

Vagner Fonseca, Phd



# O que é o Angular?



- Ele é um poderoso framework front-end para aplicações web dinâmicas em HTML, CSS e TypeScript (O compilador JavaScript)
- Ele oferece uma série de ferramentas e funcionalidades que facilitam a construção de aplicativos escaláveis e de fácil manutenção.
  - Injeção de dependências, Ligação bidirecional de dados e Arquitetura modular.

# Introdução

Evolução

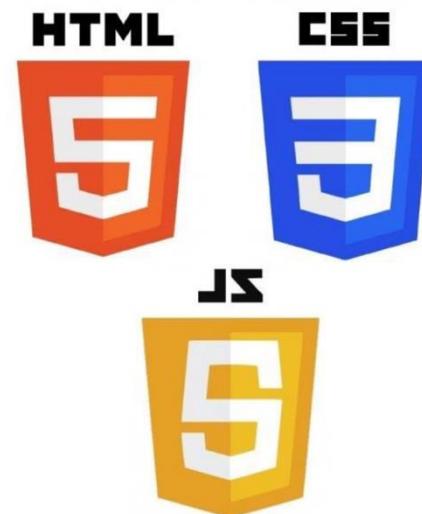


# Introdução



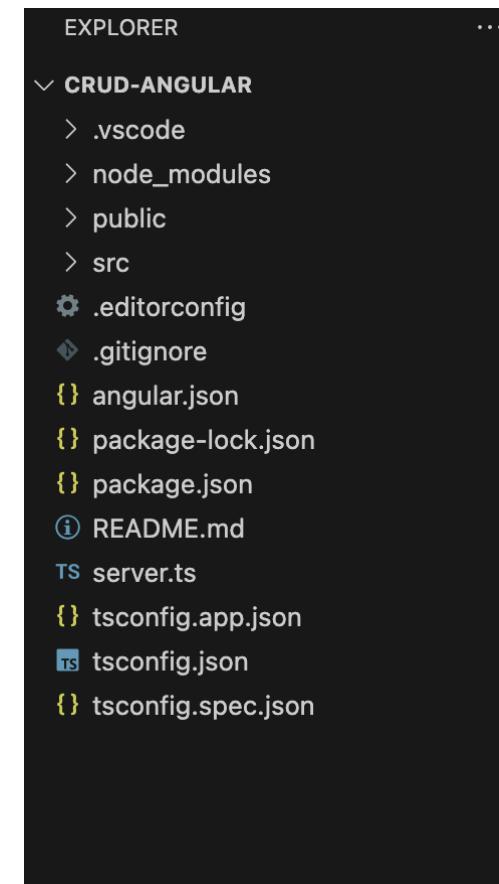
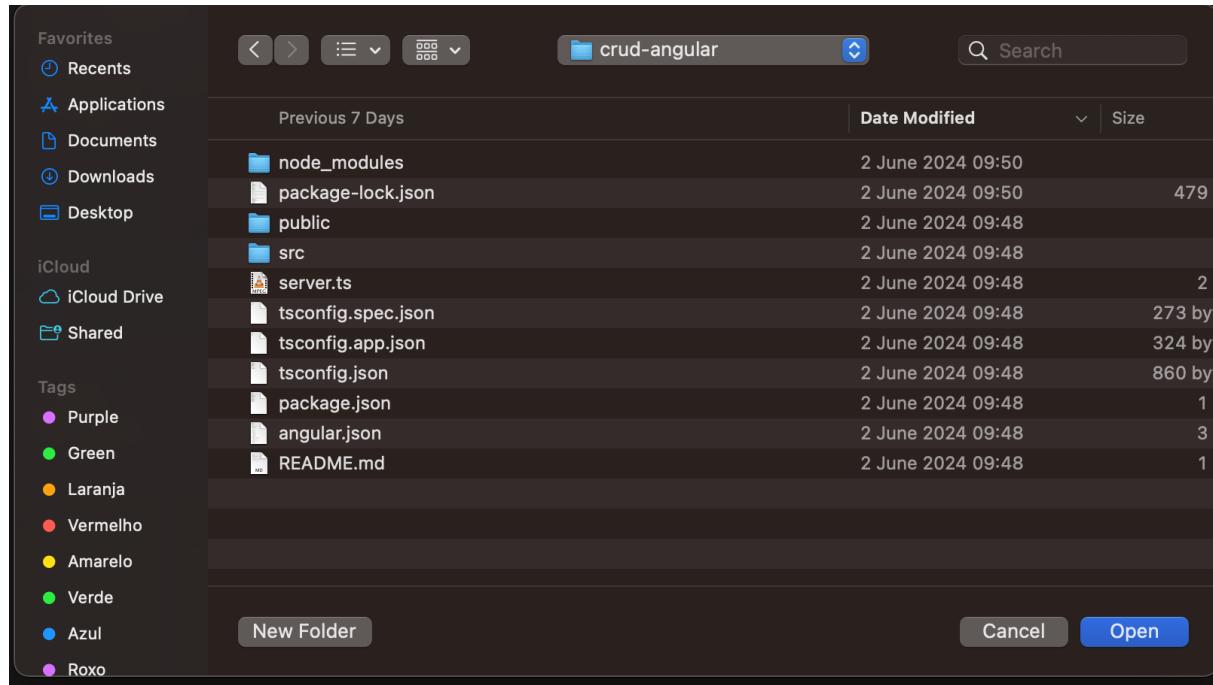
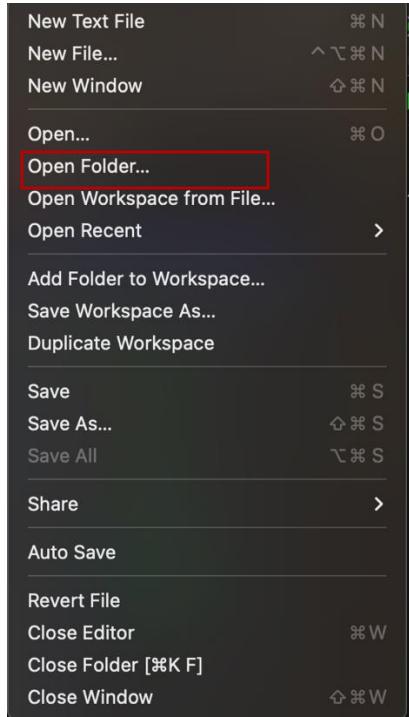
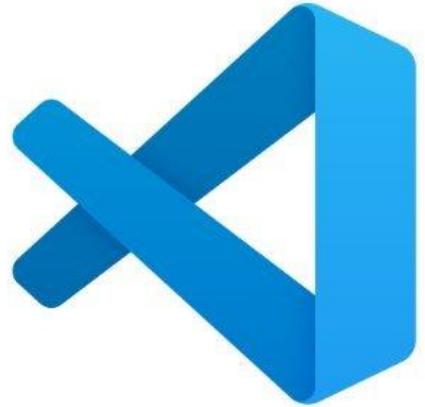
- Angular é orientado ao desenvolvimento front-end desacoplado do back-end

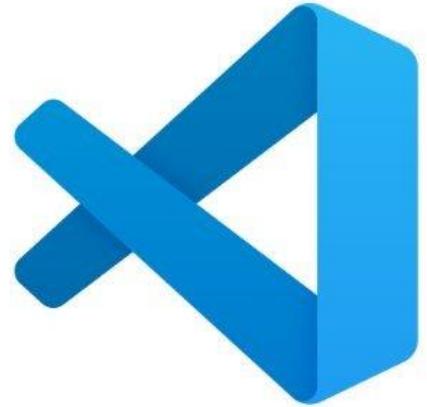
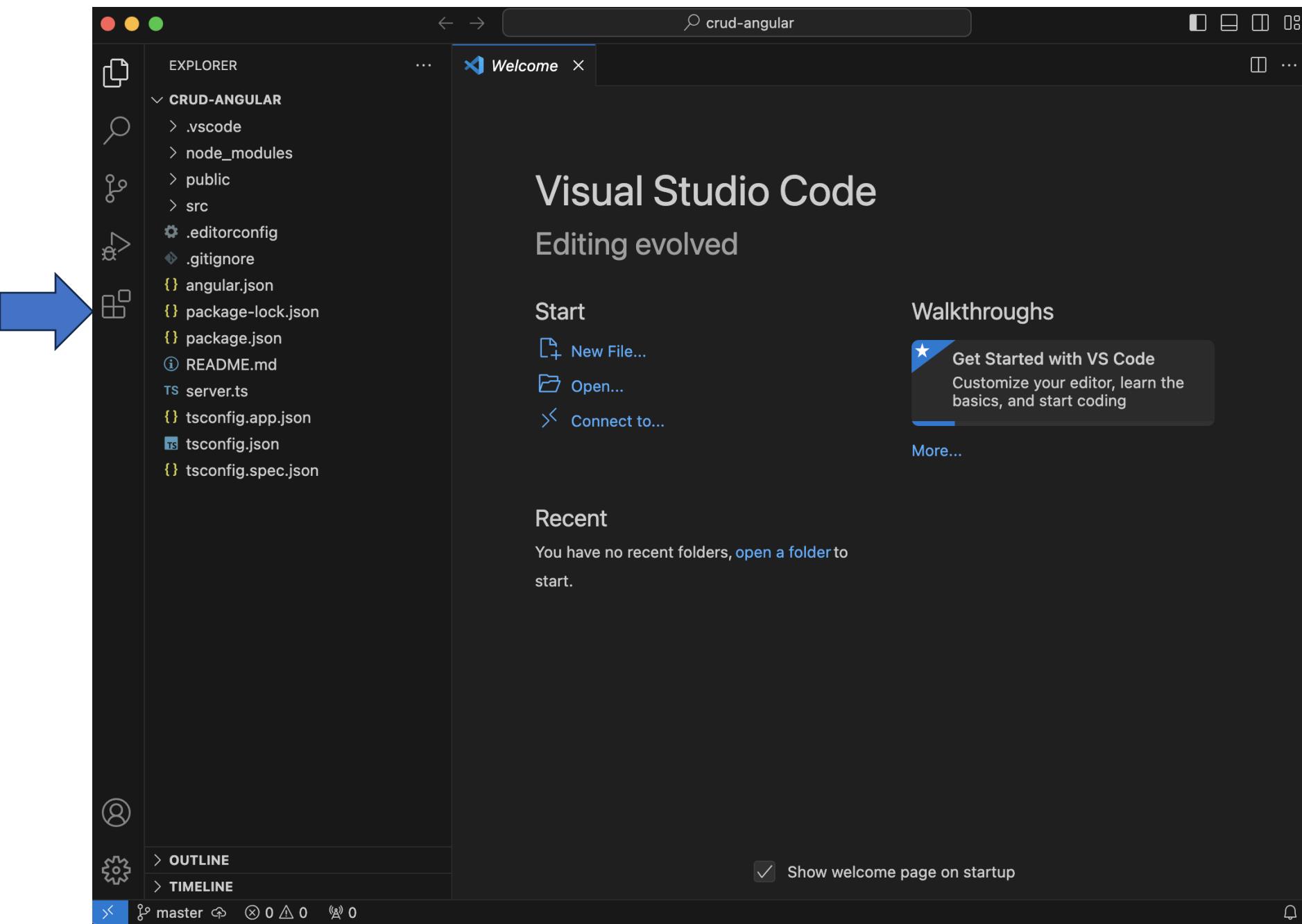
FRONT-END



BACK-END

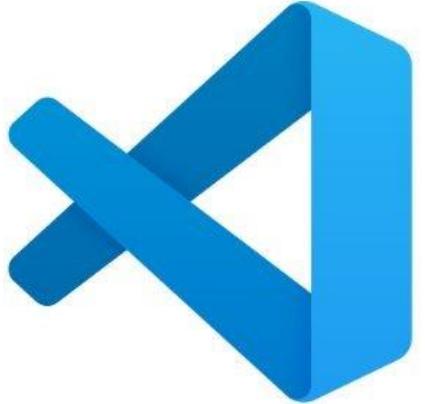






# Angular Extension Pack

The screenshot shows the Visual Studio Code Marketplace interface. The search bar at the top contains the text "crud-angular". The main panel displays the "Extension: Angular Extension Pack" item, version v1.1.2, by Loiane Groner. The extension has 357,359 installs and a 5-star rating. A brief description states: "Some of the most popular (and some I find very useful) An...". Below the description, there are "Disable" and "Uninstall" buttons, and a note that it is "This extension is enabled globally." The "DETAILS" tab is selected, showing the "Extension Pack (21)" section which lists three extensions: "Prettier - Code formatter" (by Prettier), "ESLint" (by Microsoft), and "Auto Rename Tag" (by Jun Han). To the right of the extension details, there are sections for "Categories" (listing "Extension Packs"), "Resources" (listing "Marketplace", "Issues", "Repository", "License", and "Loiane Groner"), and "More Info" (listing publication and release dates, and identifier information). On the left side of the main panel, a sidebar lists other Angular-related extensions, each with a small icon, name, description, author, and an "Install" button.



<https://marketplace.visualstudio.com/items?itemName=loiane.angular-extension-pack>



## angular.json

```
...
"projects": {
  "angular-project": {
    "projectType": "application",
    "schematics": {
      "@schematics/angular:component": {
        "style": "scss"
      },
      "@schematics/angular:application": {
        "strict": true
      }
    },
    "root": ""
  }
}
```



# Running project

ng serve

```
› (base) vagner@simaras-MacBook-Pro crud-angular % ng serve

Would you like to share pseudonymous usage data about this project with the Angular Team
at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more
details and how to change this setting, see https://angular.dev/cli/analytics.

(y/N) █
```

# Angular Material



ng add @angular/material

```
i Using package manager: npm
✓ Found compatible package version: @angular/material@18.0.1.
✓ Package information loaded.
```

```
The package @angular/material@18.0.1 will be installed and executed.
Would you like to proceed? (Y/n) y
```

```
? Choose a prebuilt theme name, or "custom" for a custom theme: (Use arrow keys)
❯ Azure/Blue           [Preview: https://material.angular.io?theme=azure-blue]
Rose/Red               [Preview: https://material.angular.io?theme=rose-red]
Magenta/Violet        [Preview: https://material.angular.io?theme=magenta-violet]
Cyan/Orange            [Preview: https://material.angular.io?theme=cyan-orange]
Custom
```

```
? Choose a prebuilt theme name, or "custom" for a custom theme: Azure/Blue
https://material.angular.io?theme=azure-blue
? Set up global Angular Material typography styles? (y/N) y
```

```
? Set up global Angular Material typography styles? Yes
? Include the Angular animations module? (Use arrow keys)
❯ Include and enable animations
Include, but disable animations
Do not include
```



# Angular Material

src/app/app.ts

```
import { MatToolbarModule } from '@angular/material/toolbar';
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet, MatToolbarModule],
  templateUrl: './app.html',
  styleUrls: ['./app.scss'
})
export class App {
  protected readonly title = signal('crud-angular');
}
```





# Angular Material

src/app/app.html

```
<mat-toolbar color="primary">
  <span>CRUD Angular + Spring</span>
</mat-toolbar>
```

src/styles.scss

```
...
.mat-toolbar.mat-primary {
  background-color: #d7e3ff;
  color: #005cb;
}
...
```

ng serve



# Angular CLI

- ng generate module courses --routing **ou** ng g m courses --routing

```
● (base) vagner@simaras-MacBook-Pro crud-angular % ng g m couses --routing
CREATE src/app/couses/couses-routing.module.ts (249 bytes)
CREATE src/app/couses/couses.module.ts (280 bytes)
```

- ng generate component courses/courses **ou** ng g c courses/courses

```
● (base) vagner@simaras-MacBook-Pro crud-angular % ng g c courses/courses
CREATE src/app/courses/courses/courses.component.scss (0 bytes)
CREATE src/app/courses/courses/courses.component.html (22 bytes)
CREATE src/app/courses/courses/courses.component.spec.ts (603 bytes)
CREATE src/app/courses/courses/courses.component.ts (239 bytes)
```



# Angular CLI - Rotas

- src/app/courses/courses-routing.module.ts

```
...
import { Courses } from './courses/courses';

const routes: Routes = [
  { path: "", component: Courses }
];
```



# Angular CLI - Rotas

- src/app/app.routes.ts

```
...
export const routes: Routes = [
  { path: '', pathMatch: 'full', redirectTo: 'courses' },
  {
    path: 'courses',
    loadChildren: () => import('./courses/courses-module').then(m => m.CoursesModule)
  }
];
...
```



# Angular CLI - Rotas

src/app/app.html

```
<mat-toolbar color="primary">
  <span>CRUD Angular + Spring</span>
</mat-toolbar>
<router-outlet></router-outlet>
```

ng serve



# Cards, Toolbar e Table

src/app/courses/courses/courses.ts

```
...
import { MatCardModule } from '@angular/material/card';
import { MatTableModule } from '@angular/material/table';
import { MatToolbarModule } from '@angular/material/toolbar';

@Component({
  selector: 'app-courses',
  standalone: true,
  imports: [ MatTableModule, MatCardModule, MatToolbarModule ],
  templateUrl: './courses.html',
  styleUrls: ['./courses.scss'
})
...
```

<https://material.angular.io/components/table/overview>



# Cards, Toolbar e Table

src/app/courses/courses/courses.html

```
<p>courses works!</p>
<mat-card>
  <mat-toolbar color="primary">Cursos Disponíveis</mat-toolbar>
  <table mat-table [dataSource]="courses" class="mat-elevation-z8">
    <ng-container matColumnDef="name">
      <th mat-header-cell *matHeaderCellDef> Curso </th>
      <td mat-cell *matCellDef="let course"> {{course.name}} </td>
    </ng-container>
    <ng-container matColumnDef="category">
      <th mat-header-cell *matHeaderCellDef> Categoria </th>
      <td mat-cell *matCellDef="let course"> {{course.category}} </td>
    </ng-container>
    <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
    <tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr>
  </table>
</mat-card>
```

<https://material.angular.io/components/table/overview>



# Criar uma interface

```
ng g interface courses/model/course
```

```
● (base) vagner@simaras-MacBook-Pro crud-angular % ng g interface courses/model/course
CREATE src/app/courses/model/course.ts (28 bytes)
```

```
src/app/courses/model/course.ts
```

```
export interface Course {
  _id: string;
  name: string;
  category: string;
}
```



## src/app/courses/courses/courses.ts

```
...
import { Component, OnInit } from '@angular/core';
import { Course } from './model/course';
...
export class Courses implements OnInit {
  courses: Course[] = [{_id: "1", name: "Curso Angular", category: "Front-end"}];
  displayedColumns: string[] = ['name', 'category'];
  dataSource = [];

  constructor(){}
  ngOnInit(): void {}
}
...
...
```

<https://material.angular.io/components/table/overview>



## src/app/courses/courses/courses.html

```
<p>courses works!</p>
<mat-card>
  <mat-toolbar color="primary">Cursos Disponíveis</mat-toolbar>
  <table mat-table [dataSource]="courses" class="mat-elevation-z8">
    <ng-container matColumnDef="name">
      <th mat-header-cell *matHeaderCellDef> Curso </th>
      <td mat-cell *matCellDef="let course"> {{course.name}} </td>
    </ng-container>
    <ng-container matColumnDef="category">
      <th mat-header-cell *matHeaderCellDef> Categoria </th>
      <td mat-cell *matCellDef="let course"> {{course.category}} </td>
    </ng-container>
    <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
    <tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr>
  </table>
</mat-card>
```

<https://material.angular.io/components/table/overview>



src/app/courses/courses/courses.scss

```
table{  
  width: 100%;  
}
```

ng serve

<https://material.angular.io/components/table/overview>



# Serviço

ng generate service courses/services/courses-services **ou** ng g s  
courses/services/courses-services

- (base) vagner@simaras-MacBook-Pro crud-angular % ng g m couses --routing  
CREATE src/app/couses/couses-routing.module.ts (249 bytes)  
CREATE src/app/couses/couses.module.ts (280 bytes)



# Serviço

src/app/courses/services/courses.ts

```
...
import { Course } from './model/course';
...

export class Courses {
  list(): Course[] {
    return [
      {_id: "1", name: "Curso Angular", category: "Front-end"}
    ];
  }
  constructor() { }
}
...
```



# Serviço

src/app/courses/courses/courses.ts

```
...
import { CoursesServices } from './services/courses-services';
...

export class Courses implements OnInit {
  courses: Course[] = [];
  displayedColumns: string[] = ['name', 'category'];
  courseService: CoursesServices;

  constructor(){
    this.courseService = new CoursesServices();
    this.courses = this.courseService.list();
  }
...
}
```

ng serve

<https://material.angular.io/>



# Serviço - Ajax

src/app/courses/courses/courses.ts

```
export class Courses implements OnInit {
  courses: Course[] = [];
  displayedColumns: string[] = ['name', 'category'];
  courseService: CoursesServices;

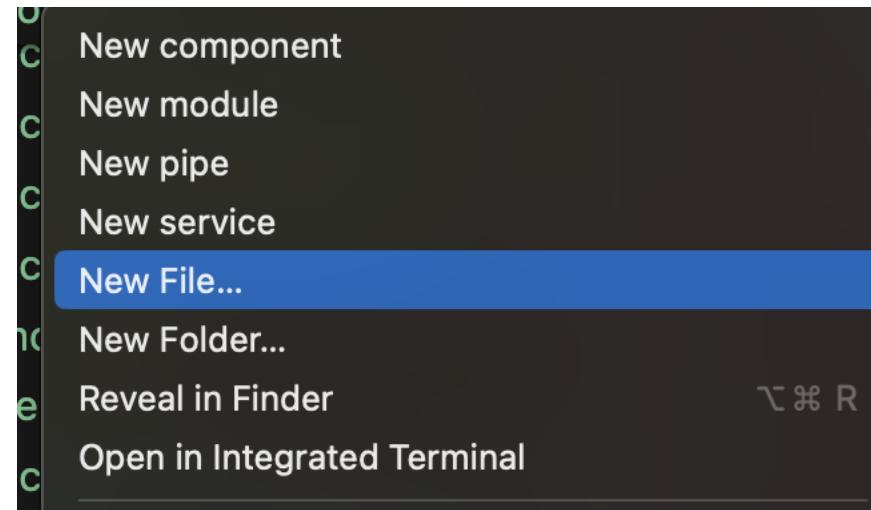
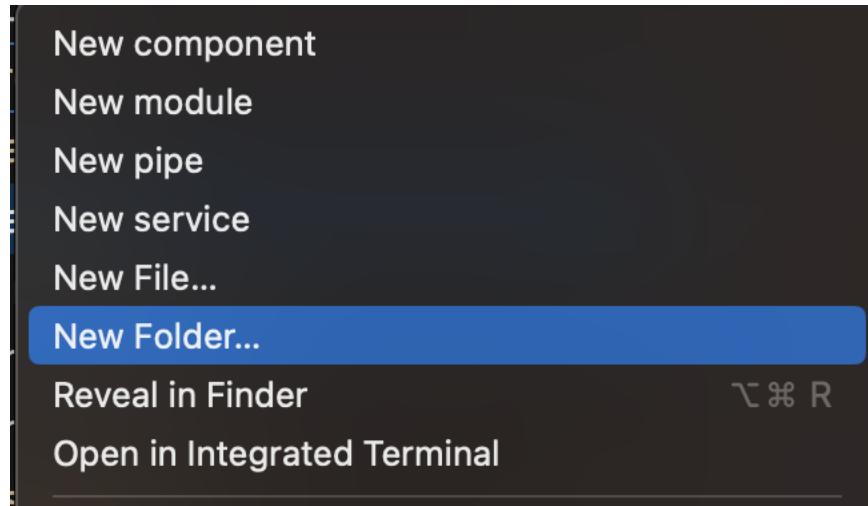
  constructor(private courseService: CoursesServices){
    //this.courseService = new CoursesService();
    //this.courses = this.courseService.list();
  }
  ngOnInit(): void {
    this.courses = this.courseService.list();
  }
}
```

ng serve

# Serviço - Ajax



Criar uma pasta **assets** dentro do **public** e um arquivo **courses.json** dentro da pasta assets





# Serviço - Ajax

public/assets/courses.json

```
[  
  {  
    "_id": "1",  
    "name": "Curso Angular",  
    "category": "Front-end"  
  }  
]
```



# Serviço - Ajax

src/app/app.config.ts

```
...
import { provideHttpClient, withFetch } from '@angular/common/http';

export const appConfig: ApplicationConfig = {
  providers: [
    provideBrowserGlobalErrorListeners(),
    provideRouter(routes), provideClientHydration(withEventReplay()),
    provideHttpClient(withFetch())
  ]
};
```



# Serviço - Ajax

src/app/courses/services/courses-service.ts

```
...
import { first, tap } from 'rxjs';
import { HttpClient } from '@angular/common/http';
...

export class CoursesServices {
  private readonly API = 'assets/courses.json';
  constructor(private httpClient: HttpClient) { }

  list(){
    return this.httpClient.get<Course[]>(this.API)
      .pipe(
        first(),
        tap(courses => console.log(courses))
      );
  }
}
```



# Serviço - Ajax

src/app/courses/courses/courses.ts

```
...
import { Observable } from 'rxjs';
...

export class Courses implements OnInit {
  courses: Observable<Course[]>;
  displayedColumns: string[] = ['name', 'category'];

  constructor(private courseService: CoursesService){
    this.courses = this.courseService.list();
  }
  ngOnInit(): void { }
}
```

ng serve



# Progress Spinner

src/app/courses/courses/courses.ts

```
import { MatProgressSpinnerModule } from '@angular/material/progress-spinner'
import { CommonModule } from '@angular/common';

@Component({
  selector: 'app-courses',
  standalone: true,
  imports: [MatTableModule, MatCardModule, MatToolbarModule,
    CommonModule, MatProgressSpinnerModule],
  templateUrl: './courses.html',
  styleUrls: ['./courses.scss']
})
```



# Progress Spinner

src/app/courses/courses/courses.ts

```
export class Courses implements OnInit {
  courses$: Observable<Course[]>;
  displayedColumns: string[] = ['name', 'category'];

  constructor(private courseService: CoursesServices){
    this.courses$ = this.courseService.list();
  }

  ngOnInit(): void { }
}
```



# Progress Spinner

src/app/courses/courses/courses.html

```
...
<div *ngIf="courses$ | async as courses; else loading">
  <table mat-table [dataSource]="courses" class="mat-elevation-z8">
    <ng-container matColumnDef="name">
      <th mat-header-cell *matHeaderCellDef> Curso </th>
      <td mat-cell *matCellDef="let course"> {{course.name}} </td>
    </ng-container>
    <ng-container matColumnDef="category">
      <th mat-header-cell *matHeaderCellDef> Categoria </th>
      <td mat-cell *matCellDef="let course"> {{course.category}} </td>
    </ng-container>
    <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
    <tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr>
  </table>
</div>
...
```



# Progress Spinner

src/app/courses/courses/courses.html

```
...
<ng-template #loading>
  <div class="loading-spinner">
    <mat-spinner></mat-spinner>
  </div>
</ng-template>
...
```



