

Evidencias Bitácora No 8			
Ficha	Aprendiz	cedula	periodo
2675810	RODNEY ZAPATA PALACIO	72209311	11/01/2025 – 25/01/2025

1. Fecha: 13/01/2025. Controlador de clientes, Adiciona clientes nuevos.

```

75
76 // POST: api/Cientes
77 // To protect from overposting attacks, see https://go.microsoft.com/fwlink/?linkid=2123754
78 [HttpPost]
79 0 referencias
80 public async Task<ActionResult<Cliente>> PostCliente(Cliente cliente)
81 {
82     _context.Cientes.Add(cliente);
83     await _context.SaveChangesAsync();
84     return CreatedAtAction("GetCliente", new { id = cliente.Id }, cliente);
85 }
86

```

2. 15/01/2025 Controlador de clientes, Eliminación de clientes.

```

87 // DELETE: api/Cientes/5
88 [HttpDelete("{id}")]
89 0 referencias
90 public async Task<IActionResult> DeleteCliente(int id)
91 {
92     var cliente = await _context.Cientes.FindAsync(id);
93     if (cliente == null)
94     {
95         return NotFound();
96     }
97     _context.Cientes.Remove(cliente);
98     await _context.SaveChangesAsync();
99     return NoContent();
100 }
101
102 1 referencia
103 private bool ClienteExists(int id)
104 {
105     return _context.Cientes.Any(e => e.Id == id);
106 }
107
108 }

```

3. 20/01/2025 al 21/01/2025. EndPoint que ejecuta la consulta de un procedimiento almacenado, que se encarga de trae los datos de la factura con sus respectiva credenciales y cuerpo de la factura en formato XML, para luego ser transmitido a la plataforma de facture.

Evidencias Bitácora No 8			
Ficha	Aprendiz	cedula	periodo
2675810	RODNEY ZAPATA PALACIO	72209311	11/01/2025 – 25/01/2025

```

WebApiServi
1  using Microsoft.EntityFrameworkCore;
2  using System.ComponentModel.DataAnnotations.Schema;
3  using System.ComponentModel.DataAnnotations;
4
5  namespace WebApiServi.Models
6  {
7      4 referencias
      public class FacturaElectronica
8      {
9          [Key] // Define 'numfac' como clave primaria
10         [Column(TypeName = "varchar(10)")] // Ajusta el tamaño según lo necesites
11         public required string numfac { get; set; }
12         public required string tipo { get; set; }
13         public required string token { get; set; }
14         public required string fe_xml_factura { get; set; }
15         public required string keycontrol { get; set; }
16         public required string x_Who { get; set; }
17         public required string url { get; set; }
18     }
19 }
20
21

```

4. 22/01/2025 al 23/01/ Configuración de CORS (Cross-Origin Resource Sharing) es un mecanismo de seguridad en los navegadores que controla qué servidores pueden acceder a recursos de una aplicación web desde un dominio diferente al que la alojó.

```

5
6  // Configurar CORS
7  builder.Services.AddCors(options =>
8  {
9      options.AddPolicy("AllowAll", policy =>
10     {
11         policy.AllowAnyOrigin() // Permite cualquier origen (*)
12         .AllowAnyMethod() // Permite cualquier método (GET, POST, PUT, DELETE, etc.)
13         .AllowAnyHeader(); // Permite cualquier encabezado
14     });
15 });

```

5. 24/01/2025 al 24/01/2025. Agregamos en el DbContext la consulta del procedimiento almacenado.

Evidencias Bitácora No 8			
Ficha	Aprendiz	cedula	periodo
2675810	RODNEY ZAPATA PALACIO	72209311	11/01/2025 – 25/01/2025

```
15 // Método para ejecutar el procedimiento almacenado
16
17
18 1 referencia
19 public async Task<List<FacturaElectronica>> ObtenerFacturaElectronica(int numfac)
20 {
21     return await this.Set<FacturaElectronica>()
22         .FromSqlRaw("EXEC _XML_factura_electronica @p0", numfac)
23         .ToListAsync();
24 }
25
26 }
```