

TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE

TENDENCIAS ACTUALES DE LA PROGRAMACIÓN

Ciclo:

Nombres Completos:

CRITERIO	DESCRIPCIÓN	EVIDENCIA(FOTOCOPIA)
Define la Entidad y un Repositorio	Crea la entidad con los atributos validados correctamente	<pre> package com.examenfinal.kevinexamenfinal.modelo; import lombok.Data; import org.springframework.data.annotation.Id; import org.springframework.data.mongodb.core.mapping.Document; import java.util.List; @Document(collection = "restaurante") @Data public class Restaurante { @Id private Long id; private String nombre; private String telefono; private String correo; private String direccion; private List<Cliente> clienteList; private Carta carta; } package com.examenfinal.kevinexamenfinal.modelo; import lombok.Data; import org.springframework.data.annotation.Id; import org.springframework.data.mongodb.core.mapping.Document; @Document(collection = "plato") @Data public class Plato { @Id private Long idPlato; private String nombrePlato; private String descripcion; private double precio; } </pre>

```
package com.examenfinal.kevinexamenfinal.modelo;

import lombok.Data;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

@Document(collection = "cliente")
@Data
public class Cliente {

    @Id
    private Long idCliente;
    private String nombreCliente;
    private String apellidoCliente;
    private String cedula;
    private String correo;
}
```

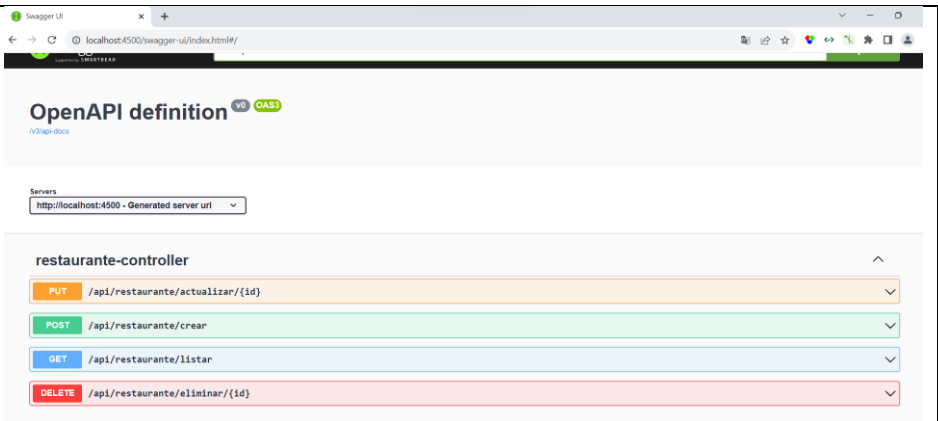
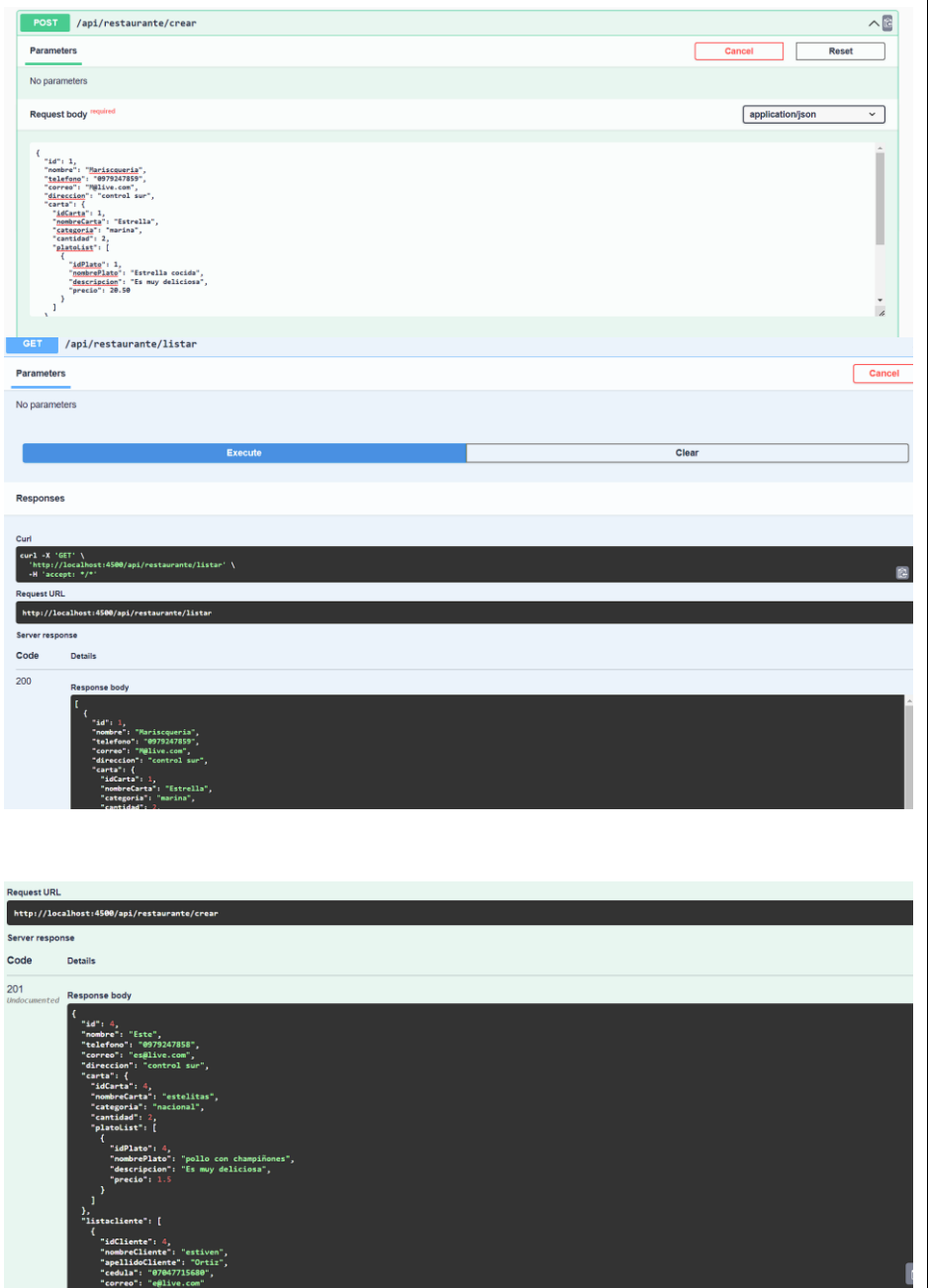
```
package com.examenfinal.kevinexamenfinal.modelo;

import lombok.Data;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;
import java.util.List;

@Document(collection = "carta")
@Data
public class Carta {

    @Id
    private Long idCarta;
    private String nombreCarta;
    private String categoria;
    private int cantidad;
    private List<Plato> platoList;
}
```

