

UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

APLICACIONES DISTRIBUIDAS



ESTILOS Y PATRONES ARQUITECTÓNICOS

NRC: 2553

Carrera: Ingeniería de Software

Integrantes:

- Erick Andrade
- Ángel Castillo
- Ricardo Lazo
- Vanessa Zurita

Docente: Ing. Darío Morales

1. Introducción	3
2. Desarrollo.....	3
2.1 Estilos de arquitectura de software.....	3
2.2 Patrones de arquitectura.....	4
2.3 Patrones de lenguaje	5
3. Preguntas.....	6
4. Conclusiones	7
5. Recomendaciones	7
6. Referencias bibliográficas	8

1. Introducción

En este taller se desarrollará un sistema básico de gestión de clientes y productos mediante un servicio web SOAP y operaciones CRUD (Crear, Leer, Actualizar y Eliminar), el propósito de este taller es familiarizar a los participantes con la implementación de un sistema interactivo que permite la manipulación de datos a través de un servicio web utilizando el protocolo SOAP, uno de los métodos más tradicionales y robustos para la comunicación entre sistemas distribuidos.

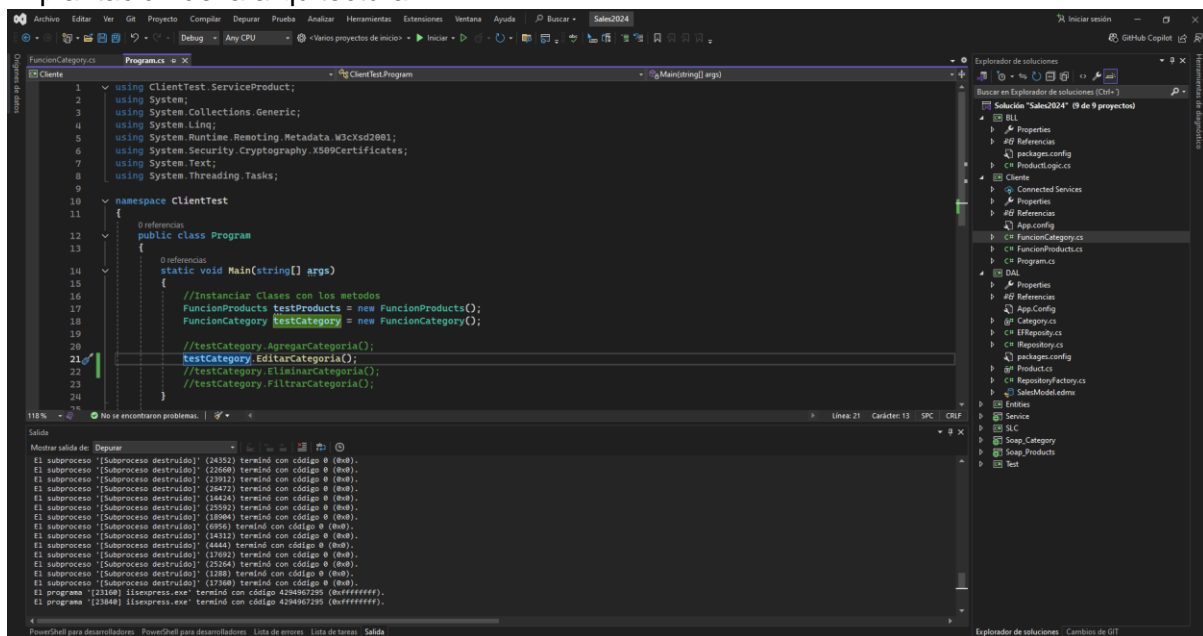
El sistema contará con dos entidades principales: Clientes y Productos. Los clientes podrán tener información personal básica como nombre, dirección y teléfono, mientras que los productos estarán definidos por atributos como nombre, descripción, precio y stock disponible. A través de las operaciones CRUD, los usuarios podrán gestionar ambas entidades de manera eficiente, permitiendo la creación, actualización, consulta y eliminación de datos.

Además, este taller proporcionará la oportunidad de comprender cómo funcionan los servicios web basados en SOAP, incluyendo la construcción y consumo de solicitudes y respuestas en formato XML, así como la integración de estos servicios con un backend que administre la persistencia de datos, en este caso, utilizando una base de datos para almacenar la información.

