

Empreses formadores

El *Departament d'Ensenyament* vol mantenir un registre de totes les empreses en les quals hi ha alumnes de la família d'informàtica que fan pràctiques de FCT o de Dual, dels diversos cicles formatius.

Per a cada alumne es vol saber el nom, els cognoms, el cicle (DAM, DAW, ASIX, SMX) i la modalitat (DUAL, FCT), l'estat de les pràctiques (ENPROCES, ACABADES) i l'empresa on les realitza.

De cada empresa es vol saber el nom, l'adreça, el CIF (que actua com a clau) i els alumnes que té fent pràctiques.

Part 1

Dissenyu una api rest amb Spring que permeti (fent servir Spring Boot Data Rest)

<https://docs.spring.io/spring-data/rest/docs/current/reference/html/>

1. fer *crud* de les empreses.

```
@Controller
@RequestMapping("/api")
public class empresaControlador {

    @Autowired
    private empresaRepo empreRepo;

    @GetMapping("empreses")
    public String llistaEmpreses(Model model) {
        model.addAttribute("empreses", empreRepo.findAll());
        return "empreses";
    }

    @GetMapping("/novaEmpresa")
    public String formNovaEmpresa(Model model) {
        model.addAttribute("empresa", new empresaEntitat());
        return "novaEmpresa";
    }

    @PostMapping("novaEmpresaAdd")
    public String crearnovaEmpresa(@ModelAttribute("empresa") empresaEntitat empresa) {
        empreRepo.save(empresa);
        return "redirect:/api/empreses";
    }

    @DeleteMapping("/empresa/{id}")
    public void deleteAlumne(@PathVariable Long id) {
        empreRepo.deleteById(id);
    }

}
```

2. fer *crud* dels alumnes.

```
@Controller
@RequestMapping("/api")
public class alumneControlador {

    @Autowired
    private alumneRepo alumnRepo;

    @Autowired
    private empresaRepo empreRepo;

    @GetMapping("/alumnes")
    public String llistaAlumnes(Model model) {
        model.addAttribute("alumnes", alumnRepo.findAll());
        return "alumnes";
    }

    @GetMapping("/nouAlumne")
    public String formNouAlumne(Model model) {
        model.addAttribute("alumne", new alumneEntitat());
        model.addAttribute("empreses", empreRepo.findAll());
        return "nouAlumne";
    }

    @PostMapping("nouAlumneAdd")
    public String crearnouAlumne(@ModelAttribute("alumne") alumneEntitat alumne) {
        alumnRepo.save(alumne);
        return "redirect:/api/alumnes";
    }

    @DeleteMapping("/alumne/{id}")
    public void deleteAlumne(@PathVariable Long id) {
        alumnRepo.deleteById(id);
    }

}
```

Indicacions: Aquesta api té com a ruta inicial /api

Part 2

A la mateixa aplicació incorporeu webs dinàmiques amb Thymeleaf per tal de <https://www.thymeleaf.org/>

1. Llistar les empreses

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8"/>
  <title>Alumne Nou</title>
</head>
<body>
  <h1>Afegir Empresa</h1>
  <form method="post" th:action="@{/api/novaEmpresaAdd}">
    <label for="cif">CIF:</label>
    <input type="text" id="cif" name="cif"><br>

    <label for="cif">Nom:</label>
    <input type="text" id="cif" name="cif"><br>

    <label for="adreca">Adreça:</label>
    <input type="text" id="adreca" name="adreca"><br>

    <button type="submit">Afegir</button>

  </form>
</body>
</html>
```

2. Llistar els alumnes

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8"/>
  <title>Alumnes</title>
</head>
<body>
  <h1>Llistat d'Alumnes</h1>
  <table>
    <tr>
      <th>Nom</th>
      <th>Cognom</th>
      <th>Cicle</th>
    </tr>
    <tr th:each="alumne : ${ns}">
      <td th:text="${alumne.nom}/>
      <td th:text="${alumne.cognom}/>
      <td th:text="${alumne.cicle}/>
    </tr>
  </table>
</body>
</html>
```

3. Fer un formulari per entrar empreses

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8"/>
  <title>Alumne Nou</title>
</head>
<body>
  <h1>Afegir Empresa</h1>
  <form method="post" th:action="@{/api/novaEmpresaAdd}">
    <label for="cif">CIF:</label>
    <input type="cif" id="cif" name="cif"><br>

    <label for="cif">Nom:</label>
    <input type="cif" id="cif" name="cif"><br>

    <label for="adreca">Adreça:</label>
    <input type="adreca" id="adreca" name="adreca"><br>

    <button type="submit">Afegir</button>

  </form>
</body>
</html>
```

4. Fer un formulari per entrar alumnes

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8"/>
  <title>Alumne Nou</title>
</head>
<body>
  <h1>Afegir Alumne</h1>
  <form method="post" th:action="@{/api/nouAlumneAdd}">
    <label for="nom">Nom:</label>
    <input type="text" id="nom" name="nom"><br>

    <label for="cognom">Cognom:</label>
    <input type="text" id="cognom" name="cognom"><br>

    <label for="cicle">Cicle:</label>
    <input type="text" id="cicle" name="cicle"><br>

    <button type="submit">Afegir</button>

  </form>
</body>
</html>
```

Part 3 (Opcional)

Una empresa pot tenir diversos alumnes fent pràctiques, però un alumne només pot fer pràctiques en una empresa

1. Establiu una relació entre alumnes i empresa

```
@OneToMany
private List<alumneEntitat> alumnes;
```

```
@ManyToOne
@JoinColumn(name = "id_empresa")
private empresaEntitat empresa;
```

SELECT * FROM ALUMNES;

ID	ID_EMPRESA	CICLE	COGNOM	ESTAT_PRACTIQUES	NOM
1	null	DAW2	Garcia	En proces	Alejandro

(1 row, 0 ms)

2. Modifiqueu l'aplicació web per tal que mostri els alumnes que inclogui un llistat dels alumnes que estan a una empresa
3. Afegiu un formulari de manera que permeti assignar els alumnes a una empresa amb uns selector

Afegir Alumne

Nom:

Cognom:

Cicle:

Empresa:

- TEC
- GOOGLE
- INFO