

CURSO : LENGUAJE DE PROGRAMACION II  
PROFESOR : Ing. Manuel Huertas Honores  
SEMESTRE : 2023-2  
CICLO : CUARTO  
SECCION : T4ET-01  
FECHA : LUNES 25 DE SEPTIEMBRE DE 2023  
DURACION : 90 MINUTOS

NOTA

ALUMNO (A) : CARLOS GABRIEL BACA MANRIQUE

### EVALUACIÓN DE LABORATORIO 1

*Tenga presente que el logro es que usted, implemente algunas opciones de una aplicación web que utilice **Maven, JPA y MySQL** en la implementación de CRUD utilizando el patrón MVC.*

#### Consideraciones generales:

- El desarrollo de la práctica es estrictamente individual.
- Está permitido el uso de apuntes, libros, materiales o recursos informáticos adicionales.
- Las preguntas deben responderse de acuerdo con los conceptos discutidos o analizados en clase y aplicando su propio criterio para dar una solución a los problemas planteados.

#### Consolidado

Pregunta	Puntaje		Llenar solo en caso de Recalificación justificada	
	Máximo	Obtenido	Sustento	Puntaje
1	20			
	...			
	...			
	...			
Nota Recalificada				

## **PREGUNTA 1**

Usted como desarrollador se le ha asignado la funcionalidad de registro de empleado, para ello debe implementar un CRUD-MVC empleando JPA y MySQL

### **TABLA : Empleado**

IdEmpleado

Apellidos

Nombres

Edad ( $\geq 18$ )

Sexo ('M' o 'F')

Salario (1500 a 2500)

REGISTRO DE EMPLEADOS					
IdEmpleado	Apellidos	Nombres	Edad	Sexo	Salario
EMP0001	TORRES	JOSE	22	M	1500
EMP0002	GARCIA	MARIA	21	F	1750
EMP0003	FLORES	JUAN	22	M	1850
EMP0004	CASAS	ANA	23	F	2000
TOTAL DE REGISTROS: 5					
<a href="#">Registrar Empleado</a>					

**NOTA: Una vez terminado el proyecto subirlo a su cuenta de Git y enviar el link correspondiente.**

REPOSITORIO: [https://github.com/BlackTail2000/CL1\\_BacaManrique](https://github.com/BlackTail2000/CL1_BacaManrique)

#### PROPIEDADES:

```
1 #hibernate
2 spring.jpa.hibernate.ddl-auto = update
3 spring.jpa.show-sql = true
4 spring.jpa.open-in-view = true
5
6 #mysql
7 spring.datasource.driver-class-name = com.mysql.cj.jdbc.Driver
8 spring.datasource.url = jdbc:mysql://localhost:3306/CL1_BacaManrique?userTimezone=true&serverTimezone=UTC
9 spring.datasource.username = root
10 spring.datasource.password = Callie&Marie123
```

#### APP:

```
1 package com.baca;
2
3 import org.springframework.boot.SpringApplication;
4
5
6 @SpringBootApplication
7 public class Cl1BacaManriqueApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(Cl1BacaManriqueApplication.class, args);
11     }
12 }
13
14
```

#### MODEL:

```

1 package com.baca.model;
2
3 import java.math.BigDecimal;
4
5 import jakarta.persistence.*;
6 import lombok.*;
7
8 @Getter @Setter
9 @NoArgsConstructor
10 @AllArgsConstructor
11 @Entity
12 public class Empleado {
13
14     @Id
15     @Column(columnDefinition = "CHAR(7)", name = "Id_Empleado")
16     private String IdEmpleado;
17     @Column(length = 45)
18     private String Apellidos;
19     @Column(length = 45)
20     private String Nombres;
21     private int Edad;
22     @Column(columnDefinition = "CHAR(1)")
23     private char Sexo;
24     @Column(columnDefinition = "DECIMAL(8,2)")
25     private BigDecimal Salario;
26 }
27

```

#### REPOSITORY:

```

1 package com.baca.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5
6 @Repository
7 public interface EmpleadoRepository extends JpaRepository<Empleado, String>{
8
9 }
10
11
12

```

#### SERVICE INTERFACE:

```

1 package com.baca.service;
2
3 import java.util.List;
4
5 import com.baca.model.Empleado;
6
7 public interface IEmpleadoService {
8
9     List<Empleado> findAllEmpleados();
10    Empleado findEmpleadoById(String IdEmpleado);
11    void saveEmpleado(Empleado empleado);
12    void removeEmpleado(Empleado empleado);
13 }
14

