

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA
CENTRO UNIVERSITARIO DE CHIMALTENANGO
CARRERA: INGENIERÍA EN SISTEMAS DE LA INFORMACIÓN
CURSO: programación II Sección “A”

“Manual técnico”

ANDER ABIMAELO LOPEZ XIQUITA
CARNE: 1990-22-1923
CHIMALTENANGO GUATEMALA
11 DE NOVIEMBRE DE 2023

Manuel Técnico

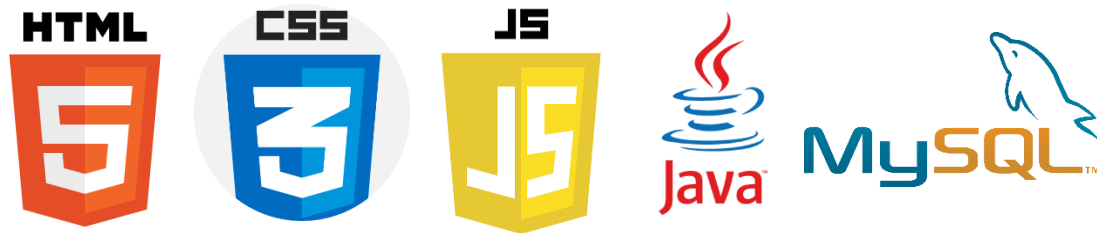
CRUD de empleados

CRUD WEB TIENDA DEPORTIVA

Este CRUD fue hecho con el objetivo de poder hacer un conteo y registro de todos los empleados que trabajan en nuestra tienda deportiva, la web podrá ser usada por varios usuarios que podrían ser los empleados al igual que podrá ser utilizada por el administrador de la tienda. En el siguiente documento será explicado cual fue el proceso y cuáles fueron las tecnologías que se utilizaron para realizar el proyecto.

Tecnologías utilizadas

Como se trata de un proyecto Web, se utilizaron las principales tecnologías del desarrollo web, como lo son HTML5 y CSS, y para el lado del backend se utilizaron las tecnologías de JAVA y MySQL.



Sin embargo, usar algunas de estas tecnologías de manera vanilla puedo haber hecho que el proyecto se llevara mas tiempo de lo requerido, por lo que se decidió utilizar algunos Frameworks como lo son: Bootstrap y Spring Boot.



Esto nos permitirá ahorrar mucho tiempo y complicaciones a la hora de desarrollar el sistema.

Front end

Esta es la parte del sistema con la que el usuario interactúa, esta le permite visualizar todos los datos y también le permite interactuar con ellos. En este caso decidimos utilizar las tecnologías de HTML5, CSS3 y JAVASCRIPT para poder maquetar las paginas, siendo html la encargada de estructurar las paginas, css3 y Bootstrap para darle diseño a la estructura html y finalmente javascript para darle pequeñas funcionalidades y animaciones a la parte del Front end.

```
51
52 <div class="container">
53   <div th:if="${warning != null}"
54     class="alert alert-warning alert-dismissible fade show">
55     [[${warning}]]>
56     <button type="button" class="close" data-dismiss="alert"
57       aria-label="Close"></button>
58   </div>
59 </div>
60
61 <div class="container">
62   <div th:if="${info != null}"
63     class="alert alert-info alert-dismissible fade show">
64     [[${info}]]>
65     <button type="button" class="close" data-dismiss="alert"
66       aria-label="Close"></button>
67   </div>
68 </div>
69
70 <footer th:fragment="footer">
71   <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
72     integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KckRr/rE9/Qp6GaAZGJwFDMVNA/6pGFF93hXpG6SKhN"
73     crossorigin="anonymous"></script>
74   <script
75     src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
76     integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
77     crossorigin="anonymous"></script>
78   <script
79     src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"
80     integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
81     crossorigin="anonymous"></script>
82 </footer>
```

Back end

Esta es la parte que se encarga de hacer que la pagina cobre vida, esta parte no la ve el usuario, ya que ocurre de tras de todo, trabajando con la base de datos y con toda la parte lógica para que finalmente se logren visualizar los datos que se necesitan.

En este caso por ser la herramienta principal del curso tuvimos que trabajar con Java como nuestro lenguaje de programación, y para la base de datos usamos MySQL.