	Define y extiende la interfaz del Repositorio correctamente	<pre> package com.examenfinal.kevinexamenfinal.repository; import com.examenfinal.kevinexamenfinal.modelo.Restaurante; import org.springframework.data.mongodb.repository.MongoRepository; public interface RestauranteRepository extends MongoRepository<Restaurante, Long> { } </pre>
Define la Clase Controladora	Crea la clase controlador con los métodos GET, POST, PUT, DELETE correctamente	<pre> /** * @RestController * @RequestMapping("/api/restaurante") */ public class RestauranteController { @Autowired RestauranteServiceImpl restauranteService; @GetMapping("/listar") public ResponseEntity<List<Restaurante>> listarProductos() { return new ResponseEntity<>(restauranteService.findAll(), HttpStatus.OK); } @PostMapping("/crear") public ResponseEntity<Restaurante> crearProducto(@RequestBody Restaurante p) { return new ResponseEntity<>(restauranteService.save(p), HttpStatus.CREATED); } @PutMapping("/actualizar/{id}") public ResponseEntity<Restaurante> actualizarProducto(@PathVariable Long id, @RequestBody Restaurante p) { Restaurante restaurante = restauranteService.findById(id); if (restaurante == null) { return new ResponseEntity<>(HttpStatus.NOT_FOUND); } else { try { restaurante.setNombre(p.getNombre()); restaurante.setTelefono(p.getTelefono()); restaurante.setCorreo(p.getCorreo()); restaurante.setDireccion(p.getDireccion()); restaurante.setListacliente(p.getListacliente()); restaurante.setCarta(p.getCarta()); return new ResponseEntity<>(restauranteService.save(p), HttpStatus.OK); } catch (DataAccessException e) { return new ResponseEntity<>(HttpStatus.INTERNAL_SERVER_ERROR); } } } @DeleteMapping("/eliminar/{id}") public ResponseEntity<Restaurante> eliminarProducto(@PathVariable Long id) { restauranteService.delete(id); return new ResponseEntity<>(HttpStatus.OK); } } </pre> <p>Puerto del swagger: 4500</p>

		
Crea el microservicio	Crea el microservicio e implementa los métodos de las operaciones CRUD correctamente	