```

#### SERVICE IMPLEMENTACIÓN:

```

1 package com.baca.service.impl;
2
3 import java.util.List;
4
5 @Service
6 public class EmpleadoService implements IEmpleadoService {
7
8     @Autowired
9     private EmpleadoRepository empleadoRepository;
10
11     @Override
12     public List<Empleado> findAllEmpleados() {
13         return empleadoRepository.findAll();
14     }
15
16     @Override
17     public Empleado findEmpleadoById(String IdEmpleado) {
18         Empleado empleado = null;
19         Optional<Empleado> optional = empleadoRepository.findById(IdEmpleado);
20         if(optional.isPresent())
21             empleado = optional.get();
22
23         return empleado;
24     }
25
26     @Override
27     public void saveEmpleado(Empleado empleado) {
28         empleadoRepository.save(empleado);
29     }
30
31     @Override
32     public void removeEmpleado(Empleado empleado) {
33         empleadoRepository.delete(empleado);
34     }
35 }
36

```

#### CONTROLLER:

```

1 package com.baca.controller;
2
3 import java.math.BigDecimal;
4
5
6
7
8
9
10
11
12
13
14
15
16
17 @Controller
18 public class EmpleadoController {
19
20     @Autowired
21     private IEmpleadoService empleadoService;
22
23
24     @GetMapping("/Empleado/Listado")
25     public String Listar(Model model) {
26         List<Empleado> empleados = empleadoService.findAllEmpleados();
27         model.addAttribute("empleados", empleados);
28
29         return "ListadoEmpleados";
30     }
31
32     @GetMapping("/Empleado/Registro")
33     public String Registro(Model model) {
34         Empleado empleado = new Empleado();
35         empleado.setEdad(18);
36         empleado.setSalario(new BigDecimal(1500));
37         model.addAttribute("empleado", empleado);
38         return "RegistrarEmpleado";
39     }
40
41     @PostMapping("/Empleado/Registrar")
42     public String Registrar(@ModelAttribute("empleado") Empleado empleado) {
43         empleadoService.saveEmpleado(empleado);
44         return "redirect:/Empleado/Listado";
45     }
46
47     @GetMapping("/Empleado/Eliminar")
48     public String Eliminar(@RequestParam("id") String IdEmpleado) {
49         Empleado empleado = empleadoService.findEmpleadoById(IdEmpleado);
50         empleadoService.removeEmpleado(empleado);
51
52         return "redirect:/Empleado/Listado";
53     }
54
55     @GetMapping("/Empleado/Actualizacion")
56     public String Actualizacion(@RequestParam("id") String IdEmpleado, Model model) {
57         Empleado empleado = empleadoService.findEmpleadoById(IdEmpleado);
58         model.addAttribute("empleado", empleado);
59         return "ActualizarEmpleado";
60     }
61
62     @PostMapping("/Empleado/Actualizar")
63     public String Actualizar(@ModelAttribute("empleado") Empleado empleado) {
64         empleadoService.saveEmpleado(empleado);
65         return "redirect:/Empleado/Listado";
66     }
67 }

```

**REGISTRAR:**

```

9<body class="bg-success py-3">
10<div class="p-5 border border-5 border-success-subtle rounded bg-light m-auto col-sm-12 col-md-9 col-lg-6">
11<h1>Registrar Empleado</h1>
12<hr>
13<form action="#" th:action="@{/Empleado/Registrar}" th:object="${empleado}" method="post">
14<div class="col-6 mb-3">
15<label class="form-label" for="IdEmpleado">Id</label>
16<input type="text" class="form-control" id="IdEmpleado" th:field="*{IdEmpleado}" autocomplete="off" required placeholder="EMPXXXX">
17</div>
18<div class="row mb-3">
19<div class="col-6 mb-3">
20<label class="form-label" for="Nombres">Nombres</label>
21<input type="text" class="form-control" id="Nombres" th:field="*{Nombres}" autocomplete="off" required placeholder="Ingresar Nombres...">
22</div>
23<div class="col-6 mb-3">
24<label class="form-label" for="Apellidos">Apellidos</label>
25<input type="text" class="form-control" id="Apellidos" th:field="*{Apellidos}" autocomplete="off" required placeholder="Ingresar Apellidos...">
26</div>
27</div>
28<div class="row mb-3">
29<div class="col-6 mb-3">
30<label class="form-label" for="Edad">Edad</label>
31<input type="number" class="form-control" id="Edad" th:field="*{Edad}" autocomplete="off" min="18">
32</div>
33<div class="col-6 mb-3">
34<label class="form-label" for="Sexo">Sexo</label>
35<select class="form-select" id="Sexo" th:field="*{Sexo}">
36<option value="M">Masculino</option>
37<option value="F">Femenino</option>
38</select>
39</div>
40</div>
41<div class="row mb-3">
42<div class="col-6 mb-3">
43<label class="form-label" id="Salario">Salario</label>
44<input type="number" step="0.01" class="form-control" id="Salario" th:field="*{Salario}" min="1500" max="2500">
45</div>
46</div>
47<div class="d-flex justify-content-around">
48<button type="submit" class="btn btn-success"><i class="bi bi-database-add"></i> Registrar</button>
49<a class="btn btn-secondary" th:href="@{/Empleado/Listado}"><i class="bi bi-door-open"></i> Regresar</a>
50</div>
51</form>
52</div>
53</body>
54<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9KDrWmte7bh95OGNy2PhcTNXj1NW7RuBCsyN/o0jlpcV8Qy46cDfL" cross
55</html>