# Progress Spinner

src/app/courses/services/courses-services.ts

```
...
export class CoursesServices {
  private readonly API = 'assets/courses.json';

  constructor(private httpClient: HttpClient) { }

  list(){
    return this.httpClient.get<Course[]>(this.API)
      .pipe(
        first(),
        delay(10000),
        tap(courses => console.log(courses))
      );
  }
}
```



# Progress Spinner

src/app/courses/courses/courses.scss

```
.loading-spinner{  
  padding: 25px;  
  background: rgba(0, 0, 0, 0.15);  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

ng serve



# Tratamento de Erros

ng g m shared

```
(base) vagner@simaras-MacBook-Pro crud-angular % ng g m shared
CREATE src/app/shared/shared.module.ts (192 bytes)
```

ng g c shared/components/error-dialog

```
(base) vagner@simaras-MacBook-Pro crud-angular % ng g c shared/components/error-dialog
CREATE src/app/shared/components/error-dialog/error-dialog.component.scss (0 bytes)
CREATE src/app/shared/components/error-dialog/error-dialog.component.html (27 bytes)
CREATE src/app/shared/components/error-dialog/error-dialog.component.spec.ts (632 bytes)
CREATE src/app/shared/components/error-dialog/error-dialog.component.ts (258 bytes)
```



# Tratamento de Erros

src/app/shared/components/error-dialog/error-dialog.ts

```
import { Component, Inject } from '@angular/core';
import { MatButtonModule } from '@angular/material/button';
import { MAT_DIALOG_DATA, MatDialogContent,
  MatDialogModule, MatDialogTitle } from '@angular/material/dialog';

@Component({
  selector: 'app-error-dialog',
  standalone: true,
  imports: [MatDialogModule, MatDialogTitle,
    MatDialogContent, MatButtonModule],
  templateUrl: './error-dialog.component.html',
  styleUrls: ['./error-dialog.component.scss'
})

export class ErrorDialog {
  constructor(@Inject(MAT_DIALOG_DATA) public data: string){}
}
```



# Tratamento de Erros

src/app/shared/components/error-dialog/error-dialog.html

```
<h3 mat-dialog-title style="color: red;">Error!</h3>
<mat-dialog-content>{{ data }}</mat-dialog-content>
<mat-dialog-actions>
  <button mat-stroked-button mat-dialog-close>Fechar</button>
</mat-dialog-actions>
```



# Tratamento de Erros

src/app/courses/courses.module.ts

```
...
import { MatDialog } from '../shared/components/error-dialog/error-dialog';

@NgModule({
  declarations: [],
  imports: [
    CommonModule,
    CoursesRoutingModule,
    ErrorDialogComponent
  ]
})
...
...
```



# Tratamento de Erros

src/app/courses/courses/courses.ts

```
...
import { catchError, Observable, of } from 'rxjs';
import { MatButtonModule } from '@angular/material/button';
import { MatDialog, MatDialogModule } from '@angular/material/dialog';
import { ErrorDialog } from '../../shared/components/error-dialog/error-dialog';

@Component({
  selector: 'app-courses',
  standalone: true,
  imports: [MatTableModule, MatCardModule, MatToolbarModule,
    CommonModule, MatProgressSpinnerModule, MatDialogModule,
    MatButtonModule],
  templateUrl: './courses.html',
  styleUrls: ['./courses.scss']
})
...
```



# Tratamento de Erros

src/app/courses/courses/courses.ts

```
...
export class Courses implements OnInit {
  ...
  constructor(
    private courseService: CoursesServices,
    public dialog: MatDialog
  ) {
    this.courses$ = this.courseService.list()
      .pipe(
        catchError(error => {
          this.onError('Erro ao carregar a lista de cursos');
          return of([]);
        })
      );
  }
  ...
}
```





# Tratamento de Erros

src/app/courses/courses/courses.ts

```
...
export class Courses implements OnInit {
  ...
  onError(errorMsg: string){
    this.dialog.open(ErrorDialog, {
      data: errorMsg,
    })
  }
  ngOnInit(): void { }
}
```

ng serve



# Customizando as Categorias

ng g pipe shared/pipes/category

```
(base) vagner@simaras-MacBook-Pro crud-angular % ng g pipe shared/pipes/category
CREATE src/app/shared/pipes/category.pipe.spec.ts (195 bytes)
CREATE src/app/shared/pipes/category.pipe.ts (241 bytes)
```



# Customizando as Categorias

src/app/shared/pipes/category.pipe.ts

```
...
export class CategoryPipe implements PipeTransform {

  transform(value: string): string {
    switch(value){
      case 'Front-end' : return 'code'
      case 'Back-end' : return 'computer'
    }
    return 'code';
  }
}
```



# Customizando as Categorias

src/app/courses/courses/courses.ts

```
...
import { MatIconModule } from '@angular/material/icon';
import { CategoryPipe } from '../../shared/pipes/category-pipe';

@Component({
  selector: 'app-courses',
  standalone: true,
  imports: [MatTableModule, MatCardModule, MatToolbarModule,
    CommonModule, MatProgressSpinnerModule, MatDialogModule,
    MatButtonModule, MatIconModule, CategoryPipe],
  templateUrl: './courses.html',
  styleUrls: ['./courses.scss']
})
...
```



# Customizando as Categorias

src/app/courses/courses/courses.html

```
...
<ng-container matColumnDef="category">
  <th mat-header-cell *matHeaderCellDef> Categoria </th>
  <td mat-cell *matCellDef="let course">{{ course.category }}<br/>
    <mat-icon aria-hidden="false" aria-label="Categoria do Curso">
      {{ course.category | category }}</mat-icon>
    </td>
</ng-container>
...
```

# Windows e MacOS - Java JDK – 22.0.1



<https://www.oracle.com/br/java/technologies/downloads/>

Product/file description	File size	Download
x64 Compressed Archive	184.14 MB	<a href="https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.zip">https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.zip</a> (sha256)
x64 Installer	164.31 MB	<a href="https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.exe">https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.exe</a> (sha256)
x64 MSI Installer	163.06 MB	<a href="https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.msi">https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.msi</a> (sha256)



Product/file description	File size	Download
ARM64 Compressed Archive	180.03 MB	<a href="https://download.oracle.com/java/22/latest/jdk-22_macos-aarch64_bin.tar.gz">https://download.oracle.com/java/22/latest/jdk-22_macos-aarch64_bin.tar.gz</a> (sha256)
ARM64 DMG Installer	179.47 MB	<a href="https://download.oracle.com/java/22/latest/jdk-22_macos-aarch64_bin.dmg">https://download.oracle.com/java/22/latest/jdk-22_macos-aarch64_bin.dmg</a> (sha256)
x64 Compressed Archive	182.21 MB	<a href="https://download.oracle.com/java/22/latest/jdk-22_macos-x64_bin.tar.gz">https://download.oracle.com/java/22/latest/jdk-22_macos-x64_bin.tar.gz</a> (sha256)
x64 DMG Installer	181.65 MB	<a href="https://download.oracle.com/java/22/latest/jdk-22_macos-x64_bin.dmg">https://download.oracle.com/java/22/latest/jdk-22_macos-x64_bin.dmg</a> (sha256)

A Instalação do  
java no Linux foi  
feito no slide 4



Mac OS





# Maven Apache

<https://maven.apache.org/download.cgi>

## Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

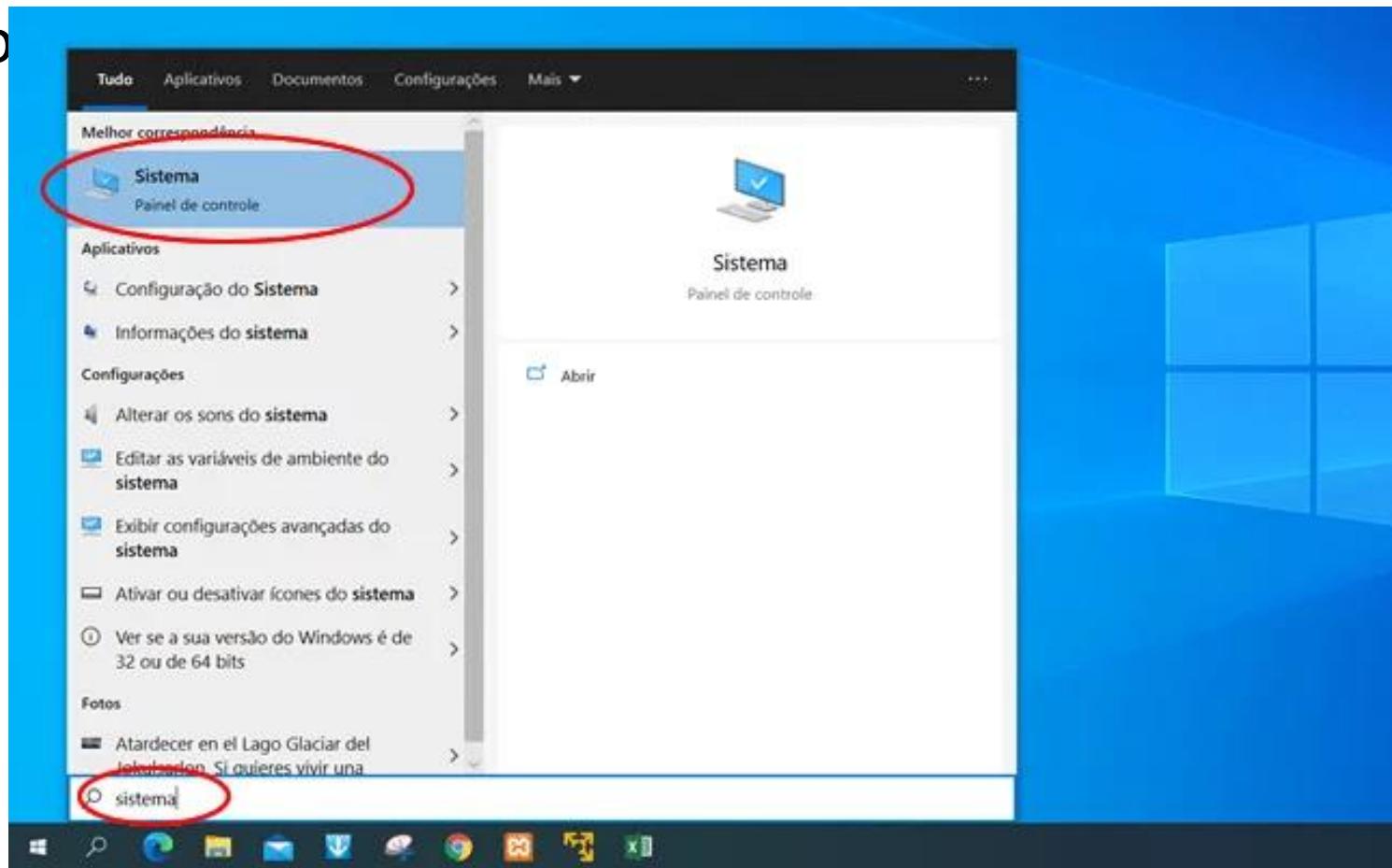


Link	Checksums	Signature
Binary tar.gz archive <a href="#">apache-maven-3.9.9-bin.tar.gz</a>	<a href="#">apache-maven-3.9.9-bin.tar.gz.sha512</a>	<a href="#">apache-maven-3.9.9-bin.tar.gz.asc</a>
Binary zip archive <a href="#">apache-maven-3.9.9-bin.zip</a>	<a href="#">apache-maven-3.9.9-bin.zip.sha512</a>	<a href="#">apache-maven-3.9.9-bin.zip.asc</a>
Source tar.gz archive <a href="#">apache-maven-3.9.9-src.tar.gz</a>	<a href="#">apache-maven-3.9.9-src.tar.gz.sha512</a>	<a href="#">apache-maven-3.9.9-src.tar.gz.asc</a>
Source zip archive <a href="#">apache-maven-3.9.9-src.zip</a>	<a href="#">apache-maven-3.9.9-src.zip.sha512</a>	<a href="#">apache-maven-3.9.9-src.zip.asc</a>

- [3.9.9 Release Notes and Release Reference Documentation](#)
- [latest source code from source repository](#)
- Distributed under the [Apache License, version 2.0](#)
- other:
  - All current release sources (plugins, shared libraries,...) available at <https://downloads.apache.org/maven/>

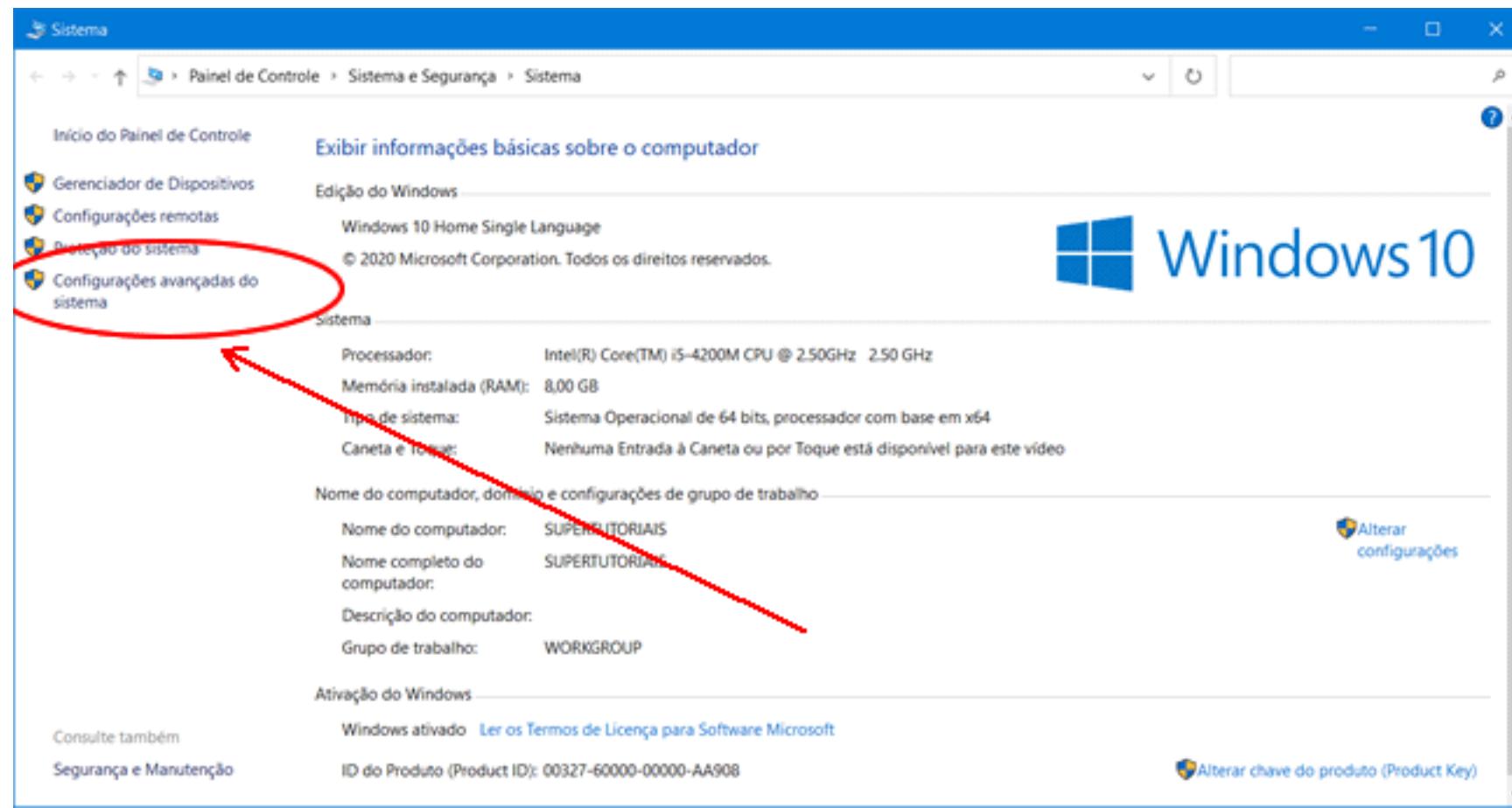
# Variáveis de Ambiente - Windows

Abra o Menu Iniciar, digite Sistema e abra a opção “Sistema” do Painel de Controle.



# Variáveis de Ambiente - Windows

Clique em “Configurações avançadas do sistema” do lado esquerdo da janela.

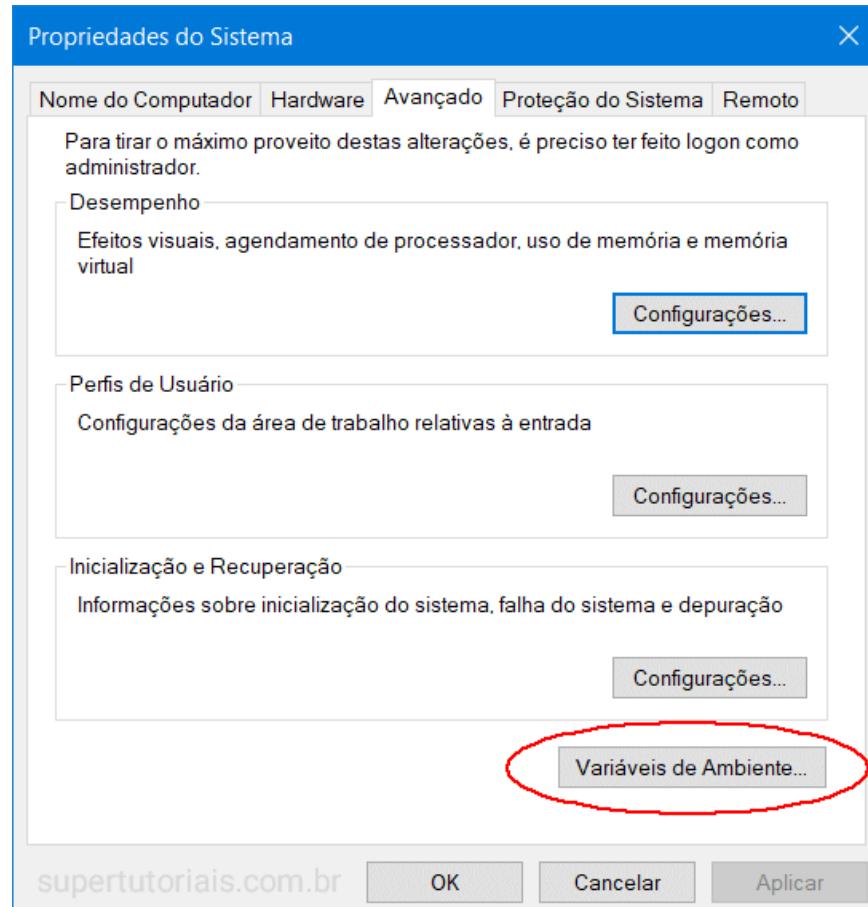


Maven™



# Variáveis de Ambiente - Windows

Na guia “Avançado”, clique em “Variáveis de Ambiente...”.

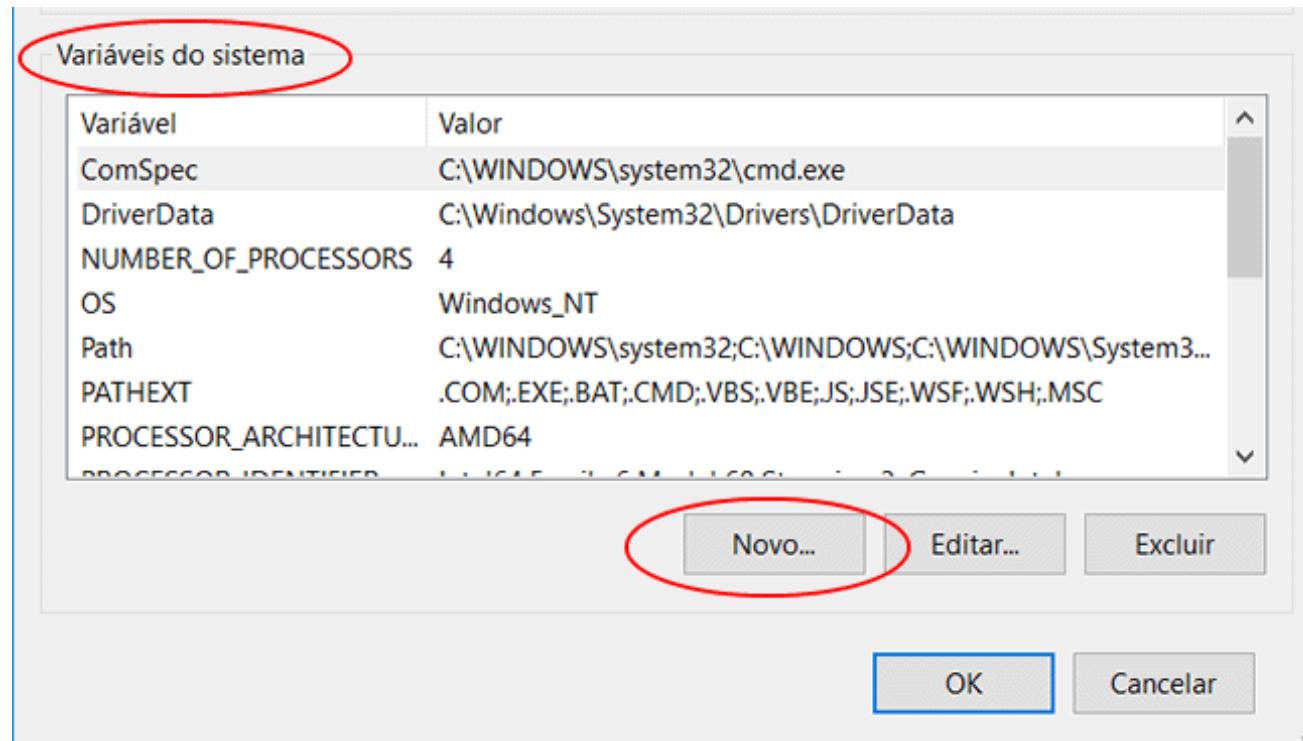


Maven™



# Variáveis de Ambiente - Windows

Nas “Variáveis do Sistema” clique em “Novo...”.



Maven™



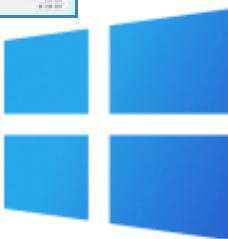
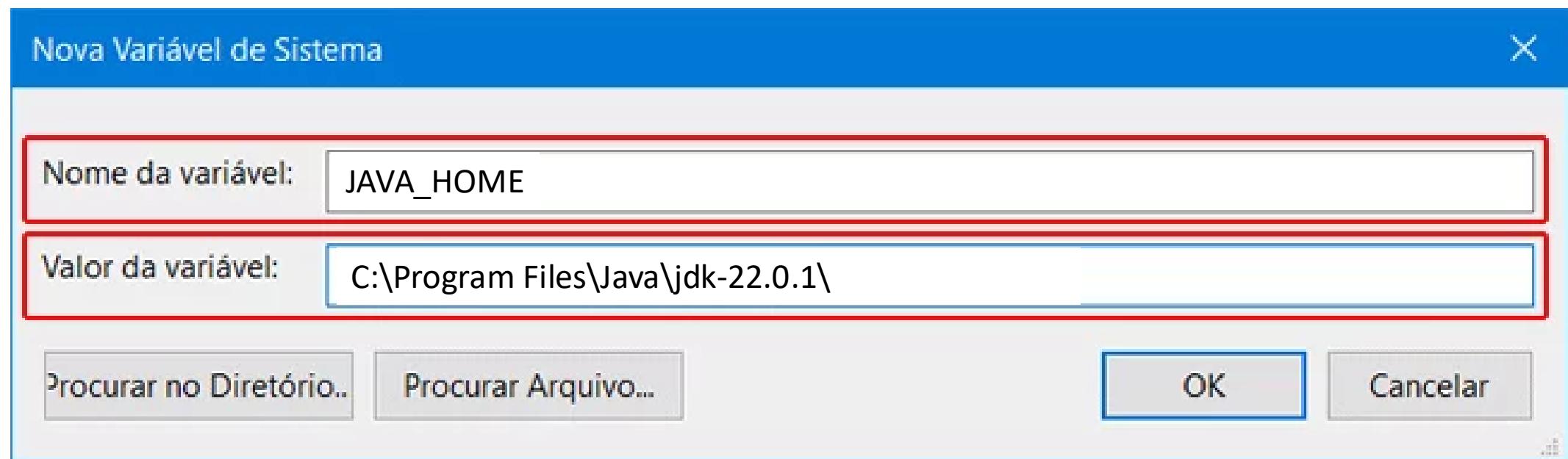
Java™





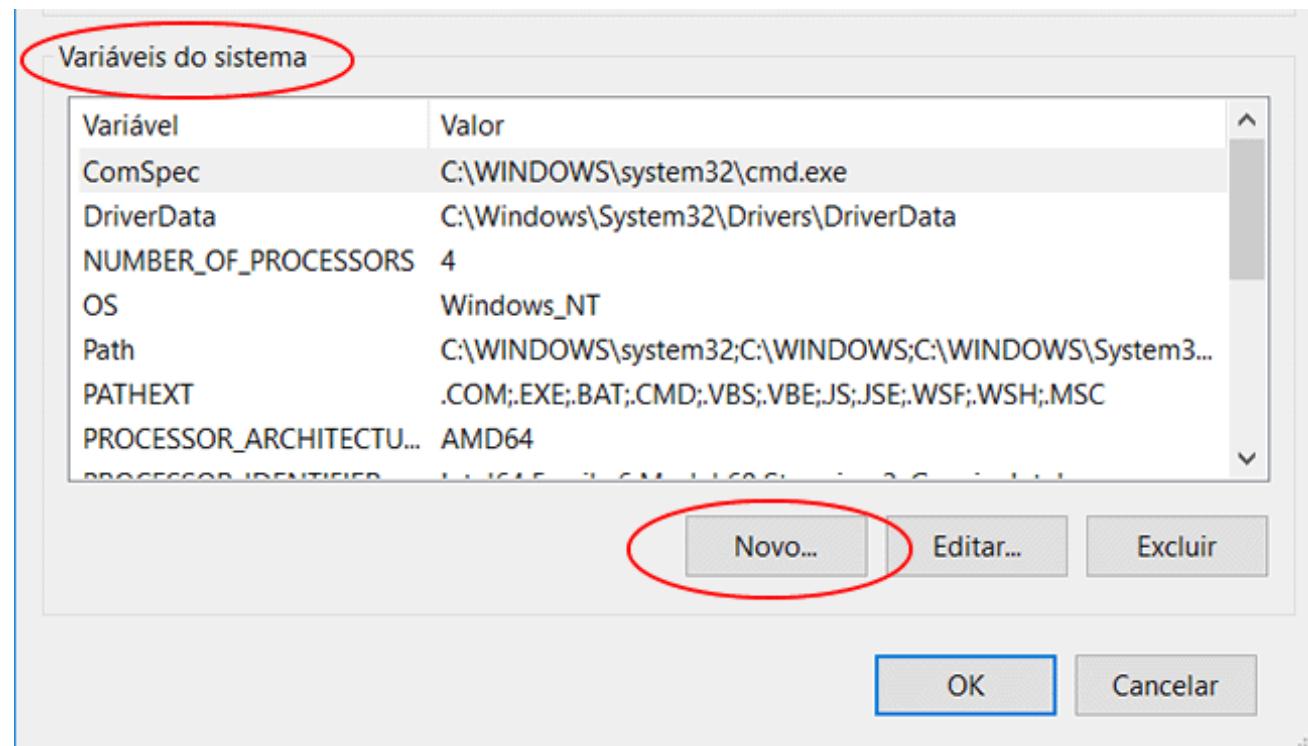
# Variáveis de Ambiente - Windows

Na janela aberta, digite o nome da variável JAVA\_HOME e, em “Valor da variável”, adicione o caminho da pasta do JDK.



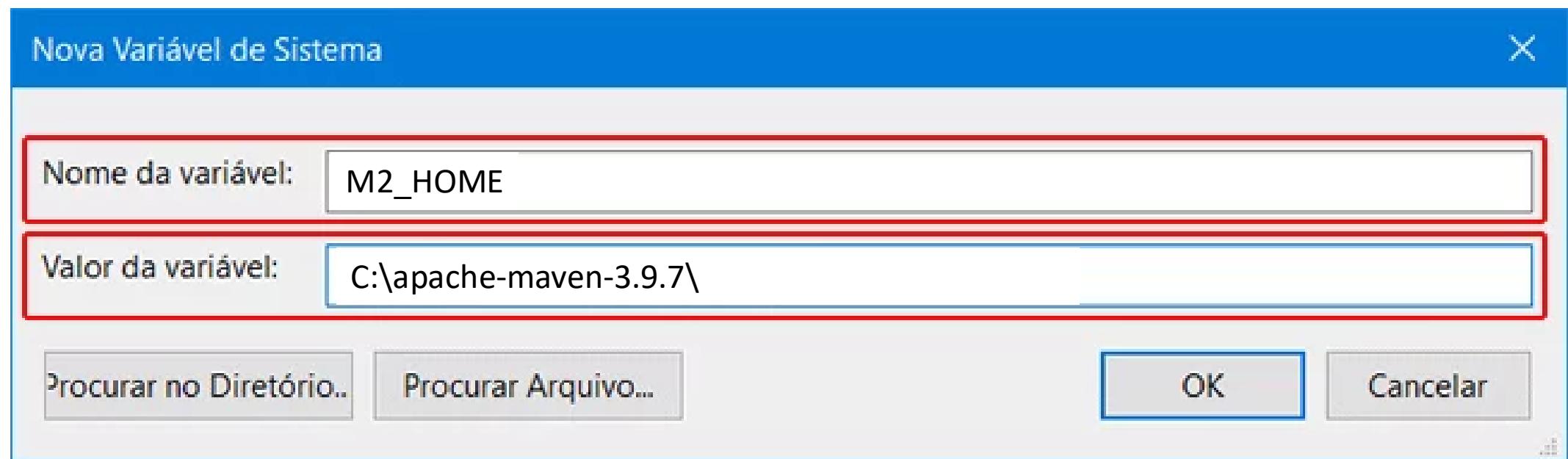
## Variáveis de Ambiente - Windows

Nas “Variáveis do Sistema” clique em “Novo...”.



# Variáveis de Ambiente - Windows

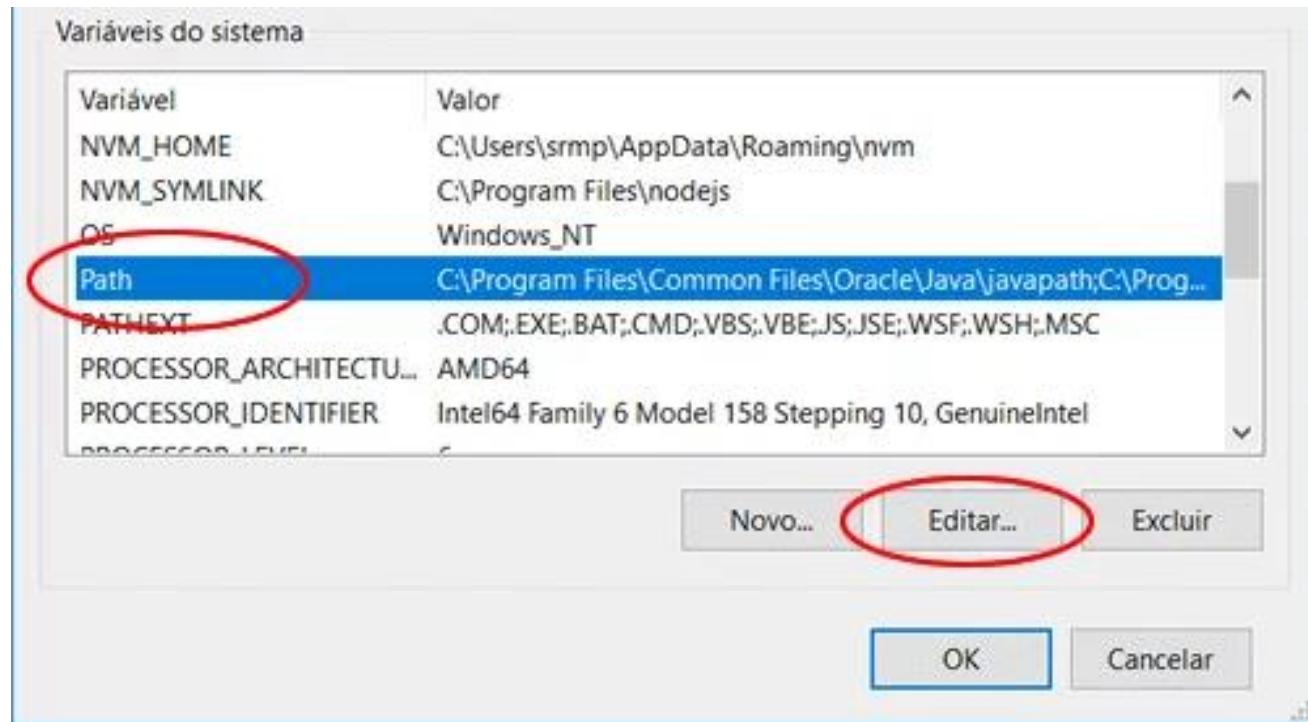
Na janela aberta, digite o nome da variável M2\_HOME e, em “Valor da variável”, adicione o caminho da pasta do Maven Apache.





# Variáveis de Ambiente - Windows

Em seguida, vamos adicionar o caminho da pasta do programa na variável “Path”.



# Variáveis de Ambiente

Na janela aberta, clique em “Novo”, insira o caminho do Java Home com a pasta “BIN” do JAVA

Exemplo:

%JAVA\_HOME%\bin



Variáveis de Ambiente

Variáveis de usuário para srmp

Editar a variável de ambiente

C:\Windows\System32\Wbem  
C:\Windows\System32\WindowsPowerShell\v1.0\  
C:\Windows\System32\OpenSSH\  
C:\Program Files (x86)\Intel\Intel(R) Management Engine Co...  
C:\Program Files\Intel\Intel(R) Management Engine Compon...  
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common  
%SystemRoot%\system32  
%SystemRoot%  
%SystemRoot%\System32\Wbem  
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\  
%SYSTEMROOT%\System32\OpenSSH\  
C:\Program Files\Java\jdk-11.0.8\bin  
C:\Program Files (x86)\Brackets\command  
C:\Program Files\dotnet\  
C:\Program Files\Java\jdk-11.0.11\bin  
C:\Program Files\PuTTY\  
%NVM\_HOME%  
%NVM\_SYMLINK%  
C:\Program Files (x86)\dotnet\  
C:\Program Files\Git\cmd  
%JAVA\_HOME%\bin

Variáveis do sistema

Variável NVM\_HOME NVM\_SYMLINK OS Path PATHEXT PROCESSOR\_ARCHITECTURE PROCESSOR\_IDENTIFIER PROCESSOR\_LEVEL PROCESSOR\_REVISION PROCESSOR\_IDENTIFIER PROCESSOR\_LEVEL PROCESSOR\_REVISION

Novo Editar Procurar... Excluir Mover para Cima Mover para baixo Editar texto... OK Cancelar

A interface de usuário mostra uma lista de variáveis de ambiente existentes no sistema. Um novo item, "%JAVA\_HOME%\bin", é inserido na lista e está circulado em vermelho. O botão "Novo" na barra lateral também está circulado em vermelho, indicando que é esse o passo que deve ser feito.

# Variáveis de Ambiente

Na janela aberta, clique em “Novo”, insira o caminho do Maven Home com a pasta “BIN” do Apache Maven

Exemplo:

%M2\_HOME%\bin



Variáveis de Ambiente

Variáveis de usuário para srmp

Editar a variável de ambiente

Variável EMAIL kryptor NVM\_HOME NVM\_SYMLINK OneDrive Path POWERSHELL... Variáveis do sistema Variável NVM\_HOME NVM\_SYMLINK OS Path PATHEXT PROCESSOR PROCESSOR... %SystemRoot%\\system32 %SystemRoot% %SystemRoot%\\System32\\Wbem %SYSTEMROOT%\\System32\\WindowsPowerShell\\v1.0\\ %SYSTEMROOT%\\System32\\OpenSSH\\ C:\\Program Files (x86)\\Intel\\Intel(R) Management Engine Components\\ Intel(R) Management Engine Component\\ C:\\Program Files\\Intel\\Intel(R) Management Engine Components\\ Intel(R) ME Tools\\ C:\\Program Files (x86)\\NVIDIA Corporation\\PhysX\\Common\\ %SystemRoot%\\system32 %SystemRoot% %SystemRoot%\\System32\\Wbem %SYSTEMROOT%\\System32\\WindowsPowerShell\\v1.0\\ %SYSTEMROOT%\\System32\\OpenSSH\\ C:\\Program Files\\Java\\jdk-11.0.8\\bin C:\\Program Files (x86)\\Brackets\\command C:\\Program Files\\dotnet\\ C:\\Program Files\\Java\\jdk-11.0.11\\bin C:\\Program Files\\PuTTY\\ %NVM\_HOME% %NVM\_SYMLINK% C:\\Program Files (x86)\\dotnet\\ C:\\Program Files\\Git\\cmd %M2\_HOME%\\bin

Novo Editar Procurar... Excluir Mover para Cima Mover para baixo Editar texto...

OK Cancelar

A interface de usuário mostra uma lista de variáveis de ambiente existentes no sistema. Um novo item, "%M2\_HOME%\\bin", é destacado com um círculo vermelho. O botão "Novo" na barra lateral também está circulado em vermelho, indicando que é o próximo passo para adicionar essa variável.

# Linux – instalando o Maven