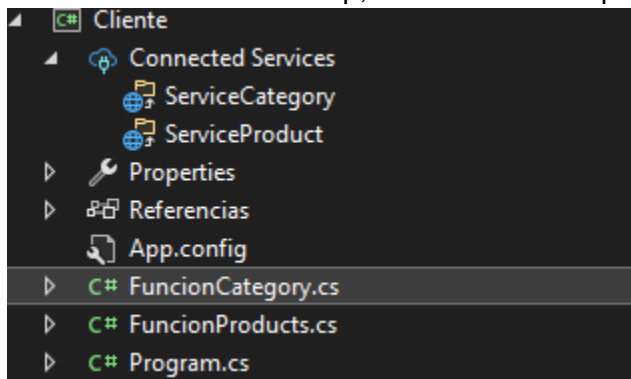
El enfoque de este taller no solo es técnico, sino que también busca proporcionar una comprensión clara sobre los principios fundamentales de la arquitectura de servicios web y cómo los mismos pueden ser aplicados en situaciones reales para mejorar la interoperabilidad entre sistemas distintos.

2. Desarrollo

Implantación de la arquitectura



Se implementaron los proyectos Soap para las dos entidades, se vincularon las bases de datos a cada servicio Soap, se hizo el cliente para consumir las clases con datos quemados



Se distribuyo de la siguiente manera dos archivos para cada entidad y el main en donde llamaremos a las funciones.

Codigo de categorias:

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

```
namespace ClientTest
```

```
{
    public class FuncionCategory
    {
        public void AgregarCategoria()
        {
            Category result = null;
            try
            {
                using (var r = new ServiceCategory.Service1Client())
                {
                    Category category = new Category();
                    category.CategoryName = "Comida";
                    category.Description = "Todo lo que es comida";
                    result = r.GetDataCategory(category);
                }
                Console.WriteLine("Categoria Ingresado con Exito");
            }
            catch (Exception ex)
            {
                Console.WriteLine("Error al ingresar un categoria" + ex);
            }
        }

        public void EditarCategoria()
        {
            bool result = false;
```

```

try
{
    using (var r = new ServiceCategory.Service1Client())
    {
        Category category = new Category();
        category.CategoryID = 7;
        category.CategoryName = "Bebida";
        result = r.UpdateDataCategory(category);
    }
    Console.WriteLine("Categoria Editado con Exito");
}
catch (Exception)
{
}
}

```

```

public void EliminarCategoria()
{
    bool result = false;
    try
    {
        using (var r = new ServiceCategory.Service1Client())
        {
            Category category = new Category();
            category.CategoryID = 4;
            category.CategoryName = "Comida";
            category.Description = "Todo lo que es comida";
            result = r.DeleteDataCategory(category);
        }
        Console.WriteLine("Categoria eliminado con Exito");
    }
    catch (Exception)
    {
        Console.WriteLine("Error al eliminar el categoria");
    }
}

```

```

public static void FiltrarCategoryo()
{
    Category[] result = null;
    try
    {
        using (var r = new ServiceCategory.Service1Client())
        {
            Category category = new Category();
            category.CategoryID = 1;
            category.CategoryName = "CategoriaActualizado";
            result = r.FilterCategory(category);
        }
    }
}

```

```

        Console.WriteLine(result[0]);
    }
    catch (Exception ex)
    {
        Console.WriteLine("No se puede buscar la categoria" + ex);
    }
}
}
}

```

Codigo para productos:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ClientTest.ServiceProduct;

namespace ClientTest
{
    public class FuncionProducts
    {
        public void AgregarProducto()
        {
            Product result = null;
            try
            {
                using (var r = new ServiceProduct.Service1Client())
                {
                    Product product = new Product();
                    product.ProductName = "Producto";
                    product.UnitPrice = 10;
                    product.CategoryID = 1;
                    product.UnitsInStock = 10;
                    result = r.GetData(product);
                }
                Console.WriteLine("Producto Ingresado con Exito");
            }
            catch (Exception ex)
            {
                Console.WriteLine("Error al ingresar un producto" + ex);
            }
        }

        public void EditarProducto()
        {
            bool result = false;
            try
            {

```

```

        using (var r = new ServiceProduct.Service1Client())
        {
            Product product = new Product();
            product.ProductID = 8;
            product.ProductName = "Producto Editado";
            product.UnitPrice = 69;
            product.CategoryID = 1;
            product.UnitsInStock = 96;
            result = r.UpdateData(product);
        }
        Console.WriteLine("Producto Editado con Exito");
    }
    catch (Exception)
    {
    }
}

```

```

public void EliminarProducto()
{
    bool result = false;
    try
    {
        using (var r = new ServiceProduct.Service1Client())
        {
            Product product = new Product();
            product.ProductID = 8;
            product.ProductName = "Producto Editado";
            product.UnitPrice = 99;
            product.CategoryID = 1;
            product.UnitsInStock = 99;
            result = r.DeleteData(product);
        }
        Console.WriteLine("Producto eliminado con Exito");
    }
    catch (Exception)
    {
        Console.WriteLine("Error al eliminar el producto");
    }
}

```