```

## LISTAR:

```

9<body class="p-5 bg-success">
10<div class="p-5 border border-5 border-success-subtle rounded bg-light">
11<h1>Listado de Empleados</h1>
12<hr>
13<div class="text-center">
14<a class="btn btn-outline-success my-3" th:href="@{/Empleado/Registro}"><i class="bi bi-database-add"></i> Registrar Nuevo</a>
15</div>
16<table class="table table-hover table-bordered table-striped">
17<thead>
18<tr>
19<th>ID</th>
20<th>Apellidos</th>
21<th>Nombres</th>
22<th>Edad</th>
23<th>Sexo</th>
24<th>Salario</th>
25<th colspan="2"></th>
26</tr>
27</thead>
28<tbody>
29<tr>
30<th:block th:each="empleado: ${empleados}">
31<tr>
32<td th:text="${empleado.getIdEmpleado()}"></td>
33<td th:text="${empleado.getApellidos()}"></td>
34<td th:text="${empleado.getNombres()}"></td>
35<td th:text="${empleado.getEdad()}"></td>
36<td th:text="${empleado.getSexo()}"></td>
37<td th:text="${empleado.getSalario()}"></td>
38<td class="text-center"><a class="btn btn-outline-primary" th:href="@{/Empleado/Actualizacion(id=${empleado.getIdEmpleado()})}"><i class="bi bi-pencil"></i> E
39<td class="text-center"><a class="btn btn-outline-danger" th:href="@{/Empleado/Eliminar(id=${empleado.getIdEmpleado()})}"><i class="bi bi-trash"></i> Eliminar
40</td>
41</tr>
42</tbody>
43<tfoot>
44<tr class="text-center">
45<th colspan="8">TOTAL DE REGISTROS: ' + ${empleados.size()}></th>
46</tr>
47</tfoot>

```

## ACTUALIZAR:

```

9<body class="bg-success py-3">
10<div class="p-5 border border-5 border-success-subtle rounded bg-light m-auto col-sm-12 col-md-9 col-lg-6">
11<h1>Editar Empleado Nro. ' + ${empleado.getIdEmpleado()}></h1>
12<hr>
13<form action="#" th:action="@{/Empleado/Actualizar}" th:object="${empleado}" method="post">
14<input type="hidden" th:field="*{IdEmpleado}">
15<div class="row mb-3">
16<div class="col-6 mb-3">
17<label class="form-label" for="Nombres">Nombres</label>
18<input type="text" class="form-control" id="Nombres" th:field="*{Nombres}" autocomplete="off" required placeholder="Ingresar Nombres...">
19</div>
20<div class="col-6 mb-3">
21<label class="form-label" for="Apellidos">Apellidos</label>
22<input type="text" class="form-control" id="Apellidos" th:field="*{Apellidos}" autocomplete="off" required placeholder="Ingresar Apellidos...">
23</div>
24</div>
25<div class="row mb-3">
26<div class="col-6 mb-3">
27<label class="form-label" for="Edad">Edad</label>
28<input type="number" class="form-control" id="Edad" th:field="*{Edad}" autocomplete="off" min="18">
29</div>
30<div class="col-6 mb-3">
31<label class="form-label" for="Sexo">Sexo</label>
32<select class="form-select" id="Sexo" th:field="*{Sexo}">
33<option value="M" th:selected="${empleado.getSexo() == 'M'}">Masculino</option>
34<option value="F" th:selected="${empleado.getSexo() == 'F'}">Femenino</option>
35</select>
36</div>
37</div>
38<div class="row mb-3">
39<div class="col-6 mb-3">
40<label class="form-label" id="Salario">Salario</label>
41<input type="number" step="0.01" class="form-control" id="Salario" th:field="*{Salario}" min="1500" max="2500">
42</div>
43</div>
44<div class="d-flex justify-content-around">
45<button type="submit" class="btn btn-primary"><i class="bi bi-pencil"></i> Editar</button>
46<a class="btn btn-secondary" th:href="@{/Empleado/Listado}"><i class="bi bi-door-open"></i> Regresar</a>