```
uzip apache-maven-3.9.9-bin.zip
```

```
sudo mv apache-maven-3.9.9 /opt/
```

```
gedit ~/.bashrc
```

```
export PATH="$PATH:/opt/apache-maven-3.9.9/bin"
```

```
source ~/.bashrc
```



# Java

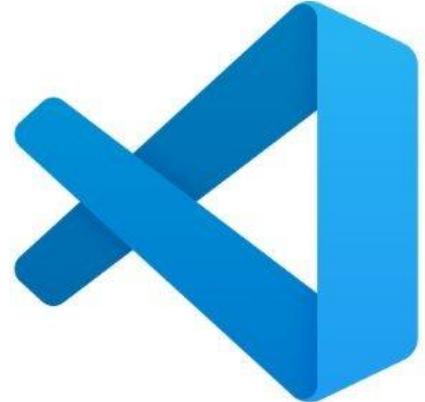
The screenshot shows the Java extension page in the Visual Studio Code Marketplace. At the top, there's a large Java logo icon. To its right, the extension name "Java" is displayed in a blue box with the version "v22.0.1". Below this, the developer information "Oracle Corporation" and the URL "oracle.com" are shown, along with the download count "306,013" and a 5-star rating "(14)". A "Disable" button with a dropdown arrow and an "Uninstall" button with a dropdown arrow are also present. Below these buttons is a gear icon.

Below the header, there are three navigation links: "DETAILS", "FEATURES", and "CHANGELOG". The "DETAILS" link is underlined, indicating it's the active tab.

The main content area is titled "Java Platform Extension for Visual Studio Code". It contains a brief description: "Java Platform extension from Oracle brings full featured development support (edit-compile-debug & test cycle) to VS Code. It offers support for Maven and Gradle projects." Below this description is a section titled "Getting Started" with two numbered steps:

1. Set JDK in View | Command Palette | Preferences:Open User Settings (JSON) ... `jdk: Jdkhome` setting to point to JDK which Language Server will run on and projects will be compiled with. More below in section Selecting the JDK
2. If no JDK is present in your system then extension can setup things for you. More below

To the right of the main content, there's a sidebar with sections for "Categories" and "Resources". The "Categories" section lists "Programming Languages", "Snippets", "Linters", "Debuggers", "Formatters", and "Testing". The "Resources" section lists "Marketplace", "Issues", and "Repository".



<https://marketplace.visualstudio.com/items?itemName=Oracle.oracle-java>

# Extension Pack for Java + Spring

The screenshot shows the Visual Studio Marketplace page for the "Extension Pack for Java + Spring" extension. The page has a dark theme with white text. At the top, there's a large icon of a computer monitor displaying a Java interface. Below the icon, the extension name "Extension Pack for Java + Spring" is displayed along with its version "v0.2.3". It also shows the author "Loiane Groner" and the URL "loiane.com", the download count "97,980", and a rating of "★★★★★ (6)". A brief description follows: "Some of the most popular and useful Java and Spring extensions". Below this, there are buttons for "Disable" and "Uninstall" and a gear icon for settings. Navigation links for "DETAILS", "FEATURES", and "CHANGELOG" are also present. The main content area is titled "Extension Pack (7)" and lists three extensions: "Extension Pack for Java", "Docker", and "REST Client". Each item includes a small icon, a title, a brief description, the Microsoft logo, and a gear icon. To the right, there are sections for "Categories" (with "Extension Packs" highlighted), "Resources" (listing "Marketplace", "Issues", "Repository", "License", and "Loiane Groner"), and "More Info".



<https://marketplace.visualstudio.com/items?itemName=loiane.java-spring-extension-pack>

# Spring Boot Extension Pack

The screenshot shows the Visual Studio Code Marketplace page for the "Spring Boot Extension Pack". The page has a dark theme. At the top, there's a large green gear icon with a leaf inside it. Below the icon, the extension name "Spring Boot Extension Pack" is displayed along with its version "v0.2.1", developer "VMware", download count "2,393,295", and a 5-star rating "(18)". A brief description follows: "A collection of extensions for developing Spring Boot applications". There are "Disable" and "Uninstall" buttons, and a settings gear icon. Below this, there are two tabs: "DETAILS" and "FEATURES", with "FEATURES" being the active tab. Under "Extension Pack (3)", there's a listing for "Spring Boot Tools" which provides validation and content assist for Spring Boot applications. It's developed by VMware and includes a settings gear icon. To the right, there are sections for "Categories" (Programming Languages, Linters, Extension Packs) and "Resources" (Marketplace Issues, Repository, VMware). At the bottom, it says "(also known as Spring Tools 4 for Visual Studio Code)".



<https://marketplace.visualstudio.com/items?itemName=vmware.vscode-boot-dev-pack>

# Thunder Client

The screenshot shows the Thunder Client extension page on the Visual Studio Marketplace. At the top, there's a purple circular icon with a white lightning bolt symbol. To its right, the title "Thunder Client" is displayed in bold, followed by "v2.24.11". Below the title, the URL "thunderclient.com" is shown with a GitHub icon, and the download count "3,950,430" with a download icon. A star rating of "★★★★★ (35)" is also present. The description "Lightweight Rest API Client for VS Code" follows. Below this, there are buttons for "Disable" and "Uninstall", and a gear icon. A horizontal navigation bar at the bottom includes "DETAILS" (which is underlined), "FEATURES", and "CHANGELOG".

**Thunder Client**

Thunder Client is a lightweight Rest API Client Extension for VS Code, hand-crafted by [Ranga Vadhineni](#) with a focus on **simplicity, clean design and local storage**.

- Featured on [Product Hunt](#)
- Website - [www.thunderclient.com](http://www.thunderclient.com)
