

TERCER EXAMEN PARCIAL



Nombre y Apellido: Fernando Baruja

Asignatura: Optativo V

Carrera: Ing. En Informática

Semestre: 9no

Turno: Noche

Periodo: 2024-1

Profesor/a: MSc. Juan José Dos Santos

Fecha: 17 de junio del 2024

Puntos Totales: 37 (27 Examen + 10 Proceso).

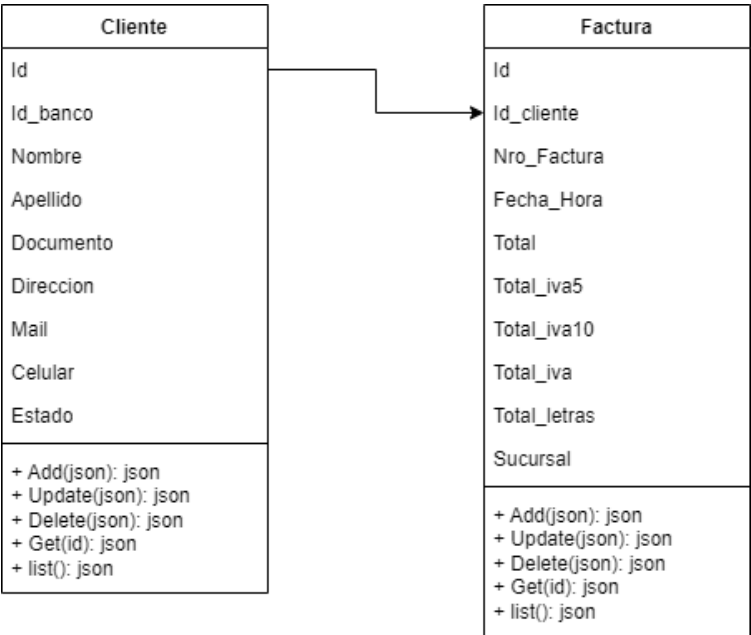
Puntos Correctos:

Porcentaje:

1ºP.: ☐ 2ºP.: ☐ 3ºP.: ☒ E.F.: ☐ E.E.: ☐

Instrucciones: Contar con todos los elementos necesarios para el examen. El examen es individual. Desarrollo en la hoja institucional con la firma del docente para que sea válido.

TEMA 1 (27P) EN BASE A LA SEGUNDA PARCIAL, REUTILIZAR EL PROYECTO Y DESARROLLAR LO APRENDIDO, EL PROYECTO DEBE ESTAR DESARROLLADO Y ES DEPENDIENTE EN LA TOTALIDAD DEL SEGUNDO EXAMEN PARCIAL, CUMPLIENDO LOS PUNTOS PROPUESTOS:



REUTILIZAR LA CAPA REPOSITORY, CON EL PROYECTO DE TIPO BIBLIOTECA DE CLASES, LA APLICACIÓN DE ENTITY FRAMEWORK TIENE QUE ESTAR COMPLETAMENTE IMPLEMENTADA.

REUTILIZAR PROYECTO DE TIPO BIBLIOTECA DE CLASES Y MANTENER LA LÓGICA Y REGLAS DE NEGOCIO, CUMPLIR CON TODOS LOS PUNTOS UTILIZANDO FLUENT VALIDATION (OBLIGATORIO):

- REUTILIZAR LA CLASE SERVICIO CLIENTE PARA LAS SIGUIENTES VALIDACIONES.
 - Nombre y Apellido Obligatorios y mínimo de caracteres 3.
 - Celular como dato numérico de longitud 10.
 - Documento no debe estar repetido, es obligatorio debe contener un mínimo de 7 caracteres.
 - Mail debe contener un mail con el formato correcto.
 - Obtener los datos del cliente solo si el estado está activo.
- DESARROLLAR LA CLASE SERVICIO FACTURA PARA LAS SIGUIENTES VALIDACIONES.
 - Nro. Factura de tipo cadena con patrones como 3 primeros caracteres con datos numéricos, 4to carácter guión, posiciones del 5 al 7 con datos numéricos, 8va posición con guión, 6 caracteres últimos con datos numéricos.
 - Total, Total_iva5, Total_iva10, Total_iva datos de tipo numéricos obligatorios.
 - Total en letras obligatorio, al menos 6 caracteres.

INDICADORES

LOS INDICADORES ESTÁN SUJETOS A LA IMPORTACIÓN Y USO DE ENTITY FRAMEWORK.