```

EVIDENCIA:

LISTADO

Listado de Empleados

Registrar Nuevo

ID	Apellidos	Nombres	Edad	Sexo	Salario		
EMP0001	TORRES	JOSE	22	M	1500.00	<div>Editar</div>	<div>Eliminar</div>
EMP0002	GARCIA	MARIA	21	F	1750.00	<div>Editar</div>	<div>Eliminar</div>
EMP0003	FLORES	JUAN	22	M	1850.00	<div>Editar</div>	<div>Eliminar</div>
EMP0004	CASAS	ANA	23	F	2000.00	<div>Editar</div>	<div>Eliminar</div>
TOTAL DE REGISTROS: 4							



REGISTRO

# Registrar Empleado

Id

EMP0005

Nombres:

CARLOS

Apellidos:

BACA

Edad:

23

Sexo:

Masculino

Salario:

2000

Registrar

Regresar

ID	Apellidos	Nombres	Edad	Sexo	Salario		
EMP0001	TORRES	JOSE	22	M	1500.00	<a href="#">Editar</a>	<a href="#">Eliminar</a>
EMP0002	GARCIA	MARIA	21	F	1750.00	<a href="#">Editar</a>	<a href="#">Eliminar</a>
EMP0003	FLORES	JUAN	22	M	1850.00	<a href="#">Editar</a>	<a href="#">Eliminar</a>
EMP0004	CASAS	ANA	23	F	2000.00	<a href="#">Editar</a>	<a href="#">Eliminar</a>
EMP0005	BACA	CARLOS	23	M	2000.00	<a href="#">Editar</a>	<a href="#">Eliminar</a>
TOTAL DE REGISTROS: 5							

EDITAR:

Editar Empleado Nro. EMP0005

Nombres:

CHARLIE

Apellidos:

BACA

Edad:

20

Sexo:

Femenino

Salario:

2500.00

Editar

Regresar

ID	Apellidos	Nombres	Edad	Sexo	Salario		
EMP0001	TORRES	JOSE	22	M	1500.00	Editar	Eliminar
EMP0002	GARCIA	MARIA	21	F	1750.00	Editar	Eliminar
EMP0003	FLORES	JUAN	22	M	1850.00	Editar	Eliminar
EMP0004	CASAS	ANA	23	F	2000.00	Editar	Eliminar
EMP0005	BACA	CHARLIE	20	F	2500.00	Editar	Eliminar
TOTAL DE REGISTROS: 5							

ELIMINAR:

Listado de Empleados

Registrar Nuevo

ID	Apellidos	Nombres	Edad	Sexo	Salario		
EMP0001	TORRES	JOSE	22	M	1500.00	Editar	Eliminar
EMP0002	GARCIA	MARIA	21	F	1750.00	Editar	Eliminar
TOTAL DE REGISTROS: 2							

Listado de Empleados

Registrar Nuevo

ID	Apellidos	Nombres	Edad	Sexo	Salario		
TOTAL DE REGISTROS: 0							

RÚBRICA PARA EL EXAMEN FINAL					
PUNTOS	EXCELENTE 20 PUNTOS	BUENO 16 PUNTOS	REGULAR 12 PUNTOS	REGULAR 8 PUNTOS	DEFICIENTE 0 PUNTOS
20 Puntos	<p>Crear correctamente el proyecto Web Dinámico con Maven (carga de dependencias en su archivo de configuración), implementa correctamente el patrón MVC para las clases correspondientes, configura correctamente el archivo de configuración para la conexión con MySQL. Hace uso del repositorio de JPA para realizar los procesos de listado y registro. Carga el proyecto y lo comparte a través de su cuenta de Git.</p>	<p>Crear correctamente el proyecto Web Dinámico con Maven (carga de dependencias en su archivo de configuración), implementa parcialmente el patrón MVC para las clases correspondientes, configura correctamente el archivo de configuración para la conexión con MySQL. Hace uso del repositorio de JPA para realizar los procesos de listado y registro. Carga el proyecto y lo comparte a través de su cuenta de Git.</p>	<p>Crear correctamente el proyecto Web Dinámico con Maven (carga de dependencias en su archivo de configuración), implementa parcialmente el patrón MVC para las clases correspondientes, configura correctamente el archivo de configuración para la conexión con MySQL. Hace uso del repositorio de JPA para realizar el proceso de listado-</p>	<p>Crear correctamente el proyecto Web Dinámico con Maven (carga de dependencias en su archivo de configuración), implementa parcialmente el patrón MVC para las clases correspondientes, configura correctamente el archivo de configuración para la conexión con MySQL.</p>	<p>No implementa Ninguna funcionalidad.</p>