- Documentation: [docs.thunderclient.com](http://docs.thunderclient.com)
- Support: [github.com/rangav/thunder-client-support](https://github.com/rangav/thunder-client-support)

**Story behind Thunder Client**

- Read Launch Blog Post on [Medium](#)

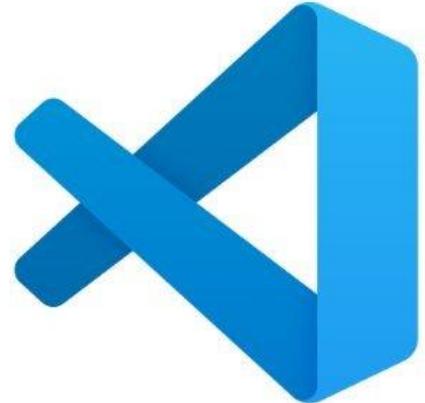
**Categories**

Programming Languages  
Snippets Testing

**Resources**

Marketplace  
Issues  
Repository  
License  
Thunder Client

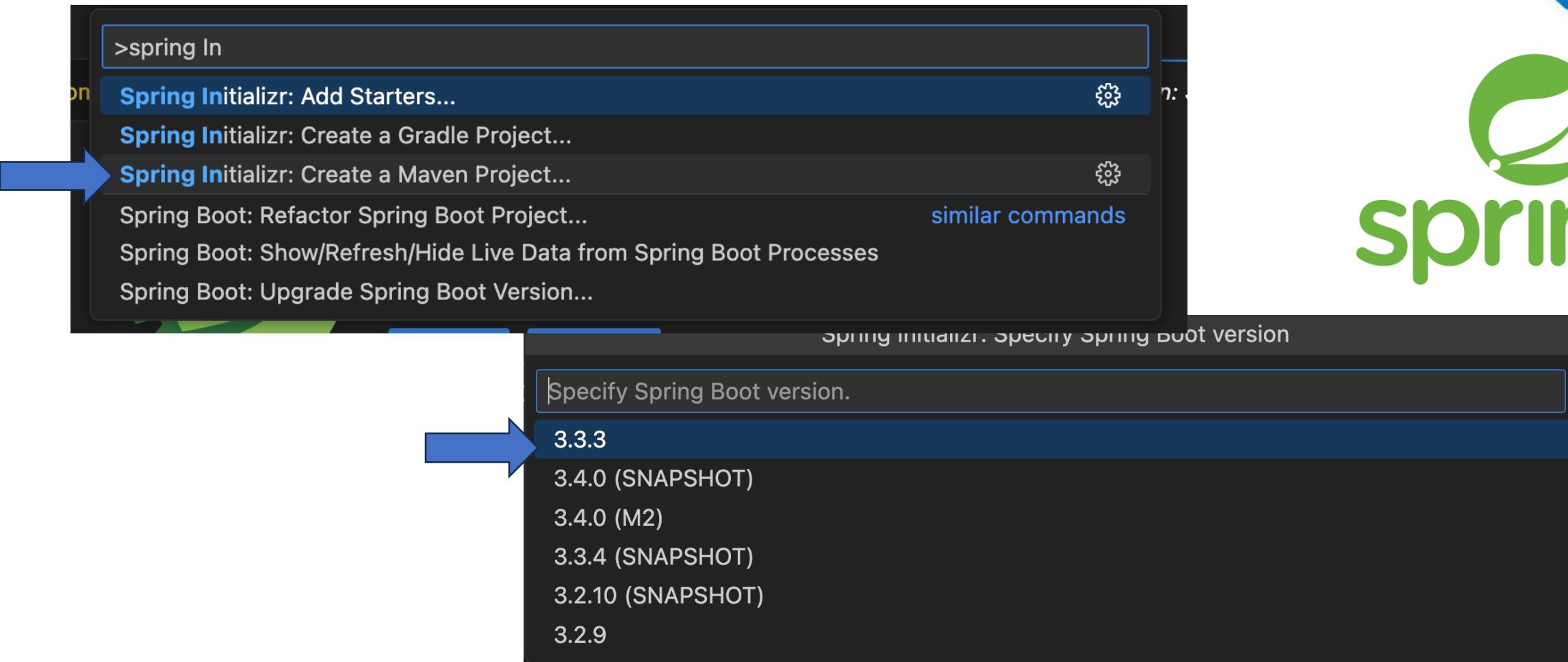
More Info



<https://marketplace.visualstudio.com/items?itemName=rangav.vscode-thunder-client>

# Criando o Projeto Back-End

Ctrl + Shift + P



>spring In

Spring Initializr: Add Starters...

Spring Initializr: Create a Gradle Project...

**Spring Initializr: Create a Maven Project...**

Spring Boot: Refactor Spring Boot Project... similar commands

Spring Boot: Show/Refresh/Hide Live Data from Spring Boot Processes

Spring Boot: Upgrade Spring Boot Version...

Spring Initializr. Specify Spring Boot version

Specify Spring Boot version.

**3.3.3**

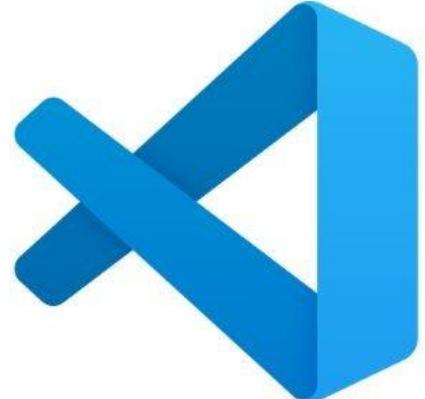
3.4.0 (SNAPSHOT)

3.4.0 (M2)

3.3.4 (SNAPSHOT)

3.2.10 (SNAPSHOT)

3.2.9



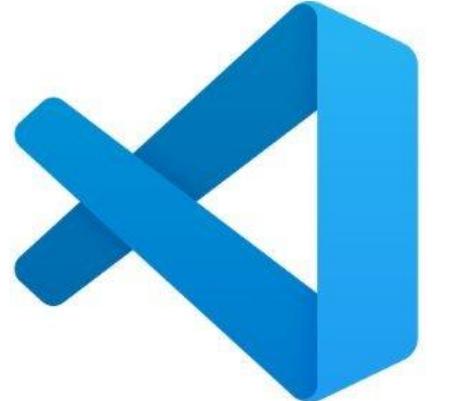
spring

# Criando o Projeto Back-End (Cont...)

Ctrl + Shift + P

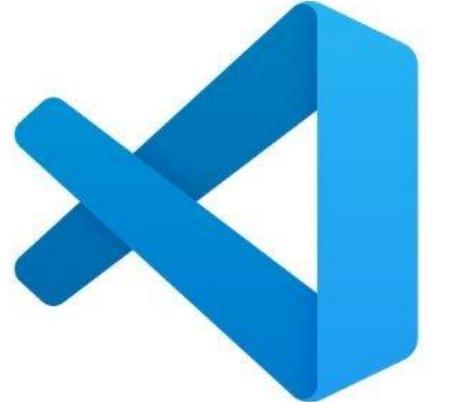
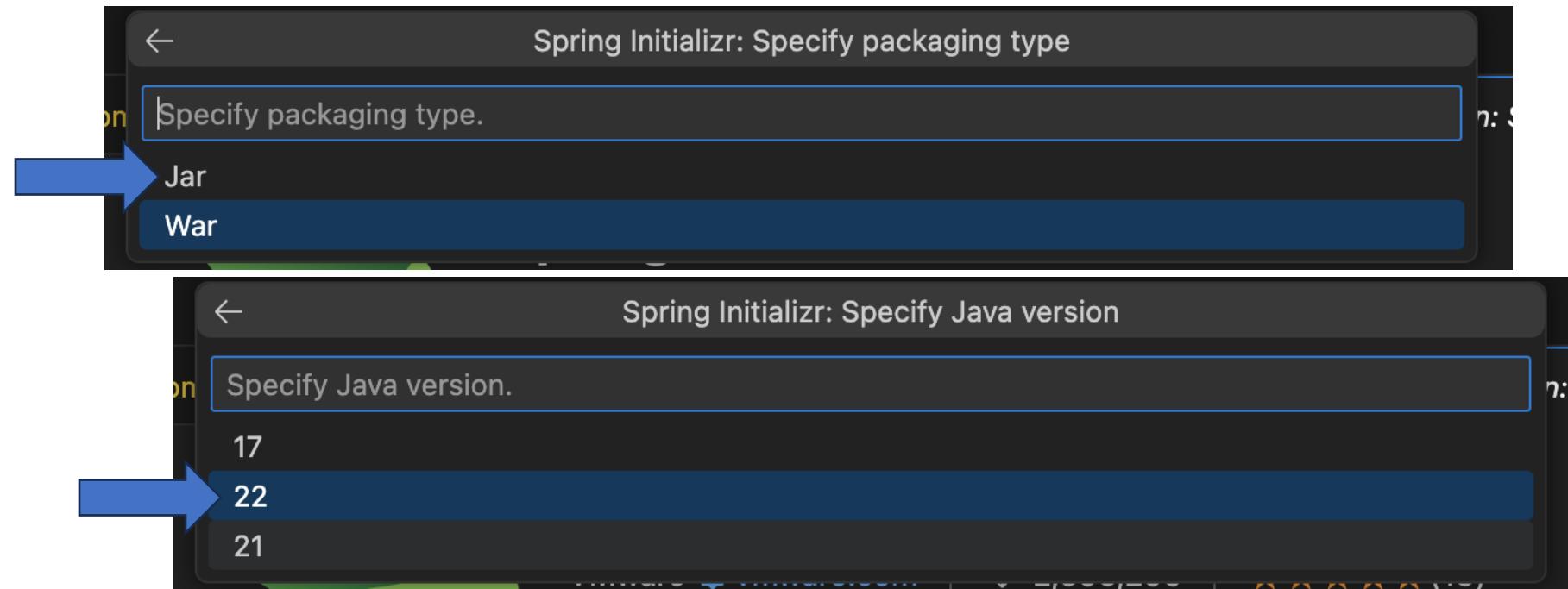
The image consists of three vertically stacked screenshots of the Spring Initializr interface, each with a blue arrow pointing from left to right indicating the flow of the process.

- Top Screenshot:** The title bar says "Spring Initializr: Specify project language". A dropdown menu is open with the following options: "Specify project language.", "Java" (which is highlighted in blue), "Kotlin", and "Groovy".
- Middle Screenshot:** The title bar says "Spring Initializr: Input Group Id". An input field contains the text "com.uneb.labweb". Below it is the placeholder text "Input Group Id for your project. (Press 'Enter' to confirm or 'Escape' to cancel)".
- Bottom Screenshot:** The title bar says "Spring Initializr: Input Artifact Id". An input field contains the text "spring-project". Below it is the placeholder text "Input Artifact Id for your project. (Press 'Enter' to confirm or 'Escape' to cancel)".



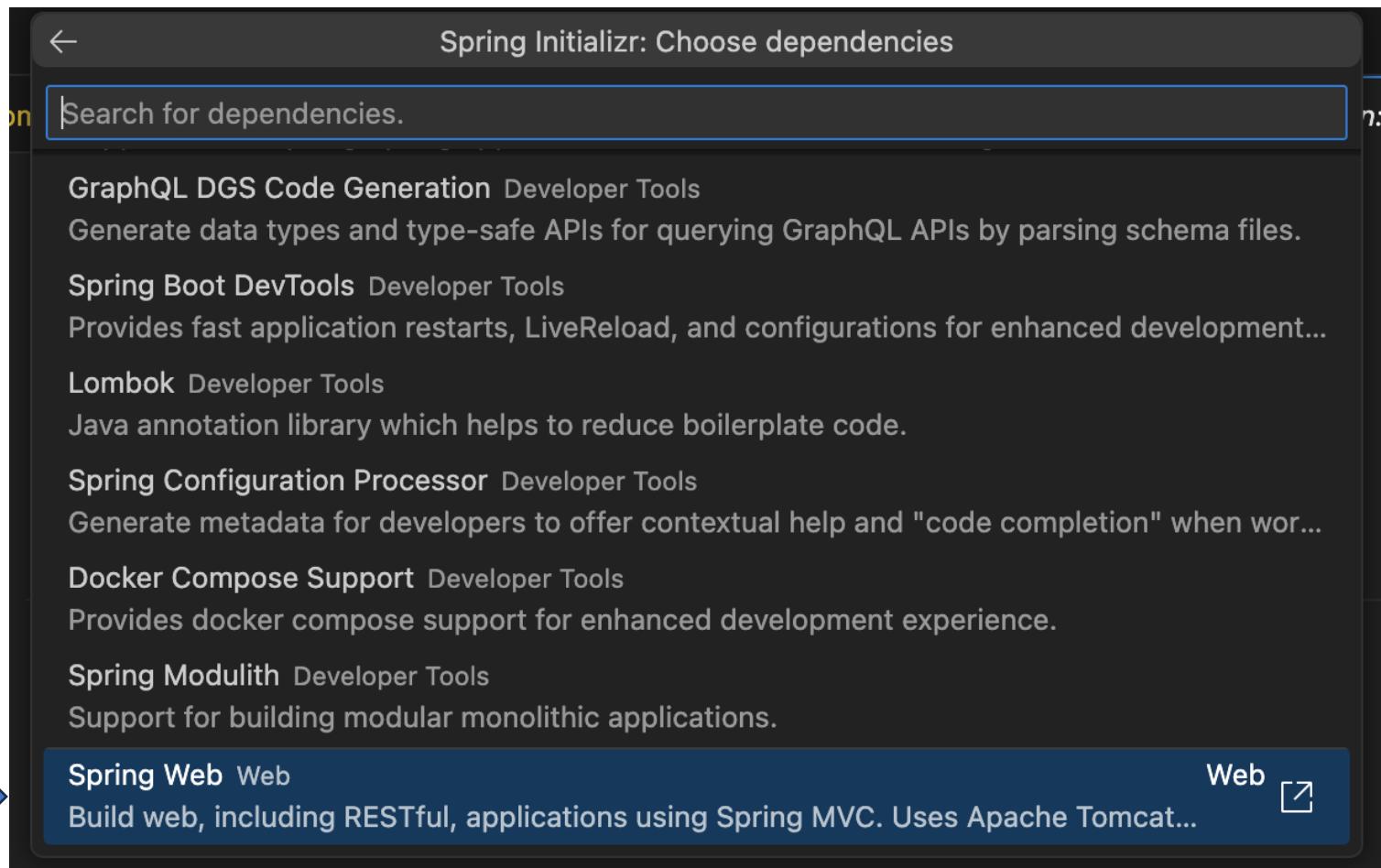
# Criando o Projeto Back-End (Cont...)

Ctrl + Shift + P



# Criando o Projeto Back-End (Cont...)

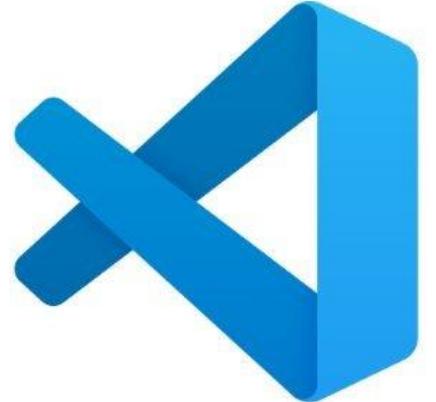
Ctrl + Shift + P



Spring Initializr: Choose dependencies

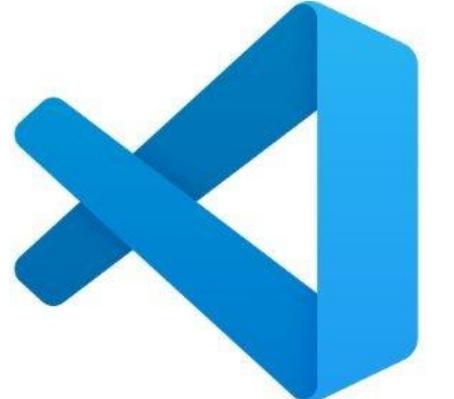
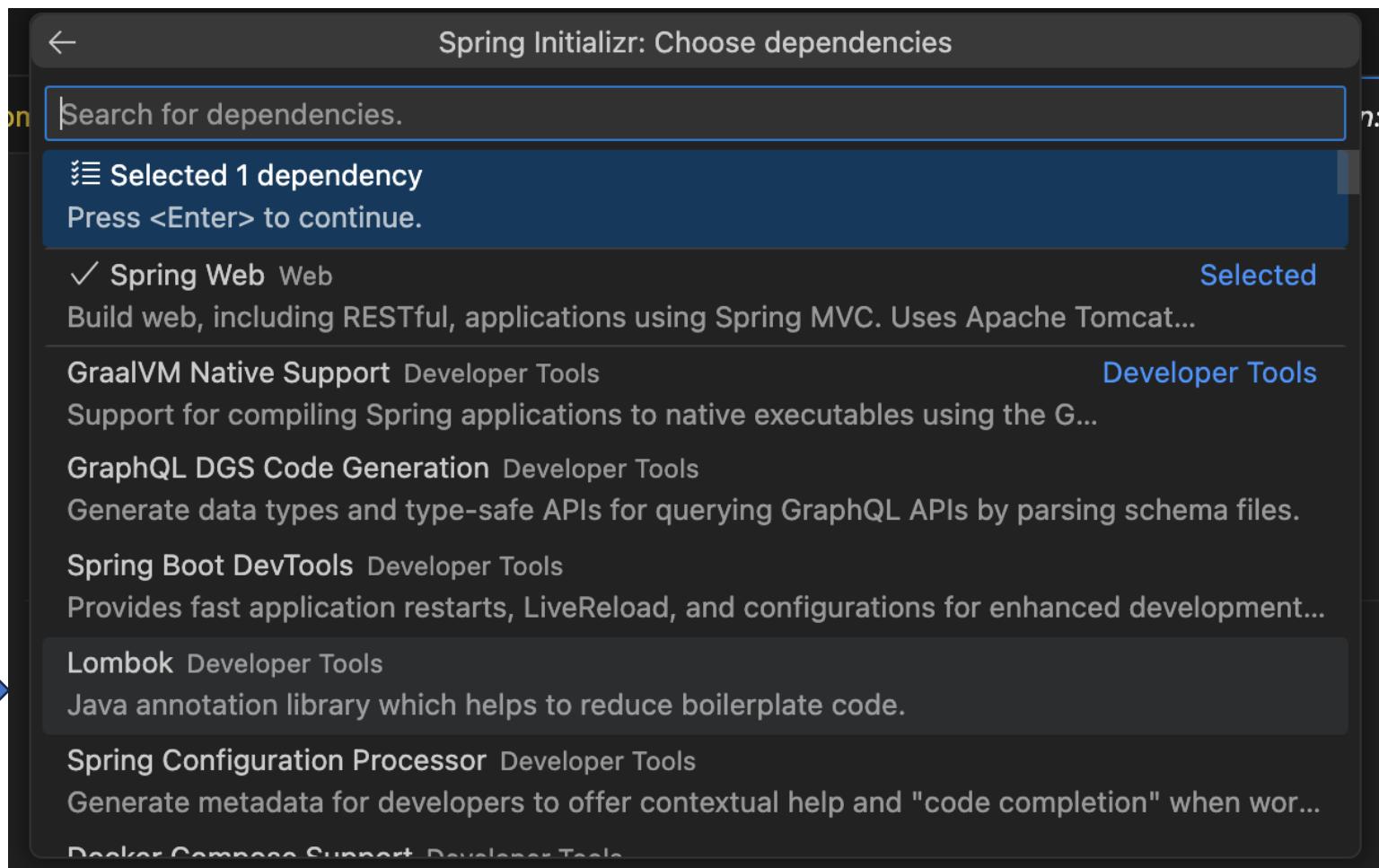
Search for dependencies.

- GraphQL DGS Code Generation Developer Tools  
Generate data types and type-safe APIs for querying GraphQL APIs by parsing schema files.
- Spring Boot DevTools Developer Tools  
Provides fast application restarts, LiveReload, and configurations for enhanced development...
- Lombok Developer Tools  
Java annotation library which helps to reduce boilerplate code.
- Spring Configuration Processor Developer Tools  
Generate metadata for developers to offer contextual help and "code completion" when wor...
- Docker Compose Support Developer Tools  
Provides docker compose support for enhanced development experience.
- Spring Modulith Developer Tools  
Support for building modular monolithic applications.
- Spring Web Web** Web   
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat...



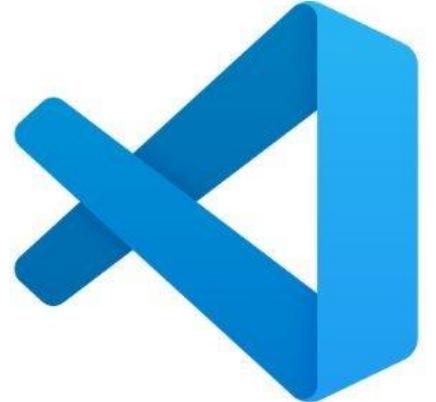
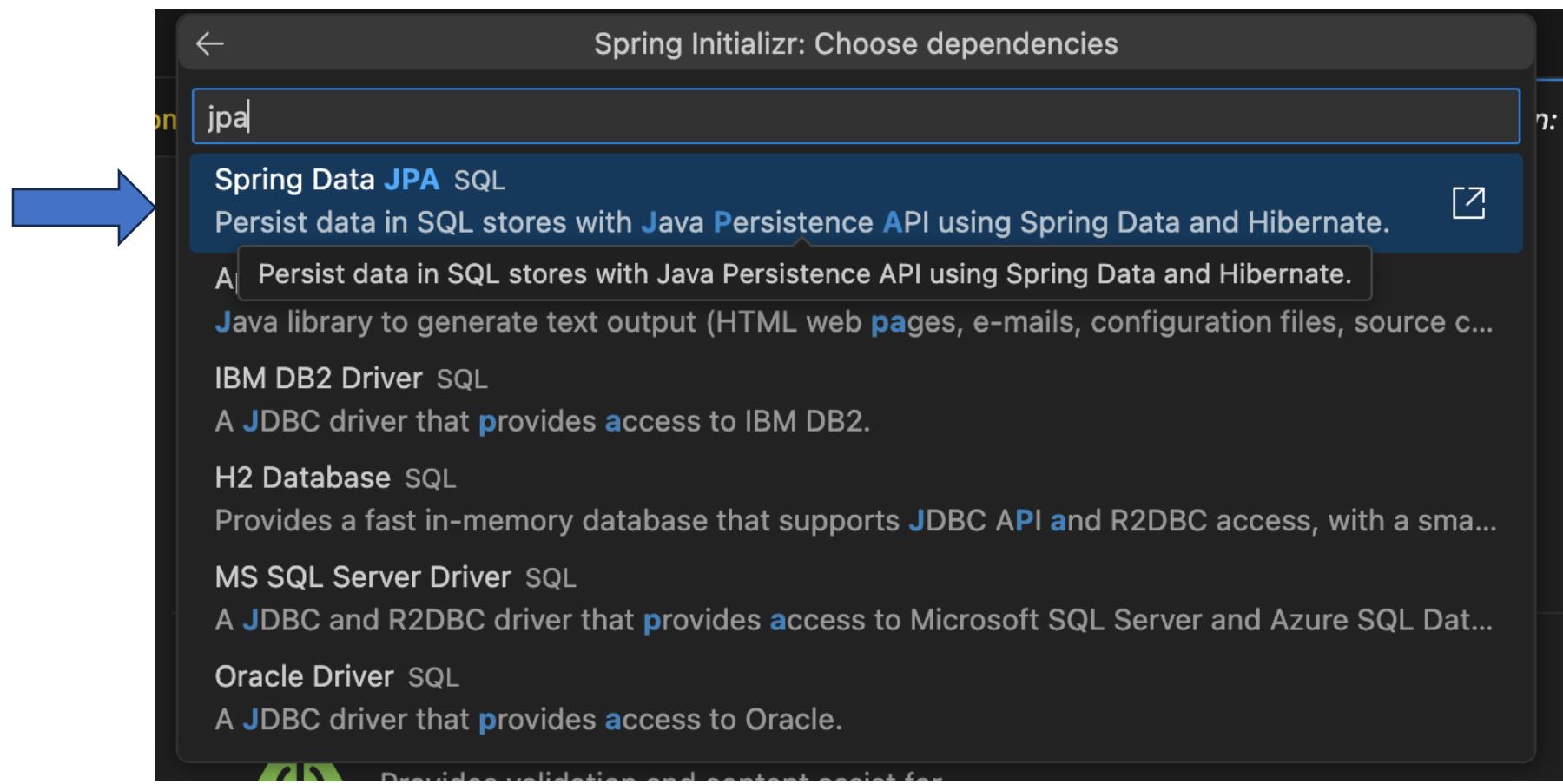
# Criando o Projeto Back-End (Cont...)

Ctrl + Shift + P



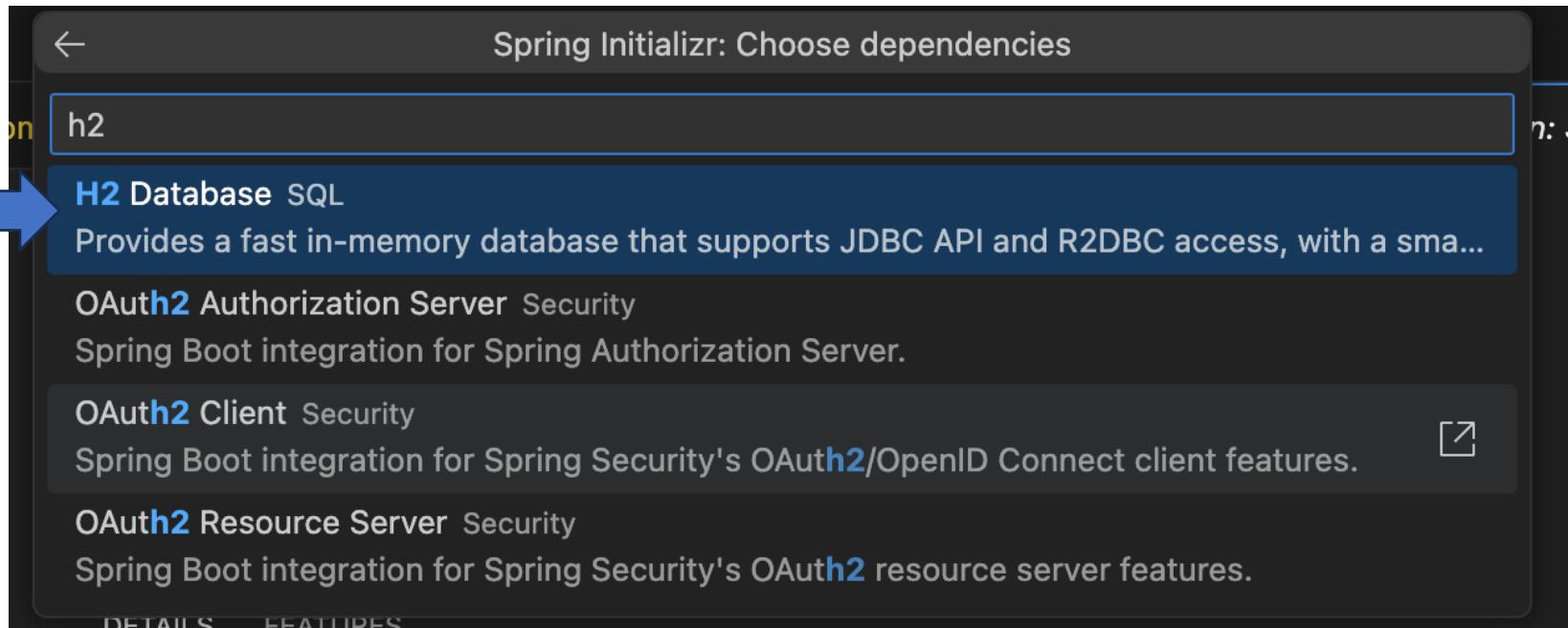
# Criando o Projeto Back-End (Cont...)

Ctrl + Shift + P



# Criando o Projeto Back-End (Cont...)

Ctrl + Shift + P



The screenshot shows the Spring Initializr interface with two dependency selection panels. The top panel has the search bar set to 'h2' and lists the 'H2 Database SQL' dependency, which is highlighted with a blue arrow. The bottom panel has the search bar set to 'spring boot de' and lists the 'Spring Boot DevTools' dependency, also highlighted with a blue arrow.

Spring Initializr: Choose dependencies

h2

**H2 Database SQL**  
Provides a fast in-memory database that supports JDBC API and R2DBC access, with a sma...

OAuth2 Authorization Server Security  
Spring Boot integration for Spring Authorization Server.

OAuth2 Client Security  
Spring Boot integration for Spring Security's OAuth2/OpenID Connect client features.

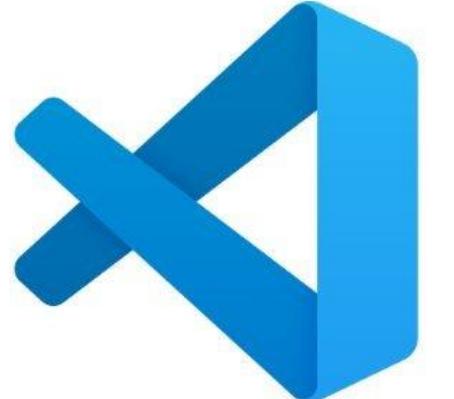
OAuth2 Resource Server Security  
Spring Boot integration for Spring Security's OAuth2 resource server features.

DETAILS FEATURES

Spring Initializr: Choose dependencies

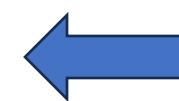
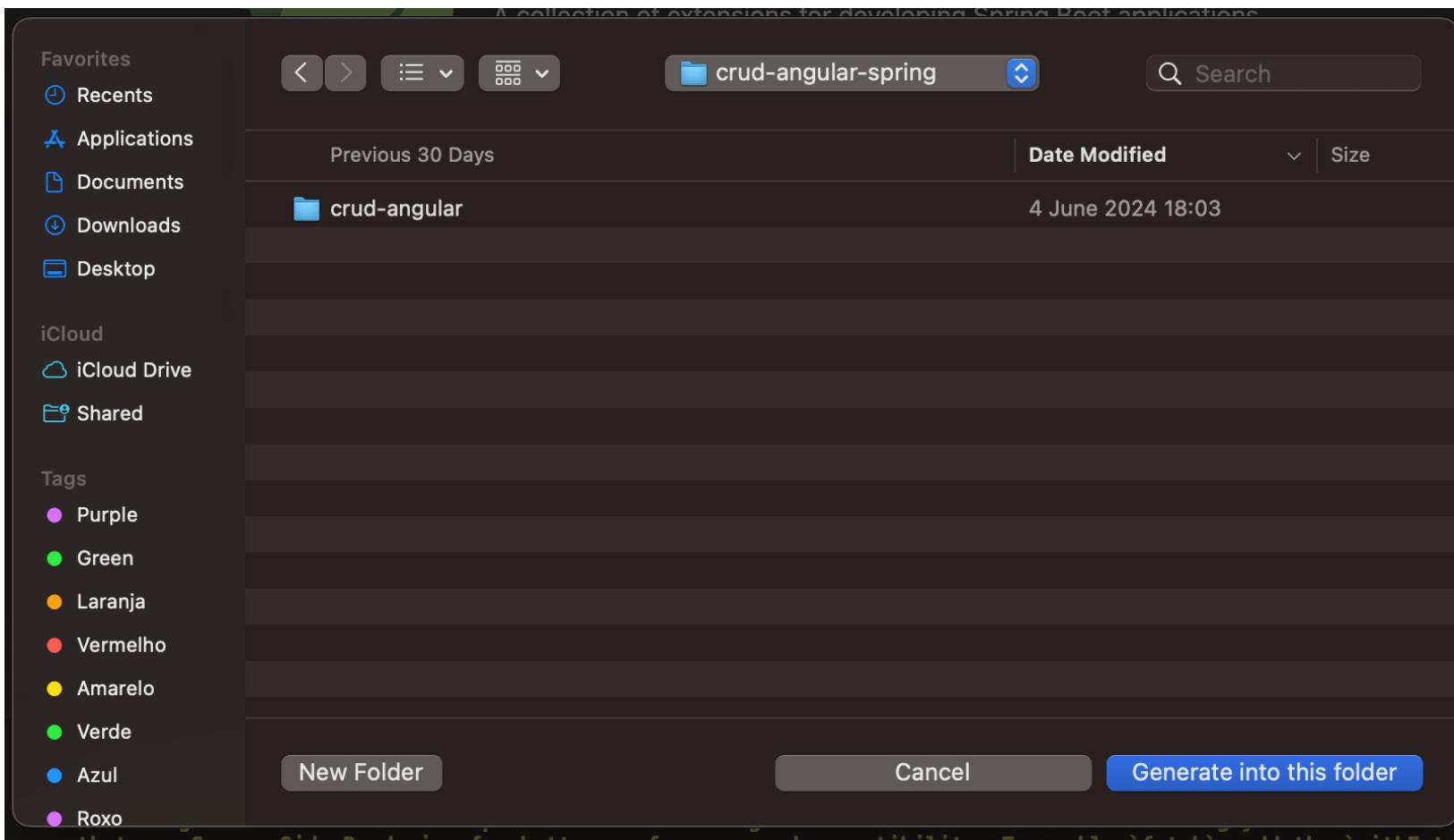
spring boot de

**Spring Boot DevTools** Developer Tools  
Provides fast application restarts, LiveReload, and configurations for enhanced develop...



# Criando o Projeto Back-End (Cont...)

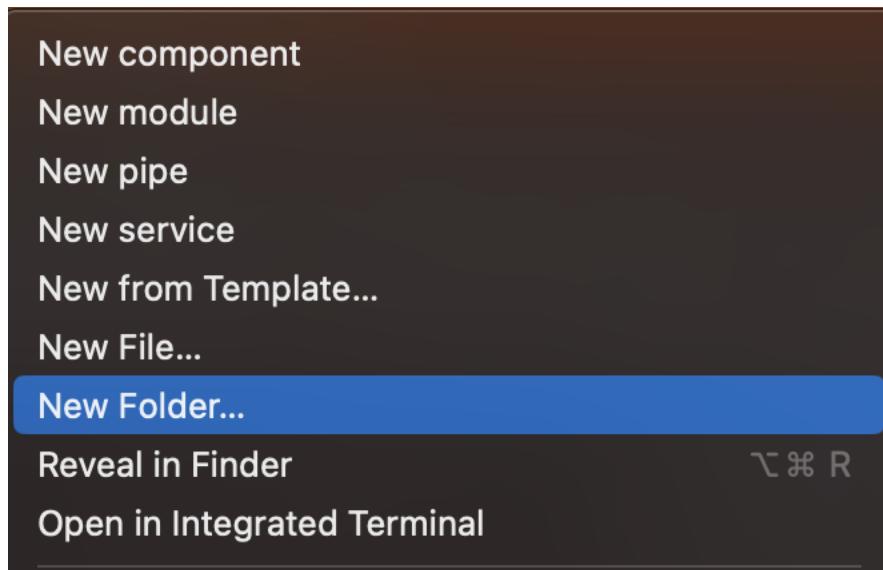
Ctrl + Shift + P



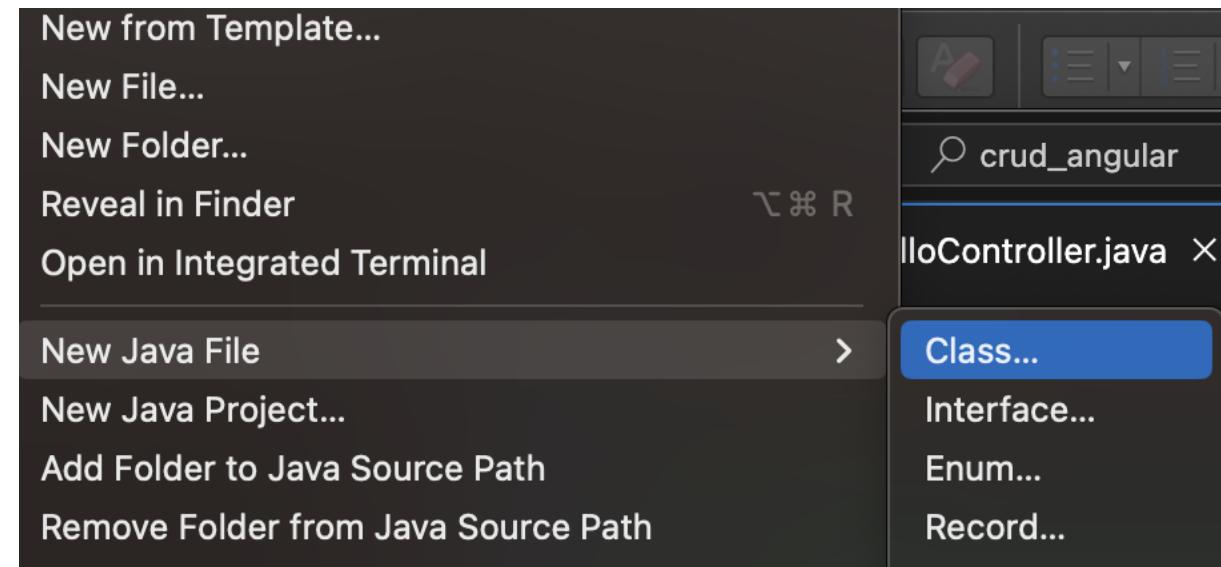


# Hello World no Spring

## Criar uma pasta controller



## Criar uma arquivo HelloController.java





# Hello World no Spring

src/main/java/com/uneb/labweb/spring\_project/controller/HelloController.java

```
package com.uneb.labweb.spring_project.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/api/hello")
public class HelloController {
    @GetMapping
    public String hello() {
        return "Hello!";
    }
}
```



# Hello World no Spring

src/main/java/com/uneb/labweb/spring\_project/SpringProjectApplication.java