		<div><div>Parameters</div><div><div>Cancel</div></div><table><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>id ^{required}</td><td></td></tr><tr><td>integer(\$int64)</td><td>4</td></tr><tr><td>(path)</td><td></td></tr></tbody></table><div><div>Execute</div><div>Clear</div></div></div> <div>Responses</div> <div>Curl</div> <div><pre>curl -X 'DELETE' \ http://localhost:6500/api/restaurant/eliminar/4/ \ -H 'accept: */*' </pre></div> <div>Request URL</div> <div><pre>http://localhost:6500/api/restaurant/eliminar/4 </pre></div>	Name	Description	id ^{required}		integer(\$int64)	4	(path)	
Name	Description									
id ^{required}										
integer(\$int64)	4									
(path)										

Docker, Mongo y consultas

Realiza la creación de la Bd, y las consultas NoSql respectivas, con el puerto respectivo.

```
version: '3.3'
services:
  app:
    build: .
    ports:
      - 4500:8080
    depends_on:
      - mongo
    environment:
      SPRING_DATA_MONGODB_HOST: mongo
      SPRING_DATA_MONGODB_PORT: 27017
      SPRING_DATA_MONGODB_DATABASE: bd_examen

  mongo:
    image: mongo
    container_name: mongoddb
    restart: always

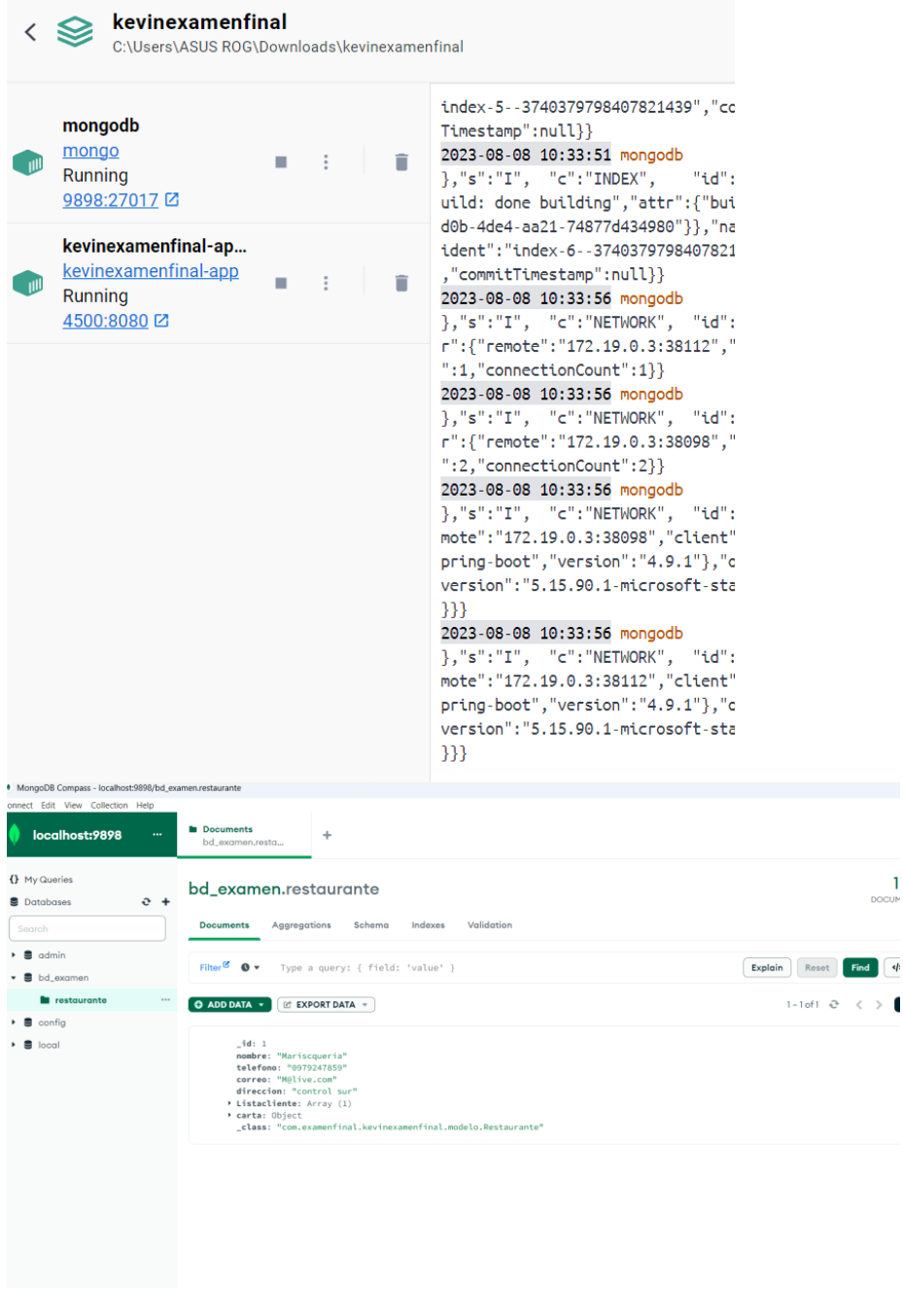
    ports:
      - 9898:27017

volumes:
  mongo-data:
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\ASUS ROG\Downloads\kevinexamenfinal> docker-compose up
[*] Running 3/3
  Network kevinexamenfinal_default Created                                0.1s
  Container mongoddb Created                                              0.2s
  Container kevinexamenfinal-app-1 Created                               0.2s
Attaching to kevinexamenfinal-app-1, mongoddb
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.884+00:00"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"","msg":"Initialized wire specifica
tion","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"o
utgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true}}}]
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.884+00:00"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"main", "msg":"Automatically disabling
TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}]
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.887+00:00"},"s":"I",  "c":"NETWORK",  "id":4648601, "ctx":"main", "msg":"Implicit TCP FastOpen u
navailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}]
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.810+00:00"},"s":"I",  "c":"REPL",     "id":5123808, "ctx":"main", "msg":"Successfully registered
PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}]
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.810+00:00"},"s":"I",  "c":"REPL",     "id":5123808, "ctx":"main", "msg":"Successfully registered
PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"}}]
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.810+00:00"},"s":"I",  "c":"REPL",     "id":5123808, "ctx":"main", "msg":"Successfully registered
PrimaryOnlyService","attr":{"service":"ShardSplitDonorService","namespace":"config.tenantSplitDonors"}}]
