf("@");

            if (posArroba != -1)
            {
                if (posArroba == 0)
                {
                    cumple = false;
                }
            }
        }
    }
}
```

```
    }  
    if (posArroba == Email.Length - 1)  
    {  
        cumple = false;  
    }  
}  
else  
{  
    cumple = false;  
}  
return cumple;  
}  
  
}  
}
```

## Cliente:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Dominio
{
    public class Cliente : Usuario
    {
        public int Saldo { get; set; }
        public Cliente(string nombre, string apellido, string email, string contrasenia, int saldo) : base(nombre, apellido, email, contrasenia)
        {
            Saldo = saldo;
        }

        // --Validacion
        public override void EsValido()
        {
            if (Nombre.Length < 2)
            {
                throw new Exception("El nombre no puede contener menos de 2 caracteres");
            }
            if (Apellido.Length < 2)
            {
                throw new Exception("El apellido no puede contener menos de 2 caracteres");
            }
            if (EmailEsValido() == false)
            {
                throw new Exception("El email no es valido");
            }
            if (Saldo < 0)
            {
                throw new Exception("El saldo no puede ser menor a 0");
            }
        }
    }
}
```

## Administrador:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Dominio
{
    public class Administrador : Usuario
    {
        public Administrador(string nombre, string apellido, string email, string
contrasenia) : base(nombre, apellido, email, contrasenia)
        {
        }

        // --Validacion
        public override void EsValido()
        {
            if (Nombre.Length < 2)
            {
                throw new Exception("El nombre no puede contener menos de 2 caracteres");
            }
            if (Apellido.Length < 2)
            {
                throw new Exception("El apellido no puede contener menos de 2 caracteres");
            }
            if (EmailEsValido() == false)
            {
                throw new Exception("El email no es valido");
            }
        }
    }
}
```

## Publicacion:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Dominio
{
    public abstract class Publicacion
    {
        private static int _ultimoId = 1;

        public List<Articulo> _articulos = new List<Articulo>();

        public int Id { get; set; }
        public string Nombre { get; set; }
        public Estado Estado { get; set; }
        public DateTime FechaPublicacion { get; set; }
        public Usuario? Cliente { get; set; }
        public DateTime? FechaFinalizacion { get; set; }
        public Usuario? Finalizador { get; set; }

        protected Publicacion(string nombre, Estado estado, DateTime fechaPublicacion,
            Usuario? cliente, DateTime? fechaFinalizacion, Usuario? finalizador)
        {
            Id = _ultimoId++;
            Nombre = nombre;
            Estado = estado;
            FechaPublicacion = fechaPublicacion;
            Cliente = cliente;
            FechaFinalizacion = fechaFinalizacion;
            Finalizador = finalizador;
        }

        protected Publicacion()
        {
            Id = _ultimoId++;
        }

        // --Alta de Articulo a la Publicacion
        public void AgregarArticulo(Articulo a)
        {
            _articulos.Add(a);
        }
    }
}
```

```
// --Validacion
public abstract void EsValido();
}
}
```

## Venta:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Dominio
{
    public class Venta : Publicacion
    {
        public bool OfertaRelampago { get; set; }
        public Venta(string nombre, Estado estado, DateTime fechaPublicacion, Usuario?
cliente, DateTime? fechaFinalizacion, Usuario? finalizador, bool ofertarelampago) :
base(nombre, estado, fechaPublicacion, cliente, fechaFinalizacion, finalizador)
        {
            OfertaRelampago=ofertarelampago;
        }

        // --Validacion
        public override void EsValido()
        {
            if (Nombre.Length<2)
            {
                throw new Exception("El nombre no puede tener menos de 2 caracteres");
            }
        }
    }
}
```



## Subasta:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Dominio
{
    public class Subasta : Publicacion
    {
        List<Oferta> _ofertas = new List<Oferta>();

        public Subasta(string nombre, Estado estado, DateTime fechaPublicacion, Usuario?
cliente, DateTime? fechaFinalizacion, Usuario? finalizador) : base(nombre, estado,
fechaPublicacion, cliente, fechaFinalizacion, finalizador)
        {
        }

        // --Alta de Oferta a la Subasta
        public void AgregarOferta(Oferta o)
        {
            _ofertas.Add(o);
        }

        // --Validacion
        public override void EsValido()
        {
            if (Nombre.Length < 2)
            {
                throw new Exception("El nombre no puede tener menos de 2 caracteres");
            }
        }
    }
}
```

## Oferta:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Dominio
{
    public class Oferta
    {
        private static int _ultimoId = 1;
        public int Id { get; set; }
        public Usuario Cliente { get; set; }
        public int Monto { get; set; }
        public DateTime Fecha { get; set; }

        public Oferta(Usuario cliente, int monto, DateTime fecha)
        {
            Id = _ultimoId++;
            Cliente = cliente;
            Monto = monto;
            Fecha = fecha;
        }
        public Oferta()
        {
            Id = _ultimoId++;
        }