```
package com.lp3.crud-angular;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class CrudAngularApplication {

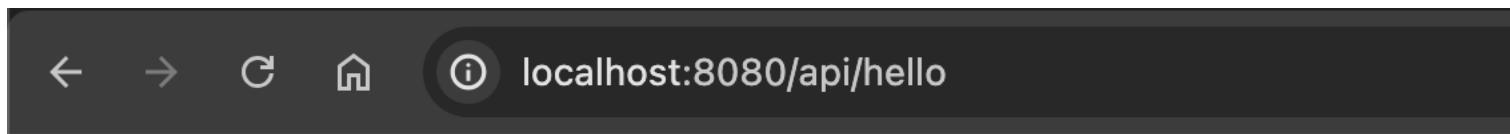
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        SpringApplication.run(CrudAngularApplication.class, args);
    }

}
```



# Hello World no Spring

`http://localhost:8080/api/hello`



Hello!



# Configurando API

Criar a classe CourseController.java dentro do controller

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
package com.uneb.labweb.spring_project.controller;

import java.util.List;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/api/courses")
public class CourseController {

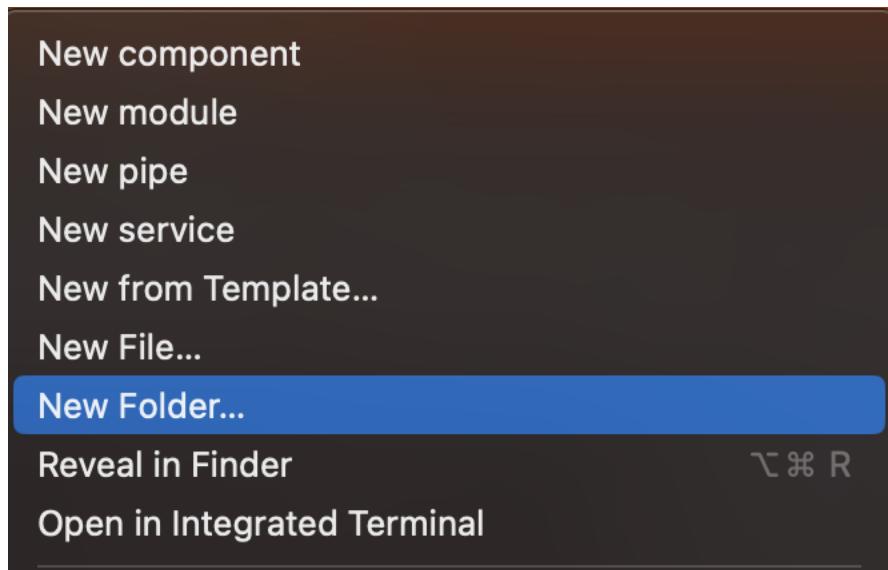
    @GetMapping
    public List<Object> list(){
        return null;
    }
}
```

# Configurando API

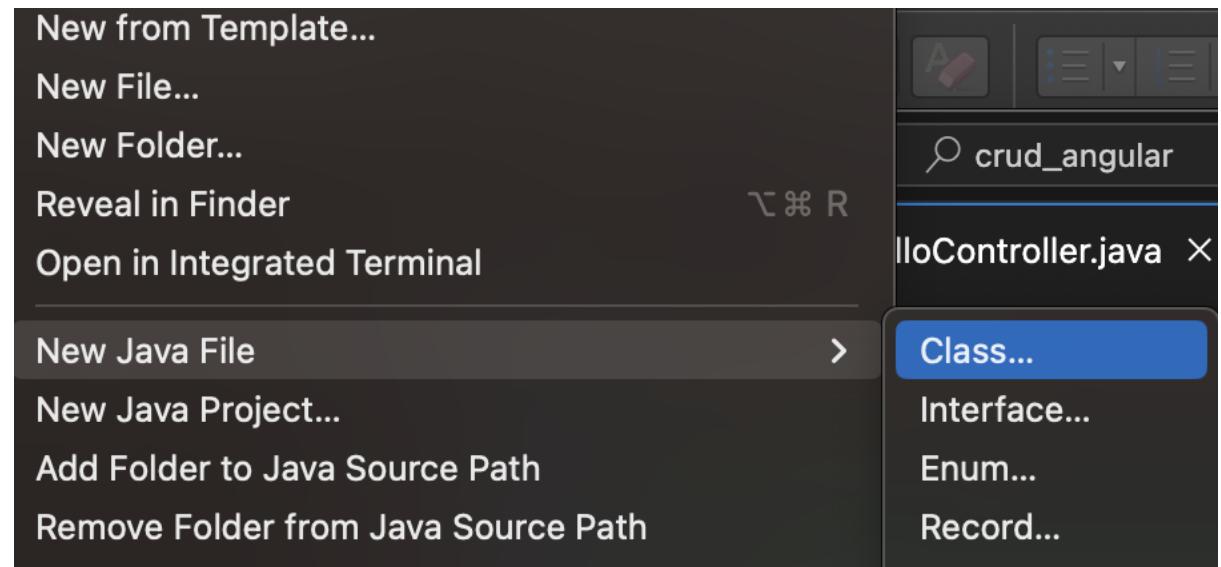


spring

## Criar uma pasta model



## Criar uma arquivo Course.java



# Configurando API

Criar a classe Course.java dentro do model

src/main/java/com/uneb/lab  
web/spring\_project/model/  
Course.java



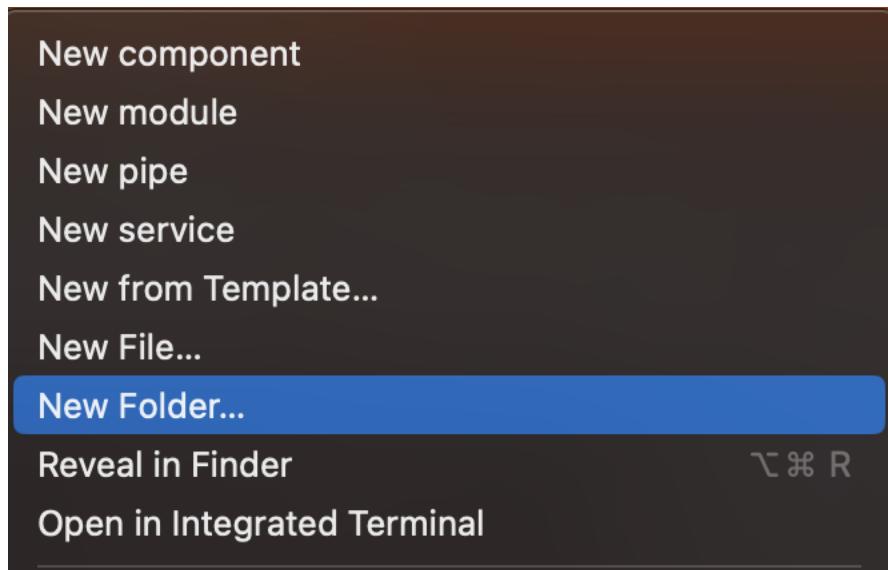
```
package com.uneb.labweb.spring_project.model;
```

```
import jakarta.persistence.Column;  
import jakarta.persistence.Entity;  
import jakarta.persistence.GeneratedValue;  
import jakarta.persistence.GenerationType;  
import jakarta.persistence.Id;  
import lombok.Data;
```

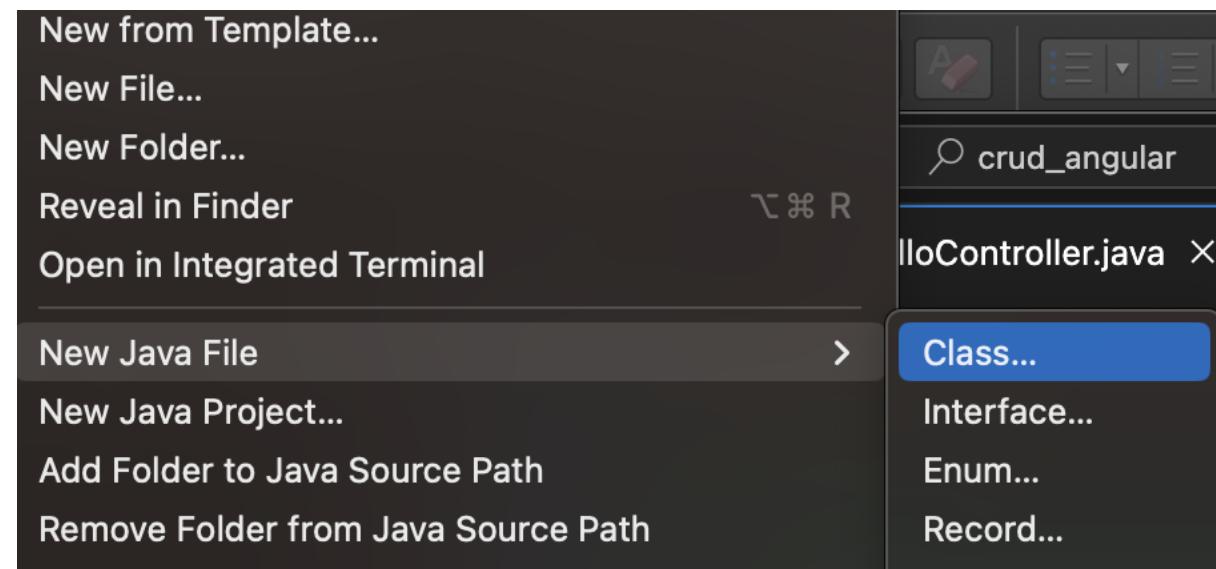
```
@Data  
@Entity  
//@Table(name = "curso")  
public class Course {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.AUTO)  
    private Long id;  
  
    @Column(length = 200, nullable = false)  
    private String name;  
  
    @Column(length = 30, nullable = false)  
    private String category;  
}
```

# Configurando API

## Criar uma pasta repository



## Criar uma arquivo CourseRepository.java



spring



# Configurando API

Criar uma interface CourseRepository.java dentro do repositor

src/main/java/com/uneb/labweb/spring\_project/repository/CourseRepository.java

```
package com.uneb.labweb.spring_project.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.uneb.labweb.spring_project.model.Course;

@Repository
public interface CourseRepository extends JpaRepository<Course, Long>{

}
```



# Configurando API

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
...
import com.uneb.labweb.spring_project.model.Course;
import com.uneb.labweb.spring_project.repository.CourseRepository;
import lombok.AllArgsConstructor;
...
@AllArgsConstructor
public class CourseController {

    private final CourseRepository courseRepository;

    @GetMapping
    public List<Course> list(){
        return courseRepository.findAll();
    }
}
```



# Configurando API

Editar a classe SpringProjectApplication.java para criar uma database temporária

```
...
import org.springframework.boot.CommandLineRunner;
import org.springframework.context.annotation.Bean;
import com.uneb.labweb.spring_project.model.Course;
import com.uneb.labweb.spring_project.repository.CourseRepository;

...
@Bean
CommandLineRunner initDatabase(CourseRepository courseRepository){
    return args -> {
        courseRepository.deleteAll();
        Course c = new Course();
        c.setName("Curso Angular");
        c.setCategory("Front-end");
        courseRepository.save(c);
    };
}
...
```



# Configurando o Banco de Dados H2

## demo/src/main/resources/application.properties

```
spring.application.name=spring-project
spring.h2.console.enabled=true
spring.h2.console.path=/h2-console
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=12345678
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.jpa.show-sql=true
# http://localhost:8080/h2-console/
```

<http://localhost:8080/h2-console/>



# Configurando o Banco de Dados H2

## demo/pom.xml

```
<dependencies>
...
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-h2console</artifactId>
</dependency>

...
</dependencies>
```

mvn clean package



# Configurando o Banco de Dados H2

<http://localhost:8080/h2-console/>

Login

Saved Settings: Generic H2 (Embedded)

---

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:testdb

User Name: sa

Password: .....

A blue arrow points from the left towards the JDBC URL field.



# Configurando o Banco de Dados H2

<http://localhost:8080/h2-console/>

The screenshot shows the H2 Database Console interface. The top navigation bar includes icons for connection management, a checked 'Auto commit' checkbox, a dropdown for 'Max rows' set to 1000, and dropdowns for 'Auto complete' (Off) and 'Auto select' (On). Below the toolbar is a sidebar with database connections ('jdbc:h2:mem:testdb'), tables ('COURSE'), schemas ('INFORMATION\_SCHEMA'), sequences, users, and a version notice ('H2 2.2.224 (2023-09-17)'). The main area contains a SQL input field with the query 'SELECT \* FROM COURSE' and a results table below it.

SQL statement:

```
SELECT * FROM COURSE
```

Select \* from COurse;

ID	CATEGORY	NAME
1	Front-end	Curso Angular

(1 row, 4 ms)

Edit



# Adicionando API

src/app/courses/services/courses-service.ts

```
...
export class CoursesServices {
  private readonly API = 'http://localhost:8080/api/courses';

  constructor(private httpClient: HttpClient) { }

  list(){
    return this.httpClient.get<Course[]>(this.API)
      .pipe(
        first(),
        //delay(5000),
        tap(courses => console.log(courses))
      );
  }
}
```



# Criando um Proxy apenas se tiver erro do CORS

Se tiver problemas do **CORS** será necessário criar na raiz do projeto do angular um arquivo com o nome **proxy.conf.js**

```
const PROXY_CONFIG = [
  {
    context: ['/api'],
    target: 'http://localhost:8080/',
    secure: false,
    logLevel: 'debug'
  }
];

module.exports = PROXY_CONFIG;
```

# Criando um Proxy apenas se tiver erro do CORS



No **package.json** adicionar a instrução no scripts no **ng serve do start --proxy-config proxy.conf.js**

```
{
  "name": "angular-project",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve --proxy-config proxy.conf.js",
    "build": "ng build",
    "watch": "ng build --watch --configuration development",
    "test": "ng test",
    "serve:ssr:angular-project": "node dist/angular-project/server/server.mjs"
  },
}
```



# Criando um Proxy apenas se tiver erro do CORS

src/app/courses/services/courses-service.ts

```
...
export class CoursesServices {
  private readonly API = 'api/courses';

  constructor(private httpClient: HttpClient) { }

  ...
}
```

O projeto deverá ser compilado como:

**npm run start**

# Preparando o formulário



src/app/courses/courses/courses.html

```
...
<!-- Actions Columns -->
<ng-container matColumnDef="actions">
  <th mat-header-cell *matHeaderCellDef>
    <button mat-mini-fab color="accent" aria-label="Adicionar um curso">
      <mat-icon>add</mat-icon>
    </button>
  </th>
  <td mat-cell *matCellDef="let course">
    <button mat-mini-fab color="primary" aria-label="Editar curso">
      <mat-icon>edit</mat-icon>
    </button>
    <button mat-mini-fab color="warn" aria-label="Remover curso">
      <mat-icon>delete</mat-icon>
    </button>
  </td>
</ng-container>
...
```

# Preparando o formulário



src/app/courses/courses/courses.ts

```
...
export class Courses implements OnInit {
  courses$: Observable<Course[]>;
  displayedColumns: string[] = ['name', 'category', 'actions'];
...
}
```



# Habilitando Formulário

## Criando rota



ng g c courses/course-form

```
› (base) vagner@simaras-MacBook-Pro crud-angular % ng g c courses/course-form
CREATE src/app/courses/course-form/course-form.component.scss (0 bytes)
CREATE src/app/courses/course-form/course-form.component.html (26 bytes)
CREATE src/app/courses/course-form/course-form.component.spec.ts (625 bytes)
CREATE src/app/courses/course-form/course-form.component.ts (254 bytes)
```

# Habilitando Formulário

## Criando rota



src/app/courses/courses-routing.module.ts

```
import { CourseForm } from './course-form/course-form';

const routes: Routes = [
  { path : "", component : Courses },
  { path : 'new', component : CourseForm }
];
```

# Habilitando Formulário

## Criando rota



src/app/courses/courses/courses.html

```
<th mat-header-cell *matHeaderCellDef>
  <button mat-mini-fab color="accent" aria-label="Adicionar um curso" (click)="onAdd()">
    <mat-icon>add</mat-icon>
  </button>
</th>
```

# Habilitando Formulário Criando rota



src/app/courses/courses/courses.ts

```
import { ActivatedRoute, Router } from '@angular/router';
...
constructor(
  private courseService: CoursesServices,
  public dialog: MatDialog,
  private router : Router,
  private route : ActivatedRoute
){
...
}

onAdd(){
  this.router.navigate(['new'], { relativeTo : this.route });
}
...
```

# Criando o Formulário



src/app/courses/courses-module.ts

```
...
import { ReactiveFormsModule } from '@angular/forms';
import { MatFormFieldModule } from '@angular/material/form-field';

@NgModule({
  declarations: [],
  imports: [
    CommonModule,
    CoursesRoutingModule,
    ErrorDialogComponent,
    ReactiveFormsModule,
    MatFormFieldModule
  ]
})
...
...
```



# Criando o Formulário

src/app/courses/services/courses-service.ts

```
...
export class CoursesServices {
...
  save(record: Course){
    return this.httpClient.post<Course>(this.API, record);
  }
...
}
```

# Criando o Formulário



src/app/courses/course-form/course-form.ts

```
import { Component, OnInit } from '@angular/core';
import { Location } from '@angular/common';
import { FormBuilder, FormGroup, ReactiveFormsModule } from '@angular/forms';
import { MatButtonModule } from '@angular/material/button';
import { MatCardModule } from '@angular/material/card';
import { MatFormFieldModule } from '@angular/material/form-field';
import { MatInputModule } from '@angular/material/input';
import { MatSelectModule } from '@angular/material/select';
import { MatSnackBar, MatSnackBarModule } from '@angular/material/snack-bar';
import { MatToolbarModule } from '@angular/material/toolbar';

import { CoursesServices } from '../services/courses-services';
...
```

# Criando o Formulário



src/app/courses/course-form/course-form.ts

```
...
@Component({
  selector: 'app-course-form',
  standalone: true,
  imports: [MatFormFieldModule, ReactiveFormsModule,
    MatInputModule, MatCardModule, MatToolbarModule,
    MatButtonModule, MatSelectModule, MatSnackBarModule],
  templateUrl: './course-form.html',
  styleUrls: ['./course-form.scss']
})
...
```

# Criando o Formulário



src/app/courses/course-form/course-form.ts

```
...
export class CourseForm implements OnInit{
  form: FormGroup;
  constructor(private formBuilder: FormBuilder,
    private service: CoursesService,
    private snackBar: MatSnackBar,
    private location : Location
  ){
    this.form = this.formBuilder.group({
      name: [null],
      category: [null]
    });
  }
...
}
```



# Criando o Formulário

src/app/courses/course-form/course-form.ts

```
...
onSubmit(){
  this.service.save(this.form.value).subscribe(
    result => this.onSuccess(),
    error => this.onError()
  );
}

private onSuccess(){
  this.snackBar.open('Novo curso salvo com sucesso', '',
    { duration : 5000, });
  this.onCancel();
}
...
```

# Criando o Formulário



src/app/courses/course-form/course-form.ts

```
...
private onError(){
  this.snackBar.open('Erro ao salvar novo curso', '',
    { duration : 5000, });
  this.onCancel();
}

onCancel(){
  this.location.back();
}

ngOnInit(): void { }

}
```

# Criando o Formulário



src/app/courses/course-form/course-form.html

```
<mat-card>
  <mat-toolbar>Cadastrar novo Curso</mat-toolbar>
  <mat-card-content>
    <form [formGroup]="form" class="min-width">
      <mat-form-field class="fill-width">
        <input matInput placeholder="Nome" formControlName="name">
      </mat-form-field>
      <mat-form-field class="fill-width">
        <mat-label>Categoria</mat-label>
        <mat-select formControlName="category">
          <mat-option value="Front-end">Front-end</mat-option>
          <mat-option value="Back-end">Back-end</mat-option>
        </mat-select>
      </mat-form-field>
    ...
  </mat-card-content>
</mat-card>
```

# Criando o Formulário



src/app/courses/course-form/course-form.html

```
...
</form>
</mat-card-content>
<mat-card-actions class="actions-center">
  <button mat-raised-button color="primary"
    (click)="onSubmit()">Salvar</button>
  <button mat-raised-button color="warn"
    (click)="onCancel()">Cancelar</button>
</mat-card-actions>
</mat-card>
...
```

# Criando o Formulário



src/app/courses/course-form/course-form.scss

```
mat-card{  
  max-width: 80%;  
  margin: 2em auto;  
  text-align: center;  
}  
.action-center{  
  justify-content: center;  
}  
.fill-width{  
  width: 100%;  
  min-width: 150px;  
}  
.min-width{  
  width: 100%;  
  min-width: 150px;  
}
```



# Persistência do Formulário

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
...
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotationResponseStatus;
import org.springframework.web.bind.annotation.RestController;
...
public class CourseController {
...
    @PostMapping
    @ResponseStatus(code = HttpStatus.ACCEPTED)
    public Course create(@RequestBody Course course) {
        return courseRepository.save(course);
    }
...
}
```

# Validando o Formulário



src/app/courses/courses/courses-form.ts

```
export class CourseForm implements OnInit{
  form = this.formBuilder.group({
    name: [""],
    category: []
  });

  constructor(private formBuilder: NonNullableFormBuilder,
  private service: CoursesServices,
  private snackBar: MatSnackBar,
  private location : Location
){}

...
```

# Validando o Formulário



src/app/courses/services/courses-service.ts

```
export class CoursesServices {  
...  
  save(record: Partial<Course>){  
    return this.httpClient.post<Course>(this.API, record);  
  }  
}
```



# Refatoração

ng g c courses/courses-list

```
● (base) vagner@simaras-MacBook-Pro angular-project % ng g c courses/courses-list
CREATE src/app/courses/courses-list/courses-list.component.scss (0 bytes)
CREATE src/app/courses/courses-list/courses-list.component.html (27 bytes)
CREATE src/app/courses/courses-list/courses-list.component.spec.ts (628 bytes)
CREATE src/app/courses/courses-list/courses-list.component.ts (258 bytes)
```



# Refatoração

src/app/courses/courses/courses.html

```
<mat-card>
  <mat-toolbar color="primary">Cursos Disponíveis</mat-toolbar>
  <div class="column-flex mat-elevation-z8">
    <app-courses-list
      *ngIf="courses$ | async as courses; else loading"
      [courses]="courses"
      (add)="onAdd()"
    ></app-courses-list>

    <ng-template #loading>
      <div class="loading-spinner">
        <mat-spinner></mat-spinner>
      </div>
    </ng-template>
  </div>
</mat-card>
```



# Refatoração

src/app/courses/courses/courses.scss

```
mat-card {  
  max-width: 80%;  
  margin: 2em auto;  
  text-align: center;  
}  
.column-flex {  
  display: flex;  
  flex-direction: column;  
  min-width: 300px;  
}  
.loading-spinner {  
  padding: 25px;  
  background: rgba(0, 0, 0, 0.15);  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```



# Refatoração

src/app/courses/courses/courses.ts

```
export class Courses implements OnInit {  
  courses$: Observable<Course[]>;  
  //displayedColumns: string[] = ['name', 'category', 'actions'];  
  ...  
}
```



# Refatoração

## src/app/courses/courses-list/courses-list.ts

```
import { Component, EventEmitter, Input, OnInit, Output } from '@angular/core';
import { MatButtonModule } from '@angular/material/button';
import { MatIconModule } from '@angular/material/icon';
import { MatTableModule } from '@angular/material/table';

import { CategoryPipe } from '../../shared/pipes/category-pipe';
import { Course } from '../../model/course';

@Component({
  selector: 'app-courses-list',
  standalone: true,
  imports: [MatTableModule, MatButtonModule, MatIconModule, CategoryPipe],
  templateUrl: './courses-list.component.html',
  styleUrls: ['./courses-list.component.scss']
})
...
```



# Refatoração

src/app/courses/courses-list/courses-list.ts

```
...
export class CoursesList implements OnInit {

  @Input() courses: Course[] = [];
  @Output() add = new EventEmitter(false);

  readonly displayedColumns: string[] = ['name', 'category', 'actions'];

  constructor(){}
  ngOnInit(): void { }

  onAdd()
    this.add.emit(true);
  }
}
```



# Refatoração

src/app/courses/courses-list/courses-list.html

```
<mat-table [dataSource]="courses">
  <ng-container matColumnDef="name">
    <mat-header-cell *matHeaderCellDef> Curso </mat-header-cell>
    <mat-cell *matCellDef="let course"> {{course.name}} </mat-cell>
  </ng-container>

  <ng-container matColumnDef="category">
    <mat-header-cell *matHeaderCellDef> Categoria </mat-header-cell>
    <mat-cell *matCellDef="let course"> {{ course.category }}<br/>
      <mat-icon aria-hidden="false" aria-label="Categoria do Curso">
        {{ course.category | category }}
      </mat-icon>
    </mat-cell>
  </ng-container>
  <!-- Actions Columns -->
  ...
</mat-table>
```



# Refatoração

src/app/courses/courses-list/courses-list.html

```
...
<ng-container matColumnDef="actions">
  <mat-header-cell *matHeaderCellDef class="action-column-header">
    <button mat-mini-fab color="accent" aria-label="Adicionar um curso"
      (click)="onAdd()">
      <mat-icon>add</mat-icon>
    </button>
  </mat-header-cell>
  <mat-cell *matCellDef="let course" class="action-column">
    <button mat-mini-fab color="primary" aria-label="Editar curso">
      <mat-icon>edit</mat-icon>
    </button>
    <button mat-mini-fab color="warn" aria-label="Remover curso">
      <mat-icon>delete</mat-icon>
    </button>
  </mat-cell>
</ng-container>
...
```



# Refatoração

src/app/courses/courses-list/courses-list.html

```
...
<mat-header-row *matHeaderRowDef="displayedColumns"></mat-header-row>
<mat-row *matRowDef="let row; columns: displayedColumns;"></mat-row>
</mat-table>
```



# Refatoração

src/app/courses/courses-list/courses-list.scss

```
.mat-table {  
  overflow: auto;  
  max-height: 600px;  
}  
  
.action-column {  
  flex: 0 0 125px;  
}  
  
.action-column-header {  
  flex: 0 0 105px;  
}
```



# Refatoração

src/app/courses/courses-list/courses-list.scss

```
.mat-table {  
  overflow: auto;  
  max-height: 600px;  
}  
  
.action-column {  
  flex: 0 0 125px;  
}  
  
.action-column-header {  
  flex: 0 0 105px;  
}
```



# Editando um Curso

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
public class CourseController {  
    ...  
    @GetMapping("/{id}")  
    public ResponseEntity<Course> findById(@PathVariable Long id){  
        return courseRepository.findById(id)  
            .map(record -> ResponseEntity.ok().body(record))  
            .orElse(ResponseEntity.notFound().build());  
    }  
    ...  
}
```

# Editando um Curso



src/app/courses/course-form/course-form.html

```
<mat-card>
  <mat-toolbar>Detalhes do Curso</mat-toolbar>
  ...
</mat-card>
```