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.810+00:00"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"main", "msg":"Multi threading initial
ized"}]
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.811+00:00"},"s":"I",  "c":"CONTROL",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starti
ng","attr":{"pid":1,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"8472ea199535"}}]
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.811+00:00"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","a
ttr":{"buildInfo":{"version":"6.0.7","gitVersion":"202ad4fda2618c652e35f5981ef2f903d8dd1fa","opensslVersion":"OpenSSL 3.0.2 15 Mar 2022","modules":[],"allo
cator":"tcmalloc","environment":{"distmod":"ubuntu2204","distarch":"x86_64","target_arch":"x86_64"}}}}]
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.811+00:00"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating Syst
em","attr":{"os":{"name":"Ubuntu","version":"22.04"}}}]
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.811+00:00"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by
command line","attr":{"options":{"net":{"bindip":"*"}}}}]
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.814+00:00"},"s":"I",  "c":"STORAGE",  "id":22297,   "ctx":"initandlisten","msg":"Using the XFS
filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem","tags":{"startupWarnings"}}]
mongoddb | [{"t":{"$date":"2023-08-08T15:33:50.815+00:00"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx":"initandlisten","msg":"Opening WiredT
iger","attr":{"config":{"createCacheSize:313M,sessionMax:38000,eviction{threads.min=4,threads.max=4},config.base=false,statistics{fast},log{enabled=tr
iger,"attr":{"config":{"createCacheSize:313M,sessionMax:38000,eviction{threads.min=4,threads.max=4},config.base=false,statistics{fast},log{enabled=tr
```

		 <p>The screenshot shows a Docker Desktop interface with two containers running: 'mongodb' and 'kevinexamenfinal-ap...'. The 'mongodb' container is running on port 27017, and the 'kevinexamenfinal-ap...' container is running on port 8080. Below the container list, the MongoDB Compass interface is shown, connected to 'localhost:27017'. The database 'bd_examen.restaurante' is selected, and the 'restaurante' collection is displayed. A document is shown with the following fields: '_id: 1', 'nombre: "Mariscueria"', 'telefono: "0979247859"', 'correo: "Mjlive.com"', 'direccion: "control sur"', 'listaclientes: Array (1)', 'carta: Object', and '_class: "com.examenfinal.kevinexamenfinal.modelo.Restaurante"'.</p>
<p>Scripts de Docker y Playground</p>	<p>Construir los archivos de docker, y Generar el código de consulta MngoDB.</p>	

		
<p>Documenta las pruebas del microservicio en Swagger</p>	<p>Documenta las pruebas del microservicio en swagger con los métodos de validación solicitados</p>	

Despliega el código fuente en GitHub y en Docker Hub	Despliega el código fuente en GitHub y en Docker Hub	
---	--	--