Java junto a Spring Boot no dejo montar toda la parte lógica de la pagina de una manera fácil y rápida.

```
33
34 @Controller
35 public class EmpleadoController {
36
37     @Autowired
38     private EmpleadoService empleadoService;
39
40     @GetMapping("/ver/{id}")
41     public String verDetallesDelEmpleado(@PathVariable(value = "id") Long id, Map<String, Object> modelo, RedirectAttributes flash) {
42         Empleado empleado = empleadoService.findOne(id);
43         if(empleado == null) {
44             flash.addFlashAttribute("error", "El empleado no existe en la base de datos");
45             return "redirect:/listar";
46         }
47
48         modelo.put("empleado", empleado);
49         modelo.put("titulo", "Detalles del empleado " + empleado.getNombre());
50         return "ver";
51     }
52
53     @GetMapping("/{}/listar", "")
54     public String listarEmpleados(@RequestParam(name = "page", defaultValue = "0") int page, Model modelo) {
55         Pageable pageRequest = PageRequest.of(page, 4);
56         Page<Empleado> empleados = empleadoService.findAll(pageRequest);
57         PageRender<Empleado> pageRender = new PageRender<>("/listar", empleados);
58
59         modelo.addAttribute("titulo", "Listado de empleados");
60         modelo.addAttribute("empleados", empleados);
61         modelo.addAttribute("page", pageRender);
62
63         return "listar";
64     }
65 }
```

Ahora mostraremos como se desarrollaron las funciones principales de la pagina web.

Listar

```
52
53 @GetMapping("/{}/listar","")
54 public String listarEmpleados(@Request Pageable pageRequest - com.gestion.empleados.controlador.EmpleadoController.listarEmpleados(int,
55 Pageable pageRequest = PageRequest Model)
56 Page<Empleado> empleados = empleadoService.findAll(pageRequest);
57 PageRender<Empleado> pageRender = new PageRender<>(url:"/listar", empleados);
58
59 modelo.addAttribute(attributeName:"titulo",attributeValue:"Listado de empleados");
60 modelo.addAttribute(attributeName:"empleados",empleados);
61 modelo.addAttribute(attributeName:"page", pageRender);
62
63 return "listar";
64
65 }
```

Ver detalles

```
39
40 @GetMapping("/ver/{id}")
41 public String verDetallesDelEmpleado(@PathVariable(value = "id") Long id,Map<String,Object> modelo,RedirectAttributes flash) {
42 Empleado empleado = empleadoService.findOne(id);
43 if(empleado == null) {
44 flash.addFlashAttribute(attributeName:"error", attributeValue:"El empleado no existe en la base de datos");
45 return "redirect:/listar";
46 }
47
48 modelo.put("empleado",empleado);
49 modelo.put("titulo", "Detalles del empleado " + empleado.getNombre());
50 return "ver";
51 }
```

Formularios

```
65
66 @GetMapping("/form")
67 public String mostrarFormularioDeRegistrarCliente(Map<String,Object> modelo) {
68 Empleado empleado = new Empleado();
69 modelo.put("empleado", empleado);
70 modelo.put("titulo", "Registro de empleados");
71 return "form";
72
73 }
74
75 @PostMapping("/form")
76 public String guardarEmpleado(@Valid Empleado empleado,BindingResult result,Model modelo,RedirectAttributes flash,SessionStatus status) {
77 if(result.hasErrors()) {
78 modelo.addAttribute(attributeName:"titulo", attributeValue:"Registro de cliente");
79 return "form";
80 }
81
82 String mensaje = (empleado.getId() != null) ? "El empleado ha sido editado con exito" : "Empleado registrado con exito";
83
84 empleadoService.save(empleado);
85 status.setComplete();
86 flash.addFlashAttribute(attributeName:"success", mensaje);
87 return "redirect:/listar";
88
89 }
90
91 @GetMapping("/form/{id}")
92 public String editarEmpleado(@PathVariable(value = "id") Long id,Map<String, Object> modelo,RedirectAttributes flash) {
93 Empleado empleado = null;
94 if(id > 0) {
95 empleado = empleadoService.findOne(id);
96 if(empleado == null) {
97 flash.addFlashAttribute(attributeName:"error", attributeValue:"El ID del empleado no existe en la base de datos");
98 return "redirect:/listar";
99 }
100 }
101 }
```

Eliminar

```
108
109 @GetMapping("/eliminar/{id}")
110 public String eliminarCliente(@PathVariable(value = "id") Long id, RedirectAttributes flash) {
111     if(id > 0) {
112         empleadoService.delete(id);
113         flash.addFlashAttribute(attributeName: "success", attributeValue: "Empleado eliminado con éxito");
114     }
115     return "redirect:/listar";
116 }
```

Exportar a PDF

```
117
118 @GetMapping("/exportarPDF")
119 public void exportarListadoDeEmpleadosEnPDF(HttpServletResponse response) throws DocumentException, IOException {
120     response.setContentType("application/pdf");
121
122     DateFormat dateFormatter = new SimpleDateFormat("yyyy-MM-dd_HH:mm:ss");
123     String fechaActual = dateFormatter.format(new Date());
124
125     String cabecera = "Content-Disposition";
126     String valor = "attachment; filename=Empleados_" + fechaActual + ".pdf";
127
128     response.setHeader(cabecera, valor);
129
130     List<Empleado> empleados = empleadoService.findAll();
131
132     EmpleadoExporterPDF exporter = new EmpleadoExporterPDF(empleados);
133     exporter.exportar(response);
134 }
135
```