```

public void FiltrarProducto()
{
    Product[] result = null;
    try
    {
        using (var r = new ServiceProduct.Service1Client())
        {
            Product product = new Product();
            product.ProductID = 1;

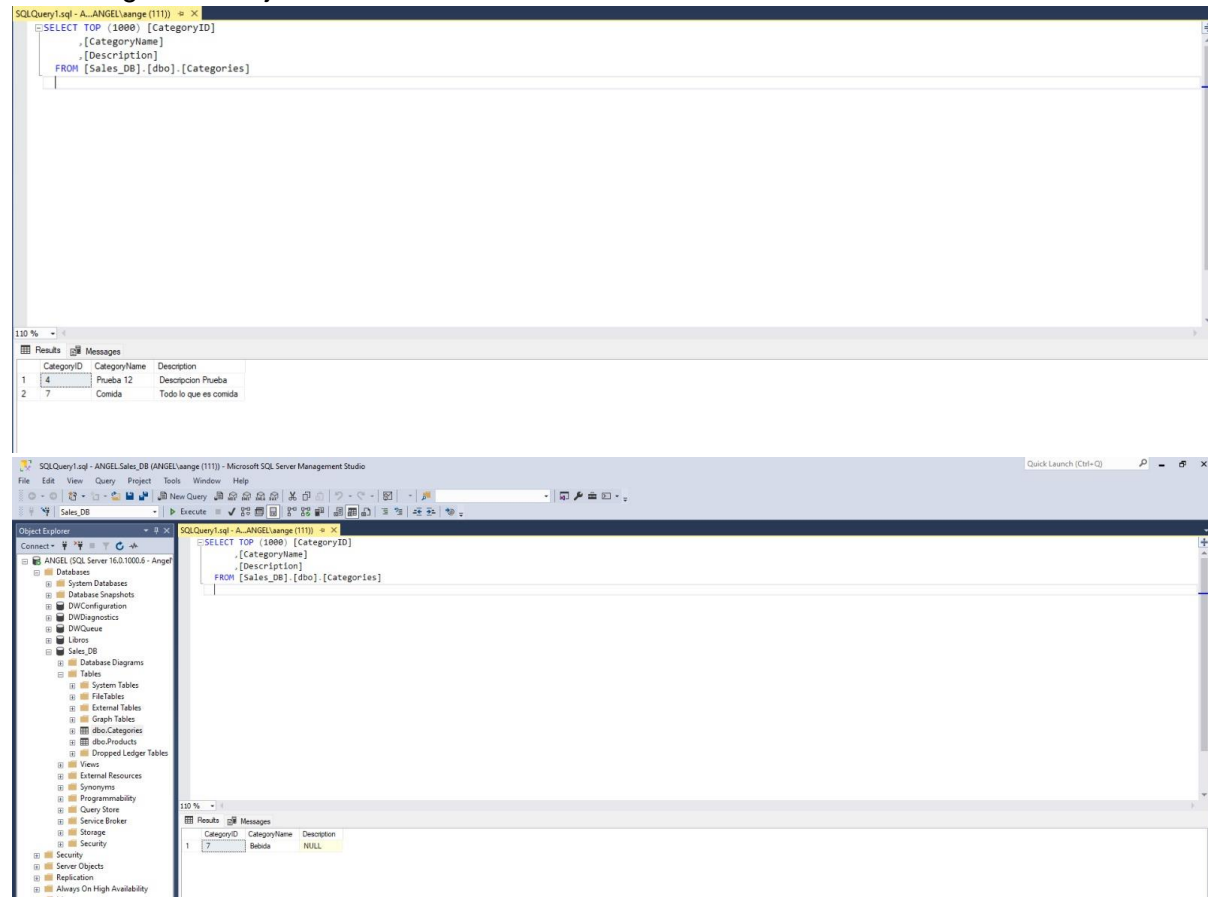
```

```

        product.ProductName = "ProductoActualizado";
        product.UnitPrice = 99;
        product.CategoryID = 1;
        product.UnitsInStock = 99;
        result = r.FilterProduct(product);
    }
    Console.WriteLine(result[0]);
}
catch (Exception ex)
{
    Console.WriteLine("No se puede buscar el producto" + ex);
}
}
}

```

Los códigos de la ejecución son :



Las siguientes imagenes por falta de tiempo no se pueden poner

Git: [vanessazurita/AplicacionesDistribuidas: 2553](https://github.com/vanessazurita/AplicacionesDistribuidas)