- CREA CORRECTAMENTE LAS 3 CAPAS (REPOSITORY, SERVICE Y WEBAPI) (3 PUNTOS)
- UTILIZA CORRECTAMENTE LA CAPA DE REPOSITORY Y LAS CLASES PROGRAMADAS (3 PUNTOS)
 - CONTIENE UNA CLASE CONEXIÓN
 - CONTIENE LAS CLASES MODELOS
 - CONTIENE LAS CLASES REPOSITORY PARA LOS MÉTODOS CRUD Y LISTADO
- UTILIZA CORRECTAMENTE LA CAPA DE SERVICE CON FLUENT VALIDATION (6 PUNTOS).
 - CONTIENE TODAS LAS VALIDACIONES DE LA CLASE CLIENTE.
 - CONTIENE TODAS LAS VALIDACIONES DE LA CLASE FACTURA.
 - CONECTA CORRECTAMENTE A LA CAPA REPOSITORY
- CREA CORRECTAMENTE LOS ENDPOINTS DE LA CAPA WEBAPI (4 PUNTOS).
- REALIZA CORRECTAMENTE EL CRUD Y EL LISTADO DE CLIENTES (5 PUNTOS).
- REALIZA CORRECTAMENTE EL CRUD Y EL LISTADO DE FACTURAS (5 PUNTOS).
- LA SOLUCIÓN NO PRESENTA NINGÚN ERROR (1 PUNTO).

```

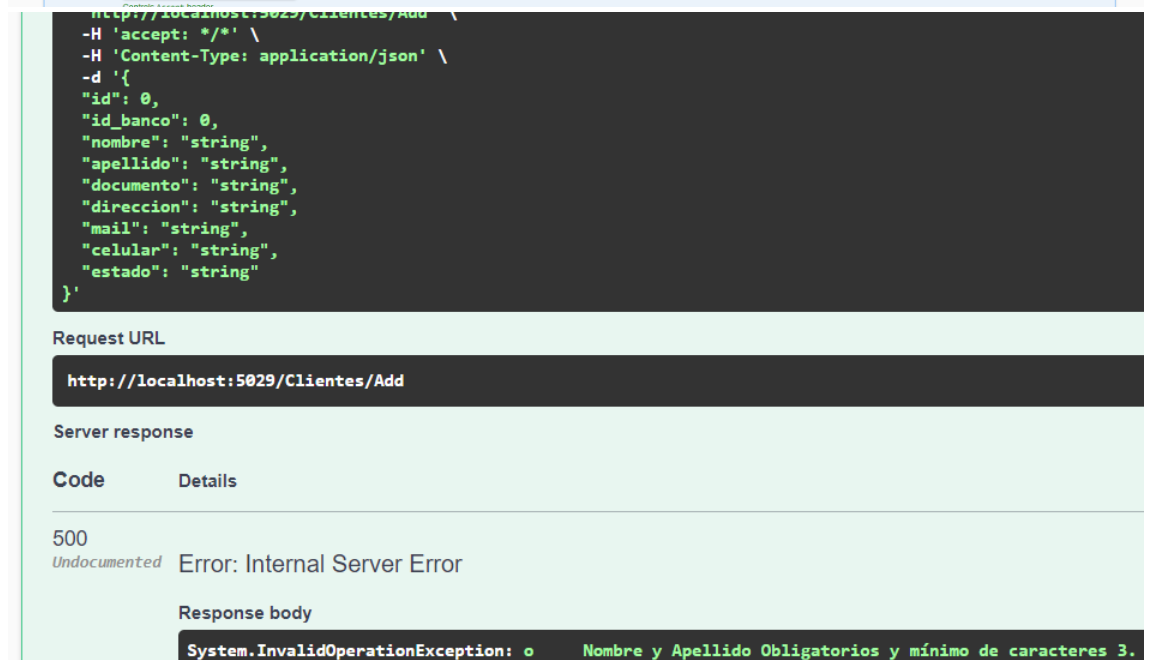
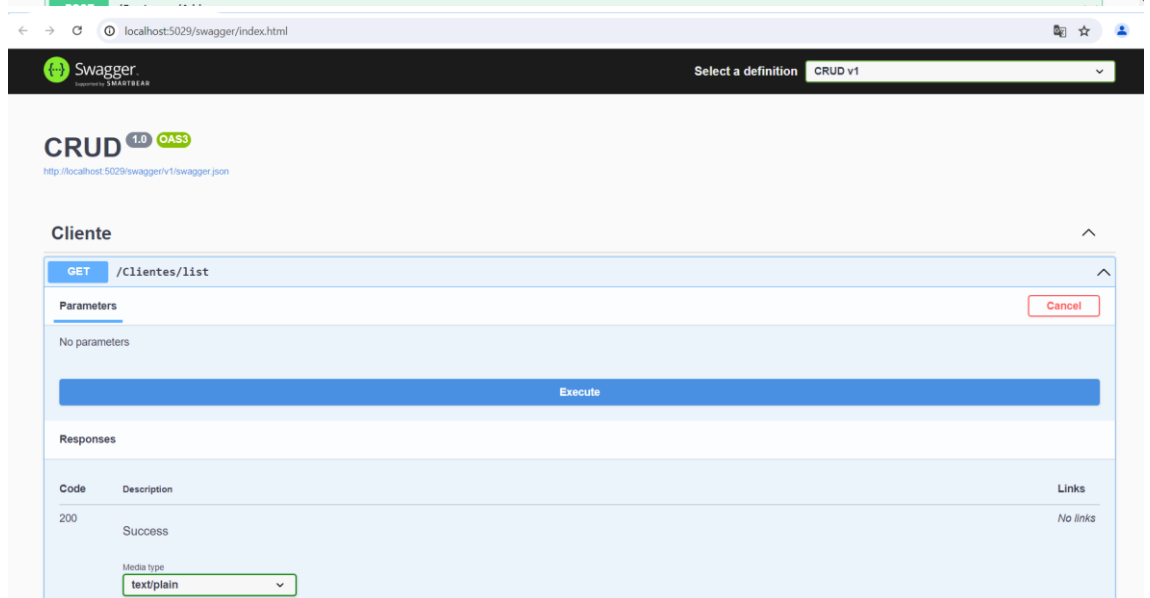
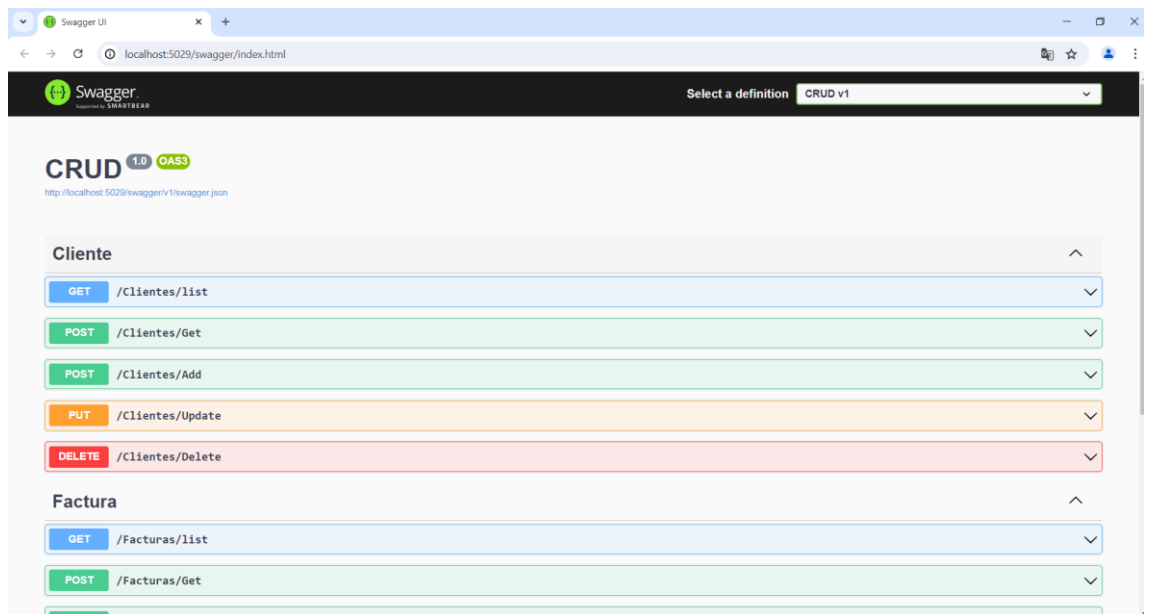
1  using CRUD.Model.Dto;
2  using FluentValidation;
3
4  namespace CRUD.Validaciones
5  {
6      1 referencia
7      public class ClienteDtoValidator : AbstractValidator<ClienteDto>
8      {
9          0 referencias
10         public ClienteDtoValidator()
11         {
12             RuleFor(x => x.id_banco).NotEmpty();
13             RuleFor(x => x.nombre).NotEmpty().MinimumLength(3);
14             RuleFor(x => x.apellido).NotEmpty().MinimumLength(3);