# Editando um Curso



src/app/courses/model/course.ts

```
export interface Course {  
  id: string;  
  name: string;  
  category: string;  
}
```

# Editando um Curso



src/app/courses/courses-list/courses-list.ts

```
...
export class CoursesList implements OnInit {

  @Input() courses: Course[] = [];
  @Output() add = new EventEmitter(false);
  @Output() edit = new EventEmitter(false);

  ...

  onEdit(course: Course) {
    this.edit.emit(course);
  }

}
```

# Editando um Curso



src/app/courses/courses-list/courses-list.html

```
...
<mat-cell *matCellDef="let course" class="action-column">
  <button mat-mini-fab color="primary" aria-label="Editar curso"
    (click)="onEdit(course)">
    <mat-icon>edit</mat-icon>
  </button>
  <button mat-mini-fab color="warn" aria-label="Remover curso">
    <mat-icon>delete</mat-icon>
  </button>
</mat-cell>
...
```

# Editando um Curso



src/app/courses/courses/courses.html

```
...
<app-courses-list
  *ngIf="courses$ | async as courses; else loading"
  [courses]="courses"
  (add)="onAdd()"
  (edit)="onEdit($event)"
></app-courses-list>
...
```

# Criando rotas da Edição do Curso



src/app/courses/courses/courses.ts

```
export class CoursesComponent implements OnInit {  
  ...  
  onEdit(course: Course) {  
    this.router.navigate(['edit', course.id], { relativeTo: this.route });  
  }  
  ...  
}
```

# Criando rotas da Edição do Curso



src/app/courses/courses-routing.module.ts

```
...
const routes: Routes = [
  { path : '', component : CoursesComponent },
  { path : 'new', component : CourseFormComponent },
  { path : 'edit/:id', component: CourseFormComponent }
];
...
```

# Carregando Informações do Curso



## ng g resolver courses/guards/course

```
• (base) vagner@simaras-MacBook-Pro crud-angular % ng g resolver courses/guards/course
CREATE src/app/courses/guards/course.resolver.spec.ts (494 bytes)
CREATE src/app/courses/guards/course.resolver.ts (134 bytes)
```

# Carregando Informações do Curso



src/app/courses/services/courses-services.ts

```
export class CoursesServices {  
...  
  loadById(id: string) {  
    return this.httpClient.get<Course>(`${this.API}/${id}`);  
  }  
}
```

# Carregando Informações do Curso



src/app/courses/guards/course.resolver.ts

```
import { Injectable } from '@angular/core';
import { ActivatedRouteSnapshot, RouterStateSnapshot } from '@angular/router';
import { Observable, of } from 'rxjs';
import { Course } from '../model/course';
import { CoursesServices } from '../services/courses-services';

@Injectable({
  providedIn: 'root'
})
...
```

# Carregando Informações do Curso



src/app/courses/guards/course.resolver.ts

```
...
export class CourseResolver {

  constructor(private service: CoursesServices) { }

  resolve(route: ActivatedRouteSnapshot,
  state: RouterStateSnapshot): Observable<Course> {
    if (route.params && route.params['id']) {
      return this.service.loadById(route.params['id']);
    }
    return of({ id: "", name: "", category: "" });
  }
}
```

# Carregando Informações do Curso



src/app/courses/course-form/course-form.ts

```
...
export class CourseForm implements OnInit {
  form = this.formBuilder.group({
    id: '',
    name: '',
    category: ''
  });
...
}
```

# Carregando Informações do Curso



src/app/courses/course-form/course-form.ts

```
...
export class CourseForm implements OnInit {
  ...
  ngOnInit(): void {
    const course: Course = this.route.snapshot.data['course'];
    this.form.setValue({
      id: course.id,
      name: course.name,
      category: course.category
    });
  }
}
```

# Carregando Informações do Curso



src/app/courses/courses-routing.module.ts

```
...
import { CourseResolver } from './guards/course-resolver';

const routes: Routes = [
  { path: '', component: Courses },
  { path: 'new', component: CourseForm,
    resolve: { course: CourseResolver } },
  { path: 'edit/:id', component: CourseForm,
    resolve: { course: CourseResolver } }
];
...
```

# Carregando Informações do Curso



src/app/courses/services/courses-service.ts

```
export class CoursesServices {  
...  
  save(record: Partial<Course>){  
    if (record.id){  
      return this.update(record);  
    }else{  
      return this.create(record);  
    }  
  }  
  private create(record: Partial<Course>){  
    return this.httpClient.post<Course>(this.API, record);  
  }  
  private update(record: Partial<Course>){  
    return this.httpClient.put<Course>(`${this.API}/${record.id}`, record);  
  }  
}
```

# Salvando Informações do Curso



src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
public class CourseController {  
    ...  
    @PutMapping("/{id}")  
    public ResponseEntity<Course> update(@PathVariable Long id,  
        @RequestBody Course course){  
        return courseRepository.findById(id)  
            .map(recordFound -> {  
                recordFound.setName(course.getName());  
                recordFound.setCategory(course.getCategory());  
                Course updated = courseRepository.save(recordFound);  
                return ResponseEntity.ok().body(updated);  
            })  
            .orElse(ResponseEntity.notFound().build());  
    }  
    ...  
}
```



# Deletando um Curso

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
public class CourseController {  
    ...  
    @DeleteMapping("/{id}")  
    public ResponseEntity<Void> delete(@PathVariable Long id){  
        return courseRepository.findById(id)  
            .map(recordFound -> {  
                courseRepository.deleteById(id);  
                return ResponseEntity.noContent().<Void>build();  
            })  
            .orElse(ResponseEntity.notFound().build());  
    }  
    ...  
}
```



# Deletando um Curso

ng g c shared/components/confirmation-dialog

```
(base) vagner@simaras-MacBook-Pro angular-project % ng g c shared/components/confirmation-dialog
CREATE src/app/shared/components/confirmation-dialog/confirmation-dialog.component.scss (0 bytes)
CREATE src/app/shared/components/confirmation-dialog/confirmation-dialog.component.html (34 bytes)
CREATE src/app/shared/components/confirmation-dialog/confirmation-dialog.component.spec.ts (677 bytes)
CREATE src/app/shared/components/confirmation-dialog/confirmation-dialog.component.ts (286 bytes)
```

# Deletando um Curso



src/app/shared/components/confirmation-dialog/confirmation-dialog.ts

```
import { Component, Inject } from '@angular/core';
import { MatButtonModule } from '@angular/material/button';
import { MatDialogModule, MAT_DIALOG_DATA, MatDialogRef } from '@angular/material/dialog';

@Component({
  selector: 'app-confirmation-dialog',
  standalone: true,
  imports: [MatButtonModule, MatDialogModule],
  templateUrl: './confirmation-dialog.html',
  styleUrls: ['./confirmation-dialog.scss']
})
...
```

# Deletando um Curso



src/app/shared/components/confirmation-dialog/confirmation-dialog.ts

```
...
export class ConfirmationDialog {

  constructor(public dialogRef: MatDialogRef<ConfirmationDialog>,
    @Inject(MAT_DIALOG_DATA) public data: string,
  ){ }

  onConfirm(result: boolean){
    this.dialogRef.close(result);
  }
}
```

# Deletando um Curso



src/app/shared/components/confirmation-dialog/confirmation-dialog.html

```
<mat-dialog-content>
  <p>{{ data }}</p>
</mat-dialog-content>
<mat-dialog-actions>
  <button mat-raised-button color="primary" (click)="onConfirm(true)">
    Sim
  </button>
  <button mat-raised-button color="warn" (click)="onConfirm(false)">
    Não
  </button>
</mat-dialog-actions>
```



# Deletando um Curso

src/app/courses/services/courses-services.ts

```
export class CoursesServices {  
  ...  
  remove(id: string){  
    return this.httpClient.delete(`.${this.API}/${id}`);  
  }  
}
```



# Deletando um Curso

src/app/courses/courses-list/courses-list.html

```
...
<mat-cell *matCellDef="let course" class="action-column">
  <button mat-mini-fab color="primary" aria-label="Editar curso"
    (click)="onEdit(course)">
    <mat-icon>edit</mat-icon>
  </button>
  <button mat-mini-fab color="warn" aria-label="Remover curso"
    click="onRemove(course)">
    <mat-icon>delete</mat-icon>
  </button>
</mat-cell>
...
...
```



# Deletando um Curso

src/app/courses/courses-list/courses-list.component.ts

```
...
export class CoursesListComponent implements OnInit {

  @Input() courses: Course[] = [];
  @Output() add = new EventEmitter(false);
  @Output() edit = new EventEmitter(false);
  @Output() remove = new EventEmitter(false);

  ...

  onRemove(course: Course) {
    this.remove.emit(course);
  }

}
```



# Deletando um Curso

src/app/courses/courses/courses.html

```
...
<app-courses-list
  *ngIf="courses$ | async as courses; else loading"
  [courses]="courses"
  (add)="onAdd()"
  (edit)="onEdit($event)"
  (remove)="onRemove($event)"
></app-courses-list>
...
```

# Deletando um Curso



src/app/courses/courses/courses.ts

```
import { MatSnackBar } from '@angular/material/snack-bar';
import { ConfirmationDialogComponent } from '../../shared/components/confirmation-dialog/confirmation-dialog.';
...

export class CoursesComponent implements OnInit {

  courses$: Observable<Course[]> | null = null;

}

}
```



# Deletando um Curso

src/app/courses/courses/courses.ts

```
export class Courses implements OnInit {  
  ...  
  constructor(  
    private courseService: CoursesService,  
    public dialog: MatDialog,  
    private router : Router,  
    private route : ActivatedRoute,  
    private snackBar: MatSnackBar,  
  ){  
    this.refresh();  
  }  
  ...  
}
```



# Deletando um Curso

src/app/courses/courses/courses.ts

```
import { MatSnackBar } from '@angular/material/snack-bar';
...
export class CoursesComponent implements OnInit {
...
refresh(){
  this.courses$ = this.courseService.list()
    .pipe(
      catchError(error => {
        this.onError('Erro ao carregar a lista de cursos');
        return of([])
      })
    );
}
...
}
```



# Deletando um Curso

src/app/courses/courses/courses.ts

```
export class CoursesComponent implements OnInit {  
  ...  
  onRemove(course: Course) {  
    const dialogRef = this.dialog.open(ConfirmationDialog, {  
      data: 'Deseja realmente excluir o curso ' + course.name,  
    });  
    dialogRef.afterClosed().subscribe((result : boolean) => {  
      if (result){  
        this.courseService.remove(course.id).subscribe(  
          () => {  
            this.refresh();  
            this.snackBar.open('Curso removido com sucesso', 'X', {  
              duration : 5000,  
              verticalPosition: 'top',  
              horizontalPosition: 'center'  
            }  
          }  
        );  
      }  
    });  
  }  
}
```



# Deletando um Curso

src/app/courses/courses/courses.ts

```
export class Courses implements OnInit {  
  ...  
  onRemove(course: Course) {  
    ...  
    } );  
    },  
    () => this.onError('Erro ao tentar remover curso.')  
  );  
}  
});  
}  
...  
}
```

# Adicionando validação no formulário



src/app/courses/course-form/course-form.ts

```
import { CommonModule, Location } from '@angular/common';
...
import { NonNullableFormBuilder, ReactiveFormsModule, Validators } from '@angular/forms';
...
@Component({
  selector: 'app-course-form',
  standalone: true,
  imports: [MatFormFieldModule, ReactiveFormsModule, MatInputModule,
    MatCardModule, MatToolbarModule, MatButtonModule, MatSelectModule,
    MatSnackBarModule, CommonModule],
  templateUrl: './course-form.component.html',
  styleUrls: ['./course-form.component.scss']
})
...
```

# Adicionando validação no formulário



src/app/courses/course-form/course-form.ts

```
...
export class CourseForm implements OnInit{

  form = this.formBuilder.group({
    id: [""],
    name: ['', [Validators.required,
      Validators.minLength(5),
      Validators.maxLength(100)]],
    category: ['', [Validators.required]],
  });
...
}
```

# Adicionando validação no formulário



src/app/courses/course-form/course-form.ts

```
...
export class CourseForm implements OnInit{

  ...
  errorMessage(fieldName: string){
    const field = this.form.get(fieldName);
    if (field?.hasError('required')){
      return 'Campo obrigatorio';
    }
    if (field?.hasError('minlength')){
      const requiredLength : number = field.errors ?
        field.errors['minlength']['requiredLength'] : 5;
      return `Tamanho mínimo precisa ser de ${requiredLength} caracteres.`;
    }
  ...
}
}
```

# Adicionando validação no formulário



src/app/courses/course-form/course-form.ts

```
...
export class CourseForm implements OnInit{
  ...
  errorMessage(fieldName: string){
    ...
    if (field?.hasError('maxlength')){
      const requiredLength : number = field.errors ?
        field.errors['maxlength']['requiredLength'] : 100;
      return `Tamanho máximo excedido de ${requiredLength} caracteres.`;
    }
    return 'Campo invalido!';
  }
  ...
}
```

# Adicionando validação no formulário



src/app/courses/course-form/course-form.html

```
...
<mat-form-field class="fill-width">
  <input matInput placeholder="Nome" formControlName="name" #name>
  <mat-hint align="end">{{name.value.length || 0}}/100</mat-hint>
  <mat-error *ngIf="form.get('name')?.invalid">{{  
    errorMessage("name")  
}}</mat-error>
</mat-form-field>
...
<mat-form-field class="fill-width">
  <mat-label>Categoria</mat-label>
  <mat-select formControlName="category"> ... </mat-select>
  <mat-error *ngIf="form.get('category')?.invalid">{{  
    errorMessage("category")  
}}</mat-error>
</mat-form-field>
...
```

# Adicionando validação no formulário



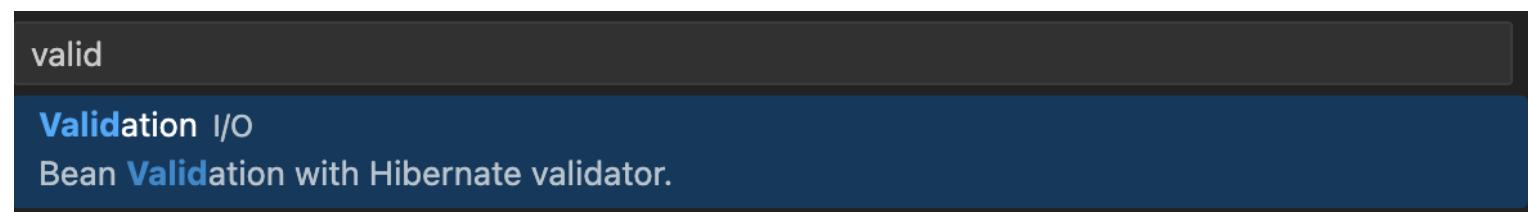
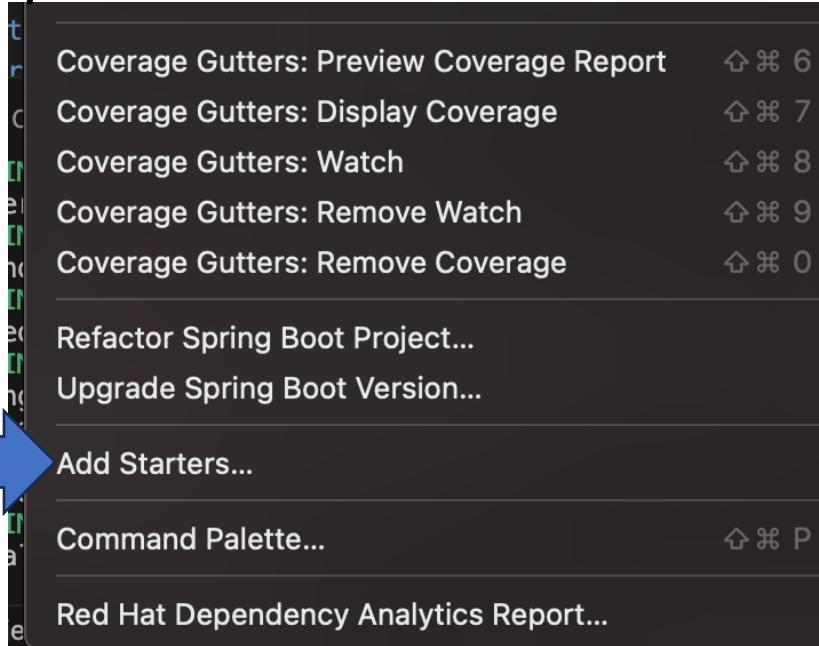
target/classes/application.properties

```
...  
  
server.error.include-stacktrace=never  
server.error.include-message=never
```

# Adicionando validação no formulário



pom.xml



Select dependencies to add.

Selected 7 dependencies

Press <Enter> to continue.

# Adicionando validação no formulário



## pom.xml

```
project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
...
<dependencies>
...
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-validation</artifactId>
</dependency>
</dependencies>
...
</project>
```

# Adicionando validação no formulário



src/main/java/com/uneb/labweb/spring\_project/model/Course.java

```
...
import jakarta.validation.constraints.NotBlank;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Pattern;
...
```



# Adicionando validação no formulário

src/main/java/com/uneb/labweb/spring\_project/model/Course.java

```
...
public class Course {
...
    @NotBlank
    @NotNull
    @Length(min = 5, max = 100)
    @Column(length = 100, nullable = false)
    private String name;

    @NotNull
    @Length(max = 10)
    @Pattern(regexp = "Back-end|Front-end")
    @Column(length = 10, nullable = false)
    private String category;
}
```

# Adicionando validação no formulário



src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
...
import jakarta.validation.Valid;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Positive;

...
@Validated
@RestController
@RequestMapping("/api/courses")
@AllArgsConstructor
...
...
```

# Adicionando validação no formulário



src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
...
public class CourseController {
...
    @PostMapping
    @ResponseStatus(code = HttpStatus.ACCEPTED)
    public Course create(@RequestBody @Valid Course course) {
        return courseRepository.save(course);
    }
...
}
```



# Adicionando validação no formulário

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
public class CourseController {  
    ...  
    @GetMapping("/{id}")  
    public ResponseEntity<Course> findById(@PathVariable @NotNull  
        @Positive Long id){  
        return courseRepository.findById(id)  
            .map(record -> ResponseEntity.ok().body(record))  
            .orElse(ResponseEntity.notFound().build());  
    }  
    ...  
}
```

# Adicionando validação no formulário



src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
public class CourseController {  
    ...  
    @PutMapping("/{id}")  
    public ResponseEntity<Course> update(@PathVariable @NotNull  
        @Positive Long id, @RequestBody @Valid Course course){  
        return courseRepository.findById(id)  
            .map(recordFound -> {  
                recordFound.setName(course.getName());  
                recordFound.setCategory(course.getCategory());  
                Course updated = courseRepository.save(recordFound);  
                return ResponseEntity.ok().body(updated);  
            })  
            .orElse(ResponseEntity.notFound().build());  
    }  
    ...  
}
```



# Adicionando validação no formulário

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
public class CourseController {  
    ...  
    @DeleteMapping("/{id}")  
    public ResponseEntity<Void> delete(@PathVariable @NotNull  
        @Positive Long id){  
        return courseRepository.findById(id)  
            .map(recordFound -> {  
                courseRepository.deleteById(id);  
                return ResponseEntity.noContent().<Void>build();  
            })  
            .orElse(ResponseEntity.notFound().build());  
    }  
    ...  
}
```



# Remoção Logica do Banco Soft Delete

src/main/java/com/uneb/labweb/spring\_project/model/Course.java

```
...
@SQLDelete(sql = "UPDATE Course SET status = 'Inativo' WHERE id = ?")
@SQLRestriction("status = 'Ativo'")
public class Course {

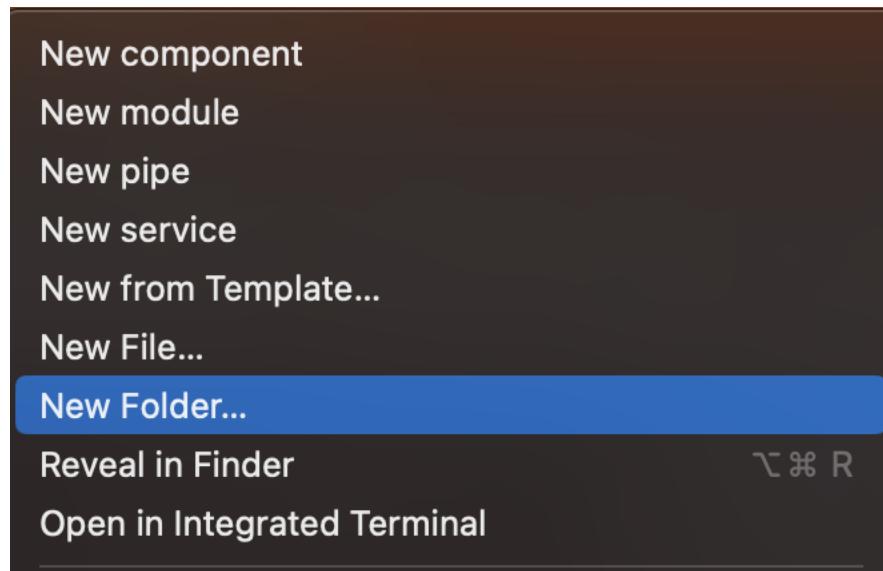
    ...
    @NotNull
    @Length(max = 10)
    @Pattern(regexp = "Ativo|Inativo")
    @Column(length = 10, nullable = false)
    private String status = "Ativo";
}
```



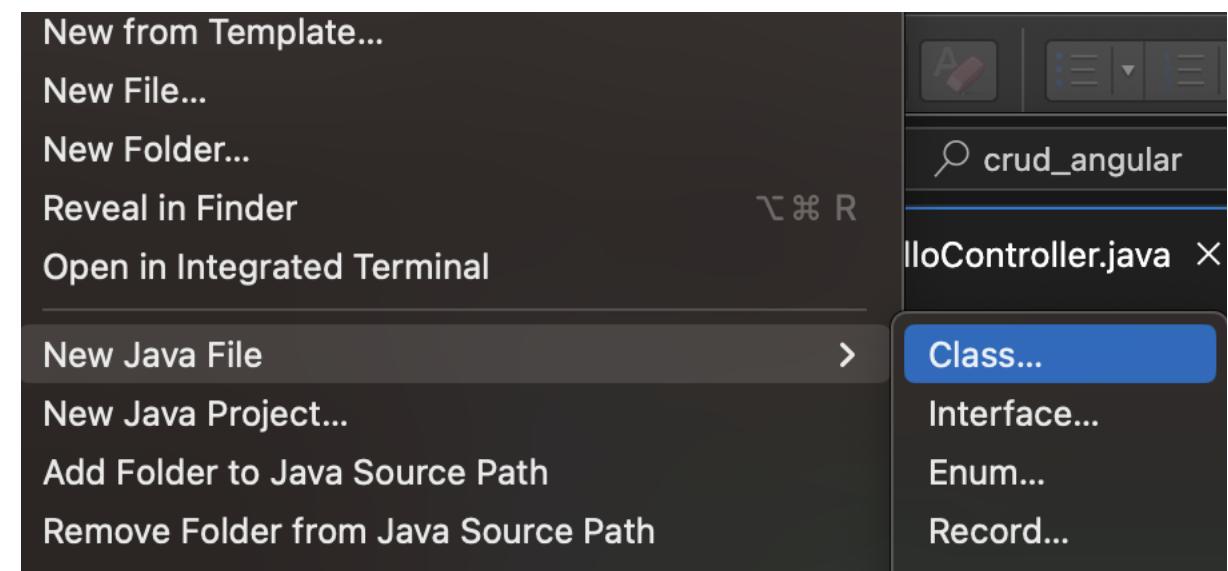
# Adicionando camada Serviço

src/main/java/com/uneb/labweb/spring\_project/

Criar uma pasta service



Criar uma class CourseService.java



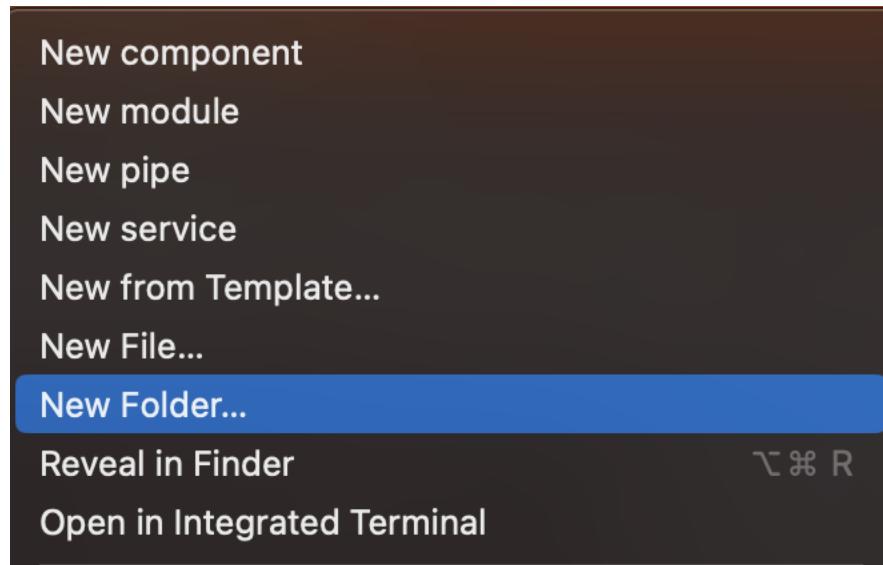
# Adicionando camada Serviço e Tratamento de Exceções



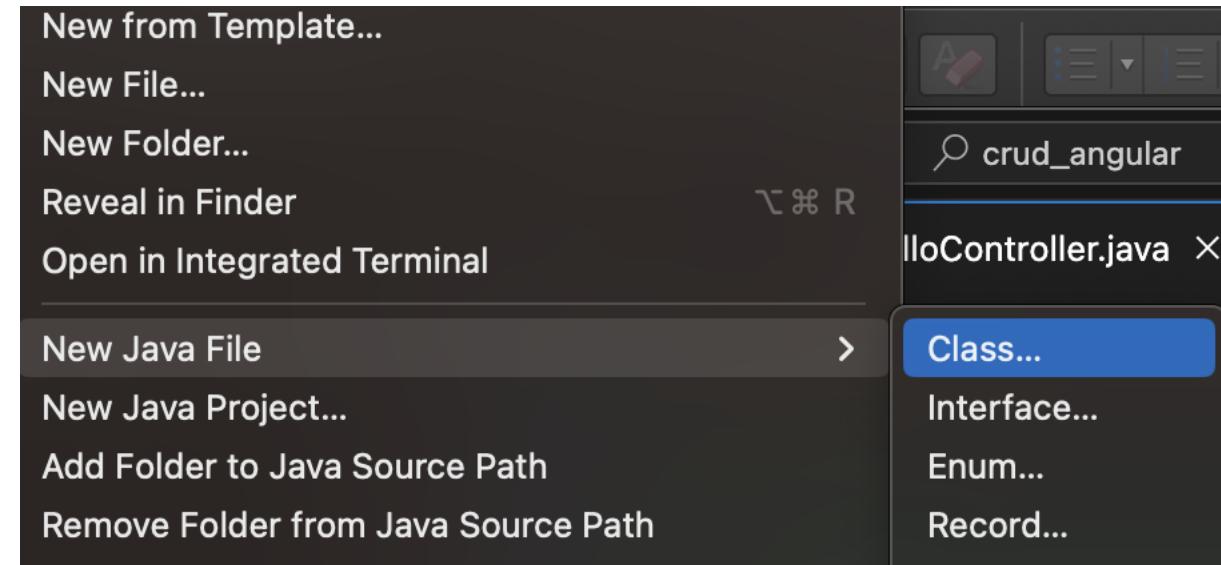
spring

src/main/java/com/uneb/labweb/spring\_project/

Criar uma pasta exception



Criar uma class RecordNotFoundException.java

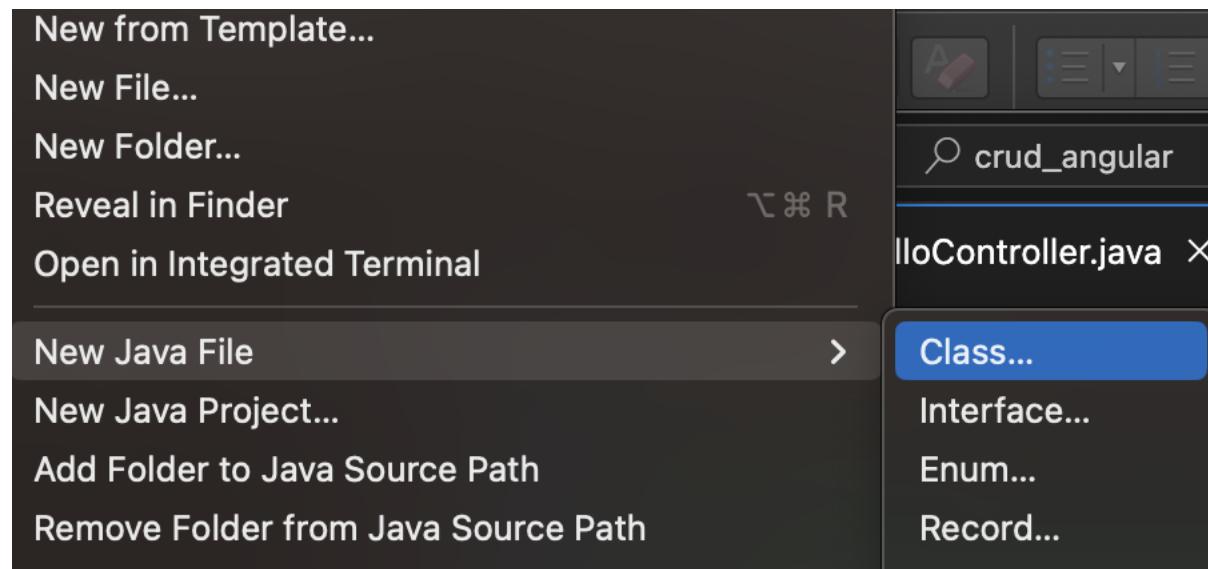


# Adicionando camada Serviço e Tratamento de Exceções



src/main/java/com/uneb/labweb/spring\_project/controller

Criar uma class ApplicationControllerAdvice.java



# Adicionando camada Serviço e Tratamento de Exceções



src/main/java/com/uneb/labweb/spring\_project/service/CourseService.java