        // --Validacion
        public void EsValido()
        {
            if (Cliente.Id < 1)
            {
                throw new Exception("El cliente no existe");
            }
            if (Monto < 1)
            {
                throw new Exception("El monto no puede ser menor a 1");
            }
        }
    }
}
```

## Articulo:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;

namespace Dominio
{
    public class Articulo
    {
        private static int _ultimoId = 1;
        public int Id { get; set; }
        public string Nombre { get; set; }
        public string Categoria { get; set; }
        public int Precio { get; set; }

        public Articulo(string nombre,string categoria, int precio)
        {
            Id = _ultimoId++;
            Nombre = nombre;
            Categoria = categoria;
            Precio = precio;
        }

        public Articulo()
        {
            Id= _ultimoId++;
        }

        // --Validacion
        public void EsValido()
        {
            if (Nombre.Length < 3)
            {
                throw new Exception("El Nombre no debe contener menos de 3 caracteres");
            }
            if (Precio < 0)
            {
                throw new Exception("El Precio no puede ser menor de 0");
            }
        }
    }
}
```

## Estado:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace Dominio  
{  
    public enum Estado  
    {  
        ABIERTA ,  
        CERRADA,  
        CANCELADA  
    }  
}
```

## 4. Evidencia de Consultas Realizadas (IA)

Gran parte de los datos otorgados por la IA fueron modificados a mano.

Ventas:

Tú dijiste:

Necesito datos de prueba

ChatGPT

Claro, ¿qué tipo de datos de prueba necesitas? Puedo ayudarte con ejemplos de datos, estructuras de datos o cualquier otro tipo de información que necesites.

Tú dijiste:

necesito 10 de este estilo

ChatGPT

¡Perfecto! Pero necesito un poco más de información sobre el estilo que buscas. ¿Quieres datos de prueba para una base de datos, ejemplos de texto, números, o algo específico?

Tú dijiste:

```
Publicacion publicacion1 = new Venta("Verano en la playa", Estado.ABIERTA,
DateTime.Now, _articulos1, null, null, null, true);
```

ChatGPT

Entendido, aquí tienes 10 ejemplos de publicaciones en un formato similar:

```
Publicacion publicacion1 = new Venta("Verano en la playa", Estado.ABIERTA,
DateTime.Now, _articulos1, null, null, null, true);
```

```
Publicacion publicacion2 = new Venta("Escapada de invierno", Estado.CERRADA,
DateTime.Now.AddDays(-5), _articulos2, null, null, null, false);
```

```
Publicacion publicacion3 = new Venta("Aventura en la montaña", Estado.ABIERTA,
DateTime.Now.AddDays(2), _articulos3, null, null, null, true);
```

```
Publicacion publicacion4 = new Venta("Curso de cocina", Estado.ABIERTA,
DateTime.Now, _articulos4, null, null, null, true);
```

```
Publicacion publicacion5 = new Venta("Concierto de rock", Estado.ABIERTA,
DateTime.Now.AddDays(-1), _articulos5, null, null, null, true);
```

```
Publicacion publicacion6 = new Venta("Exposición de arte", Estado.CERRADA,
DateTime.Now.AddDays(-10), _articulos6, null, null, null, false);
```

```
Publicacion publicacion7 = new Venta("Taller de fotografía", Estado.ABIERTA,
DateTime.Now.AddDays(3), _articulos7, null, null, null, true);
```

```
Publicacion publicacion8 = new Venta("Maratón de películas", Estado.ABIERTA,
DateTime.Now, _articulos8, null, null, null, true);
```

```
Publicacion publicacion9 = new Venta("Feria de libros", Estado.CERRADA,
DateTime.Now.AddDays(-15), _articulos9, null, null, null, false);
```

```
Publicacion publicacion10 = new Venta("Festival de música", Estado.ABIERTA,
DateTime.Now.AddDays(1), _articulos10, null, null, null, true);
```

Recuerda ajustar \_articulosX según lo que necesites para cada publicación. Si necesitas más detalles o cambios, házmelo saber.

## Clientes:

necesito datos de prueba

ChatGPT

Claro, ¿qué tipo de datos de prueba necesitas? Pueden ser números, texto, fechas, o algo más específico. Dime un poco más y te ayudo.

Tú dijiste:

necesito 10 de este estilo

ChatGPT

Perfecto, pero necesito un poco más de contexto. ¿Qué tipo de datos estás buscando? ¿Ejemplo de nombres, correos electrónicos, direcciones, números, etc.? ¡Así puedo generarlos para ti!

Tú dijiste:

```
Usuario          usuario1          =          new
Cliente("Juan","Perez","juanperez@gmail.com","123456",5000);
```

ChatGPT

Entendido, necesitas datos de prueba para instancias de la clase Cliente. Aquí tienes 10 ejemplos:

```
Usuario usuario1 = new Cliente("Juan", "Perez", "juanperez@gmail.com", "123456",
5000);
```

```
Usuario usuario2 = new Cliente("Ana", "Gomez", "anagomez@yahoo.com", "abcdef",
3000);
```

```
Usuario usuario3 = new Cliente("Luis", "Martinez", "luismartinez@hotmail.com",
"qwerty", 7000);
```

```
Usuario usuario4 = new Cliente("Sofia", "Lopez", "sofia.lopez@gmail.com", "pass123",
4500);
```

```
Usuario usuario5 = new Cliente("Carlos", "Diaz", "carlosdiaz@outlook.com",  
"mypassword", 2500);
```

```
Usuario usuario6 = new Cliente("Maria", "Rodriguez", "mrodriguez@gmail.com",  
"1234abcd", 6000);
```

```
Usuario usuario7 = new Cliente("Fernando", "Hernandez", "ferhernandez@gmail.com",  
"ferrari", 8000);
```

```
Usuario usuario8 = new Cliente("Lucia", "Cruz", "lucia.cruz@yahoo.com", "letmein",  
5500);
```

```
Usuario usuario9 = new Cliente("Jorge", "Torres", "jorge.torres@hotmail.com",  
"password1", 3500);
```

```
Usuario usuario10 = new Cliente("Valeria", "Morales", "valeria.morales@gmail.com",  
"iloveyou", 4000);
```

Si necesitas modificar alguno de los datos o algún otro tipo de información, házmelo saber.

## Administrador:

necesito 3 datos de prueba de este estilo Usuario administrador1 = new Administrador("Valeria", "Morales", "valeria.morales@gmail.com", "iloveyou");

ChatGPT

Claro, aquí tienes tres datos de prueba en ese formato:

```
Usuario administrador2 = new Administrador("Luis", "Fernández",  
"luis.fernandez@gmail.com", "password123");
```

```
Usuario administrador3 = new Administrador("Ana", "González",  
"ana.gonzalez@gmail.com", "qwerty");
```



```
Usuario administrador4 = new Administrador("Carlos", "Pérez",  
"carlos.perez@gmail.com", "12345678");
```

Si necesitas más ejemplos o alguna modificación, házmelo saber.

## Articulos:

necesito 50 datos de prueba de este estilo

ChatGPT

Claro, puedo ayudarte con eso. ¿Qué tipo de datos de prueba necesitas? Por ejemplo, ¿son nombres, direcciones, números, o algo específico para un contexto particular? Dime más para poder proporcionarte lo que necesitas.

Tú dijiste:

```
Articulo art1 = new Articulo("Camiseta", "Ropa", 500);
```

ChatGPT

¡Perfecto! Aquí tienes 50 datos de prueba en el mismo formato que mencionaste. Cada uno representa un artículo con nombre, categoría y precio:

```
Articulo art1 = new Articulo("Camiseta", "Ropa", 500);
```

```
Articulo art2 = new Articulo("Pantalones", "Ropa", 800);
```

```
Articulo art3 = new Articulo("Zapatos", "Calzado", 1200);
```

```
Articulo art4 = new Articulo("Chaqueta", "Ropa", 1500);
```

```
Articulo art5 = new Articulo("Gorra", "Accesorios", 300);
```

```
Articulo art6 = new Articulo("Vestido", "Ropa", 1000);
```

```
Articulo art7 = new Articulo("Botas", "Calzado", 1800);
```

```
Articulo art8 = new Articulo("Bufanda", "Accesorios", 400);
```

Articulo art9 = new Articulo("Cinturón", "Accesorios", 600);

Articulo art10 = new Articulo("Camisa", "Ropa", 700);

Articulo art11 = new Articulo("Sudadera", "Ropa", 900);

Articulo art12 = new Articulo("Shorts", "Ropa", 500);

Articulo art13 = new Articulo("Zapatillas", "Calzado", 1300);

Articulo art14 = new Articulo("Abrigo", "Ropa", 2000);

Articulo art15 = new Articulo("Leggings", "Ropa", 750);

Articulo art16 = new Articulo("Sandalias", "Calzado", 800);

Articulo art17 = new Articulo("Sombrero", "Accesorios", 350);

Articulo art18 = new Articulo("Chaleco", "Ropa", 950);

Articulo art19 = new Articulo("Pañuelo", "Accesorios", 200);

Articulo art20 = new Articulo("Traje", "Ropa", 2500);

Articulo art21 = new Articulo("Mochila", "Accesorios", 700);

Articulo art22 = new Articulo("Camiseta de deporte", "Ropa", 600);

Articulo art23 = new Articulo("Botines", "Calzado", 1600);

Articulo art24 = new Articulo("Jersey", "Ropa", 900);

Articulo art25 = new Articulo("Tennis", "Calzado", 1100);

Articulo art26 = new Articulo("Saco", "Ropa", 1700);

Articulo art27 = new Articulo("Pantalones cortos", "Ropa", 400);

Articulo art28 = new Articulo("Calzado deportivo", "Calzado", 1400);

Articulo art29 = new Articulo("Camisón", "Ropa", 850);

Articulo art30 = new Articulo("Chándal", "Ropa", 950);

Articulo art31 = new Articulo("Chaqueta de cuero", "Ropa", 2200);

Articulo art32 = new Articulo("Foulard", "Accesorios", 300);

Articulo art33 = new Articulo("Bailarinas", "Calzado", 700);

Articulo art34 = new Articulo("Sudadera con capucha", "Ropa", 850);

Articulo art35 = new Articulo("Chaqueta impermeable", "Ropa", 1300);

Articulo art36 = new Articulo("Pantalones de pijama", "Ropa", 400);

Articulo art37 = new Articulo("Alpargatas", "Calzado", 600);

Articulo art38 = new Articulo("Corbata", "Accesorios", 450);

Articulo art39 = new Articulo("Camisa de cuadros", "Ropa", 800);

Articulo art40 = new Articulo("Zapatos de vestir", "Calzado", 1500);

Articulo art41 = new Articulo("Camiseta sin mangas", "Ropa", 550);

Articulo art42 = new Articulo("Cazadora", "Ropa", 1800);

Articulo art43 = new Articulo("Chancletas", "Calzado", 300);

Articulo art44 = new Articulo("Bolso de mano", "Accesorios", 1200);

Articulo art45 = new Articulo("Chaqueta de mezclilla", "Ropa", 1100);

Articulo art46 = new Articulo("Pantalones de chándal", "Ropa", 800);

Articulo art47 = new Articulo("Corte clásico", "Ropa", 900);

Articulo art48 = new Articulo("Sandalias de playa", "Calzado", 350);

Articulo art49 = new Articulo("Gafas de sol", "Accesorios", 500);

Articulo art50 = new Articulo("Guantes", "Accesorios", 400);

Estos datos incluyen varias categorías de ropa, calzado y accesorios con precios variados. Si necesitas algún ajuste o más información, házmelo saber.

## Publicaciones:

necesito datos de prueba

ChatGPT

Claro, ¿qué tipo de datos de prueba necesitas? ¿Son para una base de datos, para un análisis específico, o algo más? Dime más para poder ayudarte mejor.

Tú dijiste:

necesito 10 de este estilo `Publicacion publicacion1 = new Subasta("Festival de música", Estado.ABIERTA, DateTime.Now.AddDays(1), null, null, null);`

ChatGPT

Aquí tienes 10 ejemplos de publicaciones en el formato que mencionas, utilizando distintos eventos y atributos:

```
Publicacion publicacion1 = new Subasta("Concierto de Rock", Estado.ABIERTA,
DateTime.Now.AddDays(3), null, null, null);
```

```
Publicacion publicacion2 = new Subasta("Exposición de Arte", Estado.ABIERTA,
DateTime.Now.AddDays(5), null, null, null);
```

```
Publicacion publicacion3 = new Subasta("Torneo de Videojuegos", Estado.ABIERTA,
DateTime.Now.AddDays(2), null, null, null);
```

```
Publicacion publicacion4 = new Subasta("Feria Gastronómica", Estado.ABIERTA,
DateTime.Now.AddDays(4), null, null, null);
```

```
Publicacion publicacion5 = new Subasta("Maratón de Cine", Estado.ABIERTA,
DateTime.Now.AddDays(1), null, null, null);
```

```
Publicacion publicacion6 = new Subasta("Conferencia de Tecnología",
Estado.ABIERTA, DateTime.Now.AddDays(6), null, null, null);
```

```
Publicacion publicacion7 = new Subasta("Festival de Danza", Estado.ABIERTA,  
DateTime.Now.AddDays(2), null, null, null);
```

```
Publicacion publicacion8 = new Subasta("Competencia de Cocina", Estado.ABIERTA,  
DateTime.Now.AddDays(3), null, null, null);
```

```
Publicacion publicacion9 = new Subasta("Feria de Libros", Estado.ABIERTA,  
DateTime.Now.AddDays(5), null, null, null);
```

```
Publicacion publicacion10 = new Subasta("Festival de Verano", Estado.ABIERTA,  
DateTime.Now.AddDays(7), null, null, null);
```

Si necesitas hacer alguna modificación o agregar más detalles, ¡dímelo!