15             RuleFor(x => x.documento).NotEmpty().MinimumLength(7);
16             RuleFor(x => x.mail).NotEmpty().EmailAddress();
17             RuleFor(x => x.celular).NotEmpty().Length(10).Matches("[0-9]*$");
18             RuleFor(x => x.estado).NotEmpty();
19         }
20     }
21 }

```

```

1  using CRUD.Model.Dto;
2  using FluentValidation;
3
4  namespace CRUD.Validaciones
5  {
6      1 referencia
7      public class FacturaDtoValidator : AbstractValidator<FacturaDto>
8      {
9          0 referencias
10         public FacturaDtoValidator()
11         {
12             RuleFor(x => x.nro_factura).Matches(@"^\d{3}-\d{3}-\d{6}$").WithMessage("El número de factura no cumple con el formato es
13             RuleFor(x => x.total).NotEmpty();
14             RuleFor(x => x.total_iva5).NotEmpty();
15             RuleFor(x => x.total_iva10).NotEmpty();
16             RuleFor(x => x.total_iva).NotEmpty();
17             RuleFor(x => x.total_letras).NotEmpty();
18         }
19     }
20 }

```



```
"total_letras": 0,  
"sucursal": "string"  
'
```

Request URL

```
http://localhost:5029/Facturas/Add
```

Server response

Code	Details
------	---------

500	
-----	--

Undocumented

Error: Internal Server Error

Response body

```
System.InvalidOperationException: Total en letras obligatorio, al menos 6 caracteres  
o active 'CRUD.Controllers.FacturaController'.
```