```
@Service
@Validated
public class CourseService {

    private final CourseRepository courseRepository;

    public CourseService(CourseRepository courseRepository) {
        this.courseRepository = courseRepository;
    }

    public List<Course> list(){
        return courseRepository.findAll();
    }

    ...
}
```



# Adicionando camada Serviço e Tratamento de Exceções

src/main/java/com/uneb/labweb/spring\_project/service/CourseService.java

```
public class CourseService {  
    ...  
    public Course findById(@NotNull @Positive Long id){  
        return courseRepository.findById(id)  
            .orElseThrow(() -> new RecordNotFoundException(id));  
    }  
  
    public void delete(@NotNull @Positive Long id){  
        courseRepository.delete(courseRepository.findById(id)  
            .orElseThrow(() -> new RecordNotFoundException(id)));  
    }  
    ...  
}
```



# Adicionando camada Serviço e Tratamento de Exceções

src/main/java/com/uneb/labweb/spring\_project/service/CourseService.java

```
public class CourseService {  
    ...  
    public Course update(@NotNull @Positive Long id, @Valid Course course){  
        return courseRepository.findById(id)  
            .map(recordFound -> {  
                recordFound.setName(course.getName());  
                recordFound.setCategory(course.getCategory());  
                return courseRepository.save(recordFound);  
            })  
            .orElseThrow(() -> new RecordNotFoundException(id));  
    }  
  
    public Course create(@Valid Course course) {  
        return courseRepository.save(course);  
    }  
}
```



# Adicionando camada Serviço e Tratamento de Exceções

src/main/java/com/uneb/labweb/spring\_project/exception/RecordNotFoundException.java

```
public class RecordNotFoundException extends RuntimeException {  
  
    private static final long serialVersionUID = 1L;  
  
    public RecordNotFoundException(Long id) {  
        super("Registro não encontrado com o id: " + id);  
    }  
}
```



# Adicionando camada Serviço e Tratamento de Exceções

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
@Validated
@RestController
@RequestMapping("/api/courses")
public class CourseController {

    private final CourseService courseService;

    public CourseController(CourseService courseService) {
        this.courseService = courseService;
    }

    @GetMapping
    public List<Course> list(){
        return courseService.list();
    }

    ...
}
```



# Adicionando camada Serviço e Tratamento de Exceções

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
public class CourseController {  
    ...  
    @GetMapping("/{id}")  
    public Course findById(@PathVariable @NotNull @Positive Long id){  
        return courseService.findById(id);  
    }  
  
    @PutMapping("/{id}")  
    public Course update(@PathVariable @NotNull @Positive Long id,  
        @RequestBody @Valid Course course){  
        return courseService.update(id, course);  
    }  
    ...  
}
```



# Adicionando camada Serviço e Tratamento de Exceções

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
public class CourseController {  
    ...  
    @DeleteMapping("/{id}")  
    @ResponseStatus(code = HttpStatus.NO_CONTENT)  
    public void delete(@PathVariable @NotNull @Positive Long id){  
        courseService.delete(id);  
    }  
  
    @PostMapping  
    @ResponseStatus(code = HttpStatus.ACCEPTED)  
    public Course create(@RequestBody @Valid Course course) {  
        return courseService.create(course);  
    }  
}
```



# Adicionando camada Serviço e Tratamento de Exceções

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
public class CourseController {  
    ...  
    @DeleteMapping("/{id}")  
    @ResponseStatus(code = HttpStatus.NO_CONTENT)  
    public void delete(@PathVariable @NotNull @Positive Long id){  
        courseService.delete(id);  
    }  
  
    @PostMapping  
    @ResponseStatus(code = HttpStatus.ACCEPTED)  
    public Course create(@RequestBody @Valid Course course) {  
        return courseService.create(course);  
    }  
}
```



# Adicionando camada Serviço e Tratamento de Exceções

src/main/java/com/uneb/labweb/spring\_project/controller/ApplicationControllerAdvice.java

```
@RestControllerAdvice
public class ApplicationControllerAdvice {

    @ExceptionHandler(RecordNotFoundException.class)
    @ResponseStatus(HttpStatus.NOT_FOUND)
    public String handleNotFoundException(RecordNotFoundException ex) {
        return ex.getMessage();
    }

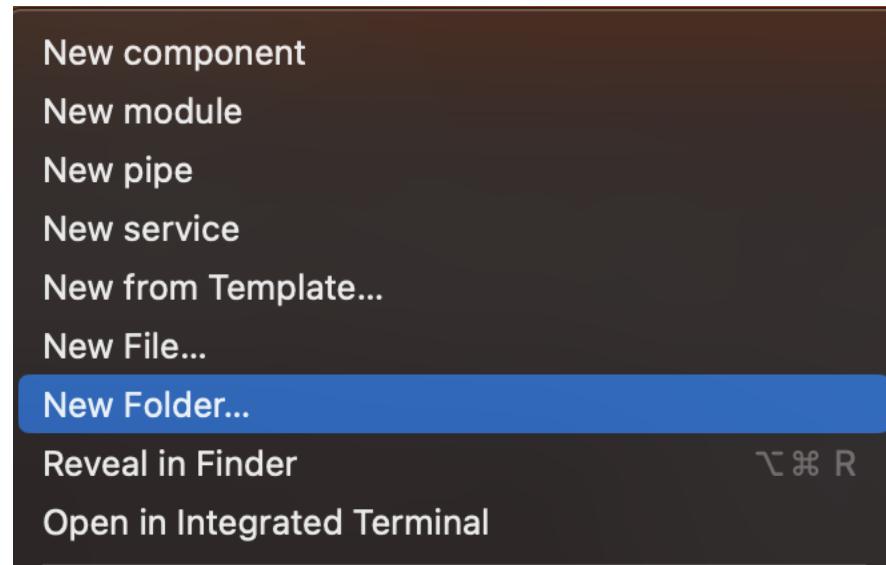
}
```



# Padrão DTO (Data Transfer Object)

src/main/java/com/uneb/labweb/spring\_project/

Criar uma pasta dto



Criar um record CourseDTO.java





# Padrão DTO (Data Transfer Object)

src/main/java/com/uneb/labweb/spring\_project/dto/CourseDTO.java

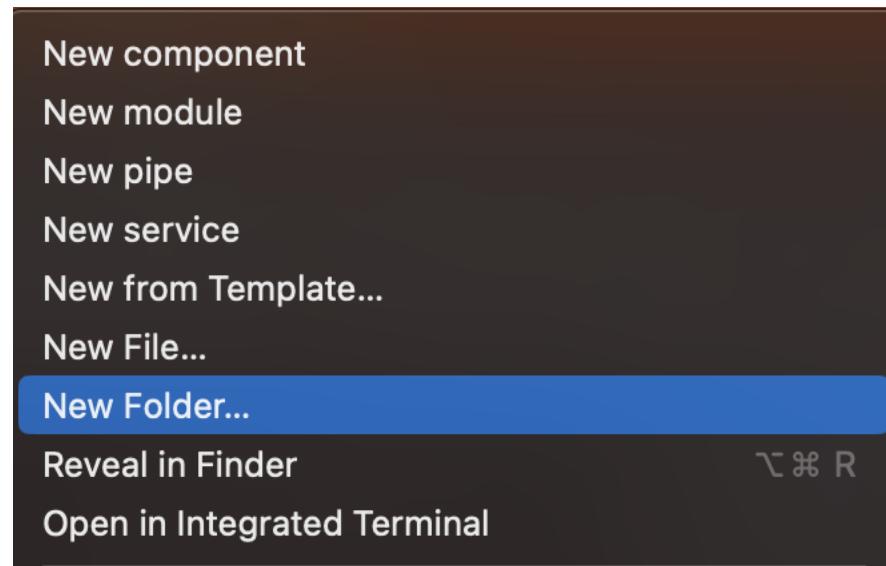
```
public record CourseDTO(  
    Long id,  
    @NotBlank @NotNull @Length(min = 5, max = 100) String name,  
    @NotNull @Length(max = 10) @Pattern(regexp = "Back-end|Front-end")  
    String category){  
}
```



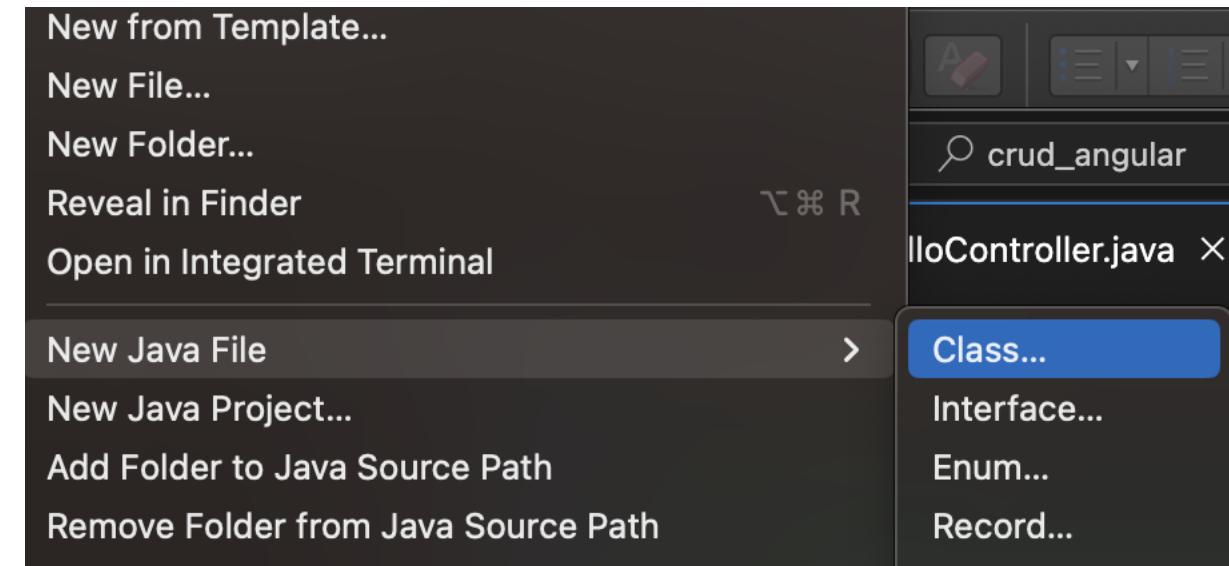
# Padrão DTO: Padrão Mapper

src/main/java/com/uneb/labweb/spring\_project/dto/

Criar uma pasta mapper



Criar um class CourseMapper.java





# Padrão DTO: Padrão Mapper

src/main/java/com/uneb/labweb/spring\_project/dto/mapper/CourseMapper.java

```
@Component
public class CourseMapper {

    public CourseDTO toDTO(Course course) {
        if (course == null){
            return null;
        }
        return new CourseDTO(
            course.getId(),
            course.getName(),
            course.getCategory()
        );
    }
    ...
}
```



# Padrão DTO: Padrão Mapper

src/main/java/com/uneb/labweb/spring\_project/dto/mapper/CourseMapper.java

```
@Component
public class CourseMapper {
...
    public Course toEntity(CourseDTO courseDTO) {
        if (courseDTO == null){
            return null;
        }
        Course course = new Course();
        if (courseDTO.id() != null) {
            course.setId(courseDTO.id());
        }
        course.setName(courseDTO.name());
        course.setCategory(courseDTO.category());
        return course;
    }
}
```



# Padrão DTO: Padrão Mapper

src/main/java/com/uneb/labweb/spring\_project/dto/mapper/CourseMapper.java

```
@Component
public class CourseMapper {
...
    public Course toEntity(CourseDTO courseDTO) {
        if (courseDTO == null){
            return null;
        }
        Course course = new Course();
        if (courseDTO.id() != null) {
            course.setId(courseDTO.id());
        }
        course.setName(courseDTO.name());
        course.setCategory(courseDTO.category());
        return course;
    }
}
```



# Padrão DTO: Padrão Mapper

src/main/java/com/uneb/labweb/spring\_project/service/CourseService.java

```
public class CourseService {  
    ...  
    private final CourseMapper courseMapper;  
  
    public CourseService(CourseRepository courseRepository,  
        CourseMapper courseMapper) {  
        this.courseRepository = courseRepository;  
        this.courseMapper = courseMapper;  
    }  
    ...  
}
```



# Padrão DTO: Padrão Mapper

src/main/java/com/uneb/labweb/spring\_project/service/CourseService.java

```
public class CourseService {  
    ...  
    public List<CourseDTO> list(){  
        return courseRepository.findAll().stream()  
            .map(courseMapper::toDTO)  
            .collect(Collectors.toList());  
    }  
    ...  
}
```



# Padrão DTO: Padrão Mapper

src/main/java/com/uneb/labweb/spring\_project/service/CourseService.java

```
public class CourseService {  
    ...  
    public CourseDTO findById(@NotNull @Positive Long id){  
        return courseRepository.findById(id).map(courseMapper::toDTO)  
            .orElseThrow(() -> new RecordNotFoundException(id));  
    }  
  
    public CourseDTO create(@Valid @NotNull CourseDTO course) {  
        return courseMapper.toDTO(  
            courseRepository.save(courseMapper.toEntity(course))  
        );  
    }  
    ...  
}
```



# Padrão DTO: Padrão Mapper

src/main/java/com/uneb/labweb/spring\_project/service/CourseService.java

```
public class CourseService {  
    ...  
    public CourseDTO update(@NotNull @Positive Long id,  
        @Valid @NotNull CourseDTO course){  
        return courseRepository.findById(id)  
            .map(recordFound -> {  
                recordFound.setName(course.name());  
                recordFound.setCategory(course.category());  
                return courseMapper.toDTO(courseRepository.save(recordFound));  
            })  
            .orElseThrow(() -> new RecordNotFoundException(id));  
    }  
    ...  
}
```



# Padrão DTO: Padrão Mapper

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
public class CourseController {  
    ...  
    @GetMapping  
    public @ResponseBody List<CourseDTO> list(){  
        return courseService.list();  
    }  
  
    @GetMapping("/{id}")  
    public CourseDTO findById(@PathVariable @NotNull @Positive Long id){  
        return courseService.findById(id);  
    }  
    ...  
}
```



# Padrão DTO: Padrão Mapper

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

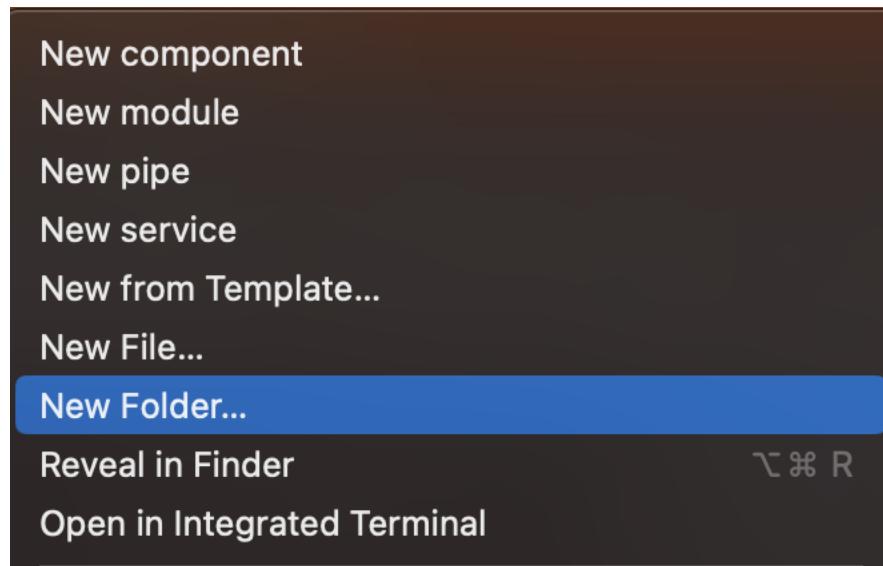
```
public class CourseController {  
    ...  
    @PutMapping("/{id}")  
    public CourseDTO update(@PathVariable @NotNull @Positive Long id,  
                           @RequestBody @Valid @NotNull CourseDTO course){  
        return courseService.update(id, course);  
    }  
    ...  
    @PostMapping  
    @ResponseStatus(code = HttpStatus.ACCEPTED)  
    public CourseDTO create(@RequestBody @Valid @NotNull CourseDTO course) {  
        return courseService.create(course);  
    }  
    ...  
}
```



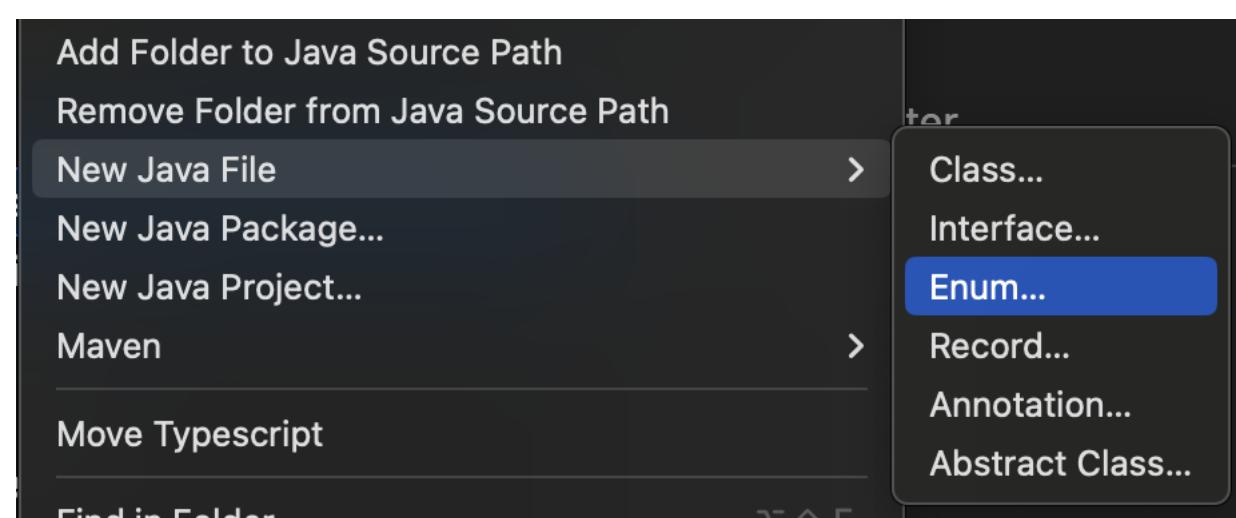
# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/

Criar uma pasta emums



Criar um enum Category.java





# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/enums/Category.java

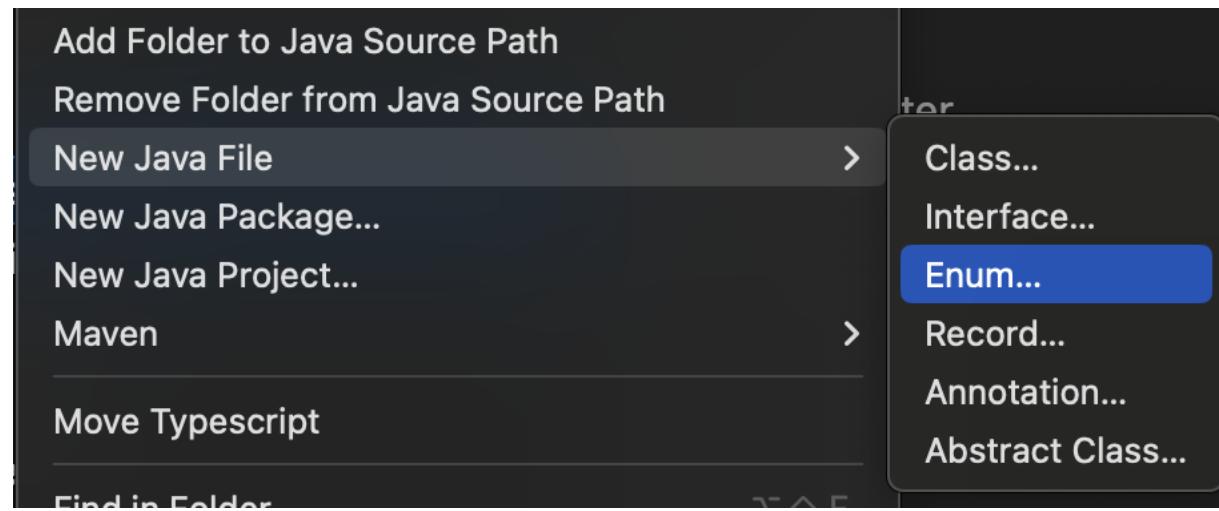
```
public enum Category {  
    BACK_END("Back-end"), FRONT_END("Front-end");  
    private String value;  
  
    Category(String value) {  
        this.value = value;  
    }  
    public String getValue() {  
        return value;  
    }  
    @Override  
    public String toString() {  
        return value;  
    }  
}
```



# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/enums/

Criar um enum Status.java





# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/enums/Status.java

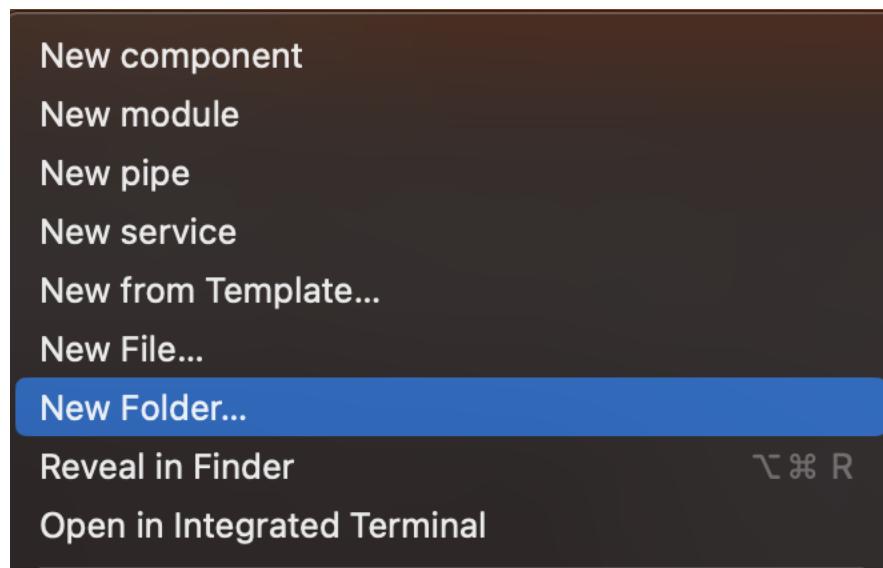
```
public enum Status {  
    ACTIVE("Ativo"), INACTIVE("Inativo");  
    private String value;  
  
    Status(String value) {  
        this.value = value;  
    }  
    public String getValue() {  
        return value;  
    }  
    @Override  
    public String toString() {  
        return value;  
    }  
}
```



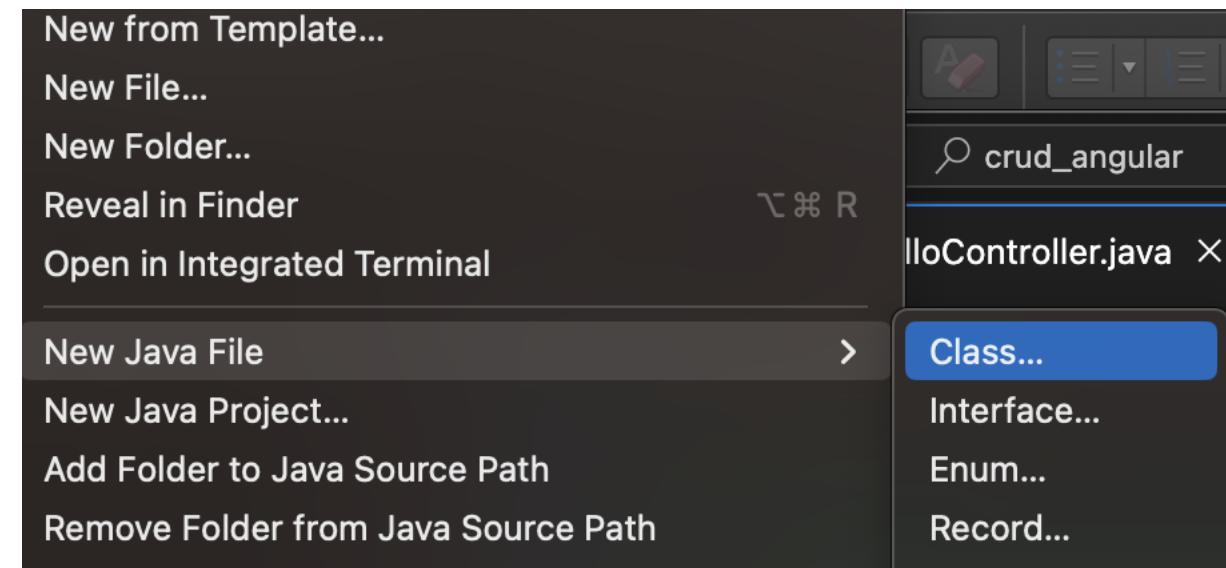
# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/enums/

Criar uma pasta converters



Criar um class CategoryConverter.java





# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/emums/converters/CategoryConverter.java

```
@Converter(autoApply = true)
public class CategoryConverter implements
    AttributeConverter<Category, String> {

    @Override
    public String convertToDatabaseColumn(Category category) {
        if (category == null){
            return null;
        }
        return category.getValue();
    }
    ...
}
```



# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/emums/converter/CategoryConverter.java

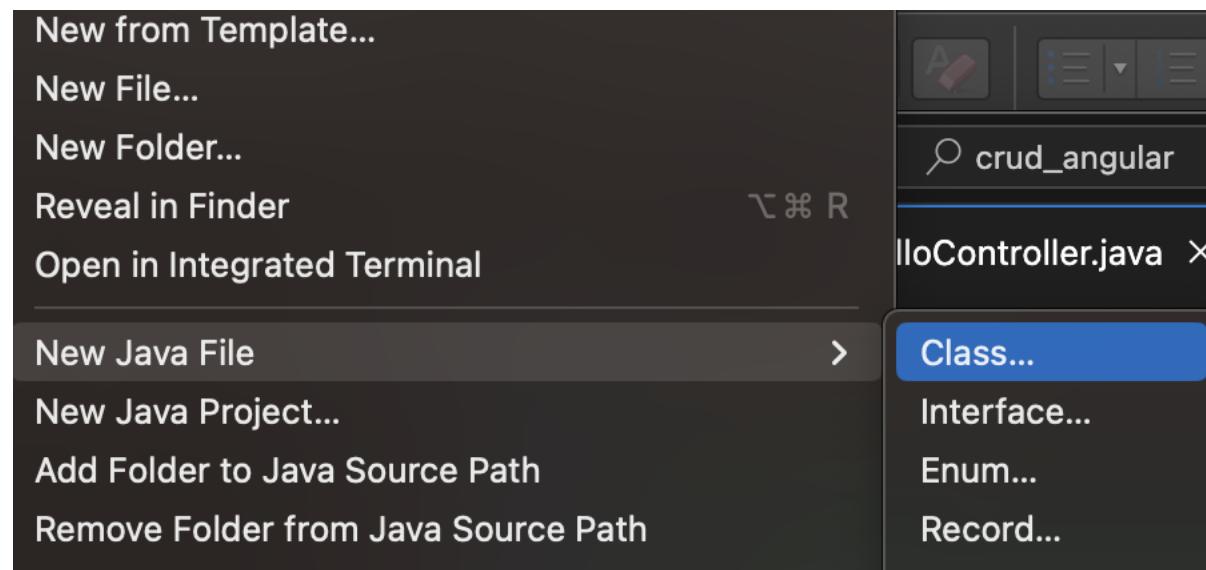
```
@Converter(autoApply = true)
public class CategoryConverter implements
    AttributeConverter<Category, String> {
    ...
    @Override
    public Category convertToEntityAttribute(String value) {
        if (value == null) {
            return null;
        }
        return Stream.of(Category.values())
            .filter(c -> c.getValue().equals(value))
            .findFirst()
            .orElseThrow(IllegalArgumentException::new);
    }
}
```



# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/enums/converter/

Criar uma class StatusConverter.java





# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/emums/converters/StatusConverter.java

```
@Converter(autoApply = true)
public class StatusConverter implements
    AttributeConverter<Status, String> {

    @Override
    public String convertToDatabaseColumn(Status status) {
        if (status == null){
            return null;
        }
        return status.getValue();
    }
    ...
}
```



# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/emums/converter/StatusConverter.java

```
@Converter(autoApply = true)
public class StatusConverter implements
    AttributeConverter<Status, String> {
    ...
    @Override
    public Status convertToEntityAttribute(String value) {
        if (value == null) {
            return null;
        }
        return Stream.of(Status.values())
            .filter(s -> s.getValue().equals(value))
            .findFirst()
            .orElseThrow(IllegalArgumentException::new);
    }
}
```



# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/model/Course.java

```
public class Course {  
    ...  
    @NotNull  
    @Column(length = 10, nullable = false)  
    @Convert(converter = CategoryConverter.class)  
    private Category category;  
  
    @NotNull  
    @Column(length = 10, nullable = false)  
    @Convert(converter = StatusConverter.class)  
    private Status status = Status.ACTIVE;  
}
```



# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/dto/mapper/CourseMapper.java

```
@Component
public class CourseMapper {

    public CourseDTO toDTO(Course course) {
        if (course == null){
            return null;
        }
        return new CourseDTO(
            course.getId(),
            course.getName(),
            course.getCategory().getValue()
        );
    }
    ...
}
```



# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/dto/mapper/CourseMapper.java

```
public class CourseMapper {  
    ...  
    public Category converterCategoryValue(String category) {  
        if (category == null){  
            return null;  
        }  
        return switch (category) {  
            case "Back-end" -> Category.BACK_END;  
            case "Front-end" -> Category.FRONT_END;  
            default -> throw new  
                IllegalArgumentException("Categoria invalida" + category);  
        };  
    }  
}
```



# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/dto/mapper/CourseMapper.java

```
public class CourseMapper {  
    ...  
    public Course toEntity(CourseDTO courseDTO) {  
        if (courseDTO == null){  
            return null;  
        }  
        Course course = new Course();  
        if (courseDTO.id() != null) {  
            course.setId(courseDTO.id());  
        }  
        course.setName(courseDTO.name());  
        course.setCategory(converterCategoryValue(courseDTO.category()));  
        return course;  
    }  
    ...  
}
```



# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/service/CourseService.java

```
public class CourseService {  
    ...  
    public CourseDTO update(@NotNull @Positive Long id,  
        @Valid @NotNull CourseDTO course){  
        return courseRepository.findById(id)  
            .map(recordFound -> {  
                recordFound.setName(course.name());  
                recordFound.setCategory(this.courseMapper.converterCategoryValue(  
                    course.category()));  
                return courseMapper.toDTO(courseRepository.save(recordFound));  
            })  
            .orElseThrow(() -> new RecordNotFoundException(id));  
    }  
    ...  
}
```



# Persistência de Enums no JPA

src/main/java/com/uneb/labweb/spring\_project/SpringProjectApplication.java

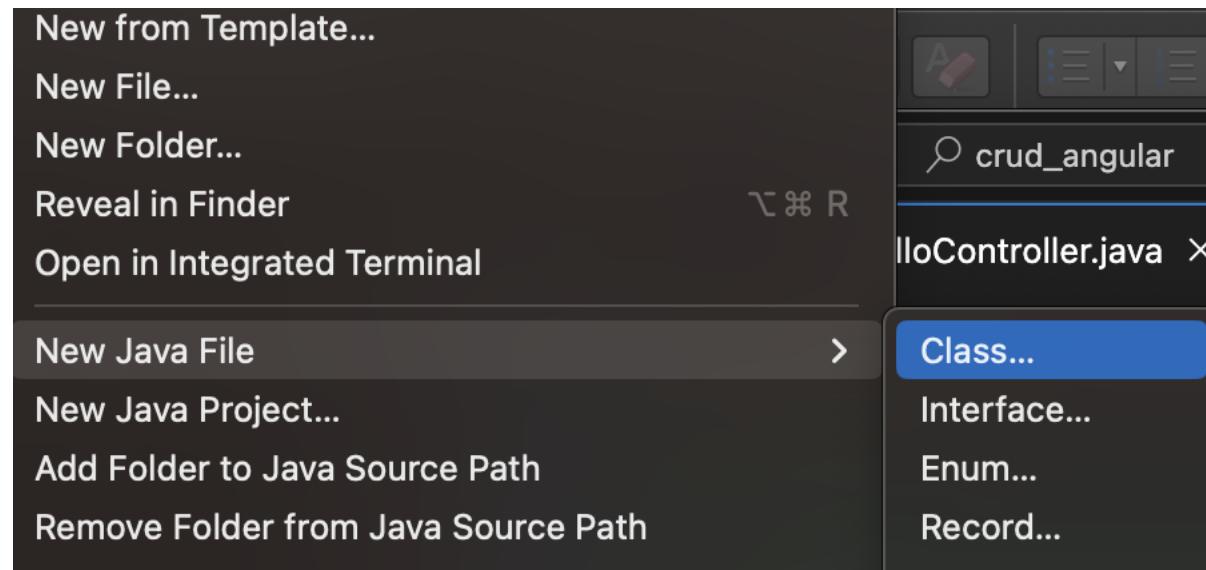
```
public class SpringProjectApplication {  
    ...  
    @Bean  
    CommandLineRunner initDatabase(CourseRepository courseRepository){  
        return args -> {  
            courseRepository.deleteAll();  
            Course c = new Course();  
            c.setName("Curso Angular");  
            c.setCategory(Category.FRONT_END);  
            courseRepository.save(c);  
        };  
    }  
    ...  
}
```



# Criando Aulas para nossos Cursos

src/main/java/com/uneb/labweb/spring\_project/model/

Criar uma class Lesson.java





# Criando Aulas para nossos Cursos

src/main/java/com/uneb/labweb/spring\_project/model/Lesson.java

```
@Data  
@Entity  
public class Lesson {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.AUTO)  
    private Long id;  
  
    @Column(length = 100, nullable = false)  
    private String name;  
  
    @Column(length = 100, nullable = false)  
    private String linkLesson;  
  
    @ManyToOne(fetch = FetchType.LAZY, optional=false)  
    @JoinColumn(name = "course_id", nullable = false)  
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)  
    private Course course;  
}
```



# Criando Aulas para nossos Cursos

src/main/java/com/uneb/labweb/spring\_project/model/Course.java

```
...
public class Course {
    ...
    @OneToMany(cascade= CascadeType.ALL, orphanRemoval=true, mappedBy="course")
    private List<Lesson> lessons = new ArrayList<>();
}

}
```



# Criando Aulas para nossos Cursos

src/main/java/com/uneb/labweb/spring\_project/SpringProjectApplication.java

```
...
public class SpringProjectApplication {
...
    CommandLineRunner initDatabase(CourseRepository courseRepository){
        return args -> {
            courseRepository.deleteAll();
            Course c = new Course();
            c.setName("Curso Angular");
            c.setCategory(Category.FRONT_END);

            Lesson l = new Lesson();
            l.setName("Introdução ao Angular");
            l.setLinkLesson("https://www.uneb.br");
            l.setCourse(c);
            c.getLessons().add(l);

            ...
        };
    }
}
```



# Criando Aulas para nossos Cursos

src/main/java/com/uneb/labweb/spring\_project/SpringProjectApplication.java

```
...
public class SpringProjectApplication {
...
    CommandLineRunner initDatabase(CourseRepository courseRepository){
        return args -> {
            ...
            l = new Lesson();
            l.setName("Instalando e Configurando Angular");
            l.setLinkLesson("https://www.uneb.br");
            l.setCourse(c);
            c.getLessons().add(l);

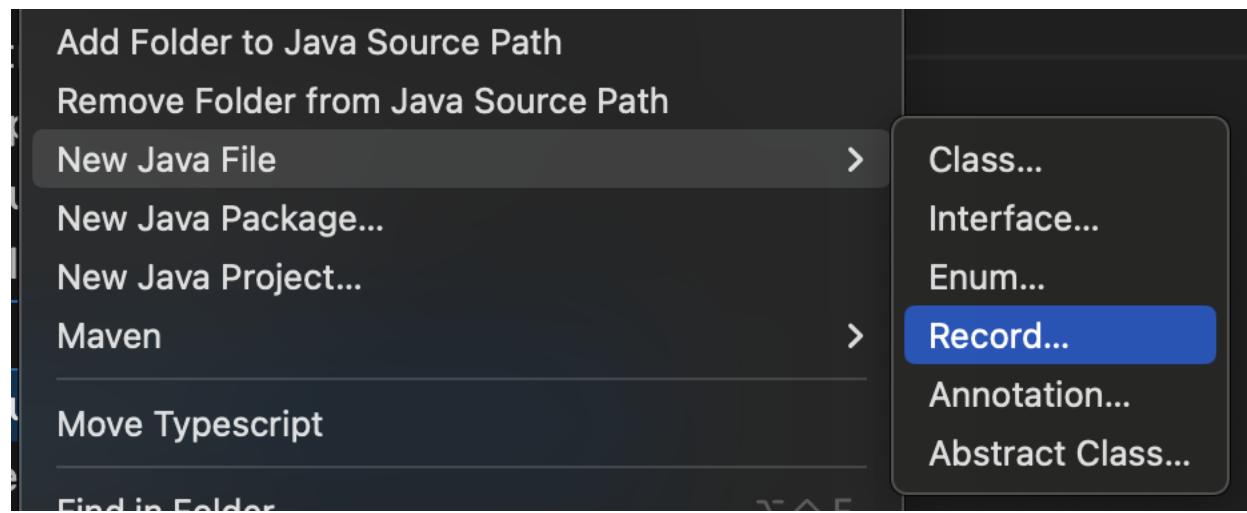
            courseRepository.save(c);
        };
    }
}
```



# DTO de Aula

src/main/java/com/uneb/labweb/spring\_project/dto/

Criar um record LessonDTO.java





# DTO de Aula

src/main/java/com/uneb/labweb/spring\_project/dto/LessonDTO.java

```
public record LessonDTO(  
    Long id,  
    String name,  
    String linkLesson) {  
}
```



# DTO de Aula

src/main/java/com/uneb/labweb/spring\_project/dto/CourseDTO.java

```
public record CourseDTO(  
    Long id,  
    @NotBlank @NotNull @Length(min = 5, max = 100) String name,  
    @NotNull @Length(max = 10) @Pattern(regexp = "Back-end|Front-end") String category,  
    List<LessonDTO> lessons)  
{  
}
```



# DTO de Aula

src/main/java/com/uneb/labweb/spring\_project/dto/mapper/CourseMapper.java

```
public class CourseMapper {

    public CourseDTO toDTO(Course course) {
        ...
        List<LessonDTO> lessons = course.getLessons()
            .stream()
            .map(lesson -> new LessonDTO(
                lesson.getId(), lesson.getName(), lesson.getLinkLesson()))
            .collect(Collectors.toList());
        return new CourseDTO(
            course.getId(),
            course.getName(),
            course.getCategory().getValue(),
            lessons
        );
    }
    ...
}
```

# Criar uma interface Aulas



ng g interface courses/model/lesson

```
vagner@Vagners-MBP angular-project % ng g interface courses/model/lesson
CREATE src/app/courses/model/lesson.ts (28 bytes)
```

src/app/courses/model/lesson.ts

```
export interface Lesson {
  id: string;
  name: string;
  linkLesson: string;
}
```

src/app/courses/model/course.ts

```
import { Lesson } from './lesson';

export interface Course {
  id: string;
  name: string;
  category: string;
  lessons?: Lesson[];
}
```

# Criando um FormArray



src/app/courses/guards/course.resolver.ts

```
...
export class CourseResolver {
  ...
  resolve(route: ActivatedRouteSnapshot,
    state: RouterStateSnapshot): Observable<Course> {
  ...
  return of({ id: "", name: "", category: "", lessons: [] });
}
...
}
```

# Criando um FormArray



src/app/courses/course-form/course-form.component.ts

```
...
export class CourseFormComponent implements OnInit{
  form!: FormGroup;
  ...

  getLessonsFormArray(){
    return (<UntypedFormArray>this.form.get('lessons')).controls;
  }
  ...
}
```

# Criando um FormArray



src/app/courses/course-form/course-form.component.ts

```
...
export class CourseFormComponent implements OnInit{
  ...
  private createLesson(lesson : Lesson =
    { id: "", name: "", linkLesson: " " }){
    return this.formBuilder.group({
      id: [lesson.id],
      name: [lesson.name],
      linkLesson: [lesson.linkLesson]
    })
  }
  ...
}
```

# Criando um FormArray



src/app/courses/course-form/course-form.component.ts

```
...
export class CourseFormComponent implements OnInit{
  ...
  private retrieveLessons(course: Course){
    const lessons = [];
    if (course?.lessons){
      course.lessons?.forEach(lesson =>
        lessons.push(this.createLesson(lesson)));
    }else{
      lessons.push(this.createLesson());
    }
    return lessons;
  }
  ...
}
```

# Criando um FormArray



src/app/courses/course-form/course-form.component.ts

```
...
export class CourseFormComponent implements OnInit{
  ...
  ngOnInit(): void {
    ...
    this.form = this.formBuilder.group({
      id: [course.id],
      name: [course.name, [Validators.required,
        Validators.minLength(5),
        Validators.maxLength(100)]],
      category: [course.category, [Validators.required]],
      lessons: this.formBuilder.array(this.retrieveLessons(course))
    });
  }
  ...
}
```

# Criando um FormArray



src/app/courses/course-form/course-form.component.html

```
<form [formGroup]="form" class="min-width">
...
<mat-toolbar>
  <span>Aulas</span>
  <button mat-icon-button color="accent">
    <mat-icon aria-label="Adicionar uma aula">add</mat-icon>
  </button>
</mat-toolbar>
<table style="width: 100%;" formArrayName="lessons"
  *ngFor="let lesson of getLessonsFormArray(); let i = index">
  <tbody [formGroupName]="i">
    <tr>
      <td style="width: 65%;">
        <mat-form-field class="fill-width">
...

```

# Criando um FormArray



src/app/courses/course-form/course-form.component.html

```
<form [formGroup]="form" class="min-width">
...
    <mat-label>Título da aula</mat-label>
    <input matInput formControlName="name">
    </mat-form-field>
</td>
<td style="width: 35%;">
    <mat-form-field class="fill-width">
        <mat-label>Link da aula</mat-label>
        <input matInput formControlName="linkLesson">
    </mat-form-field>
</td>
</tr>
</tbody>
</table>
</form>
```

# Criando um FormArray



src/app/courses/course-form/course-form.component.scss

```
...
.form-array-error{
  text-align: left;
}
```

# Adicionando e Removendo Aulas



src/app/courses/course-form/course-form.component.ts

```
...
export class CourseFormComponent implements OnInit{
  ...
  addNewLesson(){
    const lessons = this.form.get('lessons') as UntypedFormArray;
    lessons.push(this.createLesson());
  }

  removeLesson(index: number){
    const lessons = this.form.get('lessons') as UntypedFormArray;
    lessons.removeAt(index);
  }
  ...
}
```

# Adicionando e Removendo Aulas



src/app/courses/course-form/course-form.component.html

```
<form [formGroup]="form" class="min-width">
...
<mat-toolbar>
  <span>Aulas</span>
  <button mat-icon-button color="accent" (click)="addNewLesson()">
    <mat-icon aria-label="Adicionar uma aula">add</mat-icon>
  </button>
</mat-toolbar>
...
</form>
...
```

# Adicionando e Removendo Aulas



src/app/courses/course-form/course-form.component.html

```
<form [formGroup]="form" class="min-width">
...
<td>
  <button mat-icon-button color="warn" (click)="removeLesson(i)">
    <mat-icon aria-label="Remover a aula">delete</mat-icon>
  </button>
</td>
...
</form>
...
```

# Validando o Formulário

## Aulas



ng g s shared/form/form-utils

```
vagner@Vagners-MBP angular-project % ng g s shared/form/form-utils
CREATE src/app/shared/form/form-utils.service.spec.ts (373 bytes)
CREATE src/app/shared/form/form-utils.service.ts (138 bytes)
```

src/app/shared/form/form-utils.service.ts

```
import { UntypedFormArray, UntypedFormControl, UntypedFormGroup } from '@angular/forms';
...
export class FormUtilsService {

  constructor() {}

  ...
}
```

# Validando o Formulário Aulas



src/app/shared/form/form-utils.service.ts

```
export class FormUtilsService {  
...  
    validateAllFormFields(formGroup: UntypedFormGroup | UntypedFormArray){  
        Object.keys(formGroup.controls).forEach(field => {  
            const control = formGroup.get(field);  
            if (control instanceof UntypedFormControl){  
                control.markAsTouched({ onlySelf: true });  
            }else if (control instanceof UntypedFormGroup ||  
                control instanceof UntypedFormArray){  
                control.markAsTouched({ onlySelf: true });  
                this.validateAllFormFields(control);  
            }  
        });  
    }  
...  
}
```

# Validando o Formulário

## Aulas



src/app/shared/form/form-utils.service.ts

```
export class FormUtilsService {  
...  
  getMessage(formGroup: UntypedFormGroup, fieldName: string){  
    const field = formGroup.get(fieldName) as UntypedFormControl;  
    return this.getMessageFromField(field);  
  }  
  
  getFormArrayFieldErrorMessage(formGroup: UntypedFormGroup,  
    formArrayName: string, fieldName: string, index: number){  
    const formArray = formGroup.get(formArrayName) as UntypedFormArray;  
    const field = formArray.controls[index].get(fieldName) as UntypedFormControl;  
    return this.getMessageFromField(field);  
  }  
...  
}
```

# Validando o Formulário

## Aulas

src/app/shared/form/form-utils.service.ts



```
export class FormUtilsService {  
...  
  getErrorMessageFromField(field: UntypedFormControl){  
    if (field?.hasError('required')){  
      return 'Campo obrigatorio';  
    }  
    if (field?.hasError('minlength')){  
      const requiredLength : number = field.errors ?  
        field.errors['minlength']['requiredLength'] : 5;  
      return `Tamanho mínimo precisa ser de ${requiredLength} caracteres.`;  
    }  
    if (field?.hasError('maxlength')){  
      const requiredLength : number = field.errors ?  
        field.errors['maxlength']['requiredLength'] : 100;  
      return `Tamanho máximo excedido de ${requiredLength} caracteres.`;  
    }  
    return 'Campo invalido!';  
  }  
...  
}
```

# Validando o Formulário Aulas



src/app/shared/form/form-utils.service.ts

```
export class FormUtilsService {  
...  
  isFormArrayRequired(formGroup: UntypedFormGroup, formArrayName: string){  
    const formArray = formGroup.get(formArrayName) as UntypedFormArray;  
    return !formArray.valid &&  
      formArray.hasError('required') &&  
      formArray.touched;  
  }  
}
```

# Validando o Formulário Aulas



src/app/courses/course-form/course-form.component.ts

```
export class CourseFormComponent implements OnInit{  
  ...  
  constructor(private formBuilder: NonNullableFormBuilder,  
  private service: CoursesService,  
  private snackBar: MatSnackBar,  
  private location : Location,  
  private route : ActivatedRoute,  
  public formUtils: FormUtilsService){  
  }  
  ...  
}
```

# Validando o Formulário

## Aulas



src/app/courses/course-form/course-form.component.ts

```
export class CourseFormComponent implements OnInit{  
  ...  
  onSubmit(){  
    if (this.form.valid){  
      this.service.save(this.form.value).subscribe(  
        result => this.onSuccess(),  
        error => this.onError()  
      );  
    }else{  
      this.formUtils.validateAllFormFields(this.form);  
    }  
  }  
  ...  
}
```

# Validando o Formulário Aulas



src/app/courses/course-form/course-form.component.ts

```
export class CourseFormComponent implements OnInit{  
  ...  
  ngOnInit(): void {  
    ...  
    this.form = this.formBuilder.group({  
      id: [course.id],  
      name: [course.name, [Validators.required,  
        Validators.minLength(5),  
        Validators.maxLength(100)]],  
      category: [course.category, [Validators.required]],  
      lessons: this.formBuilder.array(  
        this.retrieveLessons(course), Validators.required)  
    });  
  }  
  ...  
}
```

# Validando o Formulário Aulas



src/app/courses/course-form/course-form.component.ts

```
...
export class CourseFormComponent implements OnInit{

  ...
  private createLesson(lesson : Lesson =
    { id: "", name: "", linkLesson: " " }){
    return this.formBuilder.group({
      id: [lesson.id],
      name: [lesson.name, [Validators.required, Validators.minLength(5),
        Validators.maxLength(100)]],
      linkLesson: [lesson.linkLesson, [Validators.required,
        Validators.minLength(10), Validators.maxLength(100)]]})
  }
  ...
}
```

# Validando o Formulário

## Aulas



src/app/courses/course-form/course-form.component.html

```
...
<form [formGroup]="form" class="min-width">
  ...
  <mat-form-field class="fill-width">
    <mat-label>Nome do Curso</mat-label>
    <input matInput placeholder="Nome" formControlName="name" #name>
    <mat-hint align="end">{{name.value.length || 0}}/100</mat-hint>
    <mat-error *ngIf="form.get('name')?.invalid">{{ formUtils.getErrorMessage(form, "name") }}</mat-error>
  </mat-form-field>
  ...
</form>
...
```

# Validando o Formulário

## Aulas



src/app/courses/course-form/course-form.component.html

```
...
<form [formGroup]="form" class="min-width">
  ...
  <mat-select formControlName="category">
    <mat-option value="Front-end">Front-end</mat-option>
    <mat-option value="Back-end">Back-end</mat-option>
  </mat-select>
  <mat-error *ngIf="form.get('category')?.invalid">{{ formUtils.getErrorMessag...</mat-error>
  ...
</form>
...
```

# Validando o Formulário Aulas



src/app/courses/course-form/course-form.component.html

```
...
</mat-toolbar>

<mat-error *ngIf="formUtils.isFormArrayRequired(form, 'lessons')"
  class="form-array-error">
  Adicione uma aula
</mat-error>

<table...>
...
```

# Validando o Formulário Aulas



src/app/courses/course-form/course-form.component.html

```
...
<td style="width: 65%;">
  <mat-form-field class="fill-width">
    <mat-label>Título da aula</mat-label>
    <input matInput formControlName="name"/>
    <mat-error *ngIf="getLessonsFormArray()[i].get('name')">>{{  

      formUtils.getFormArrayFieldErrorMessage(form, "lessons", "name", i)
    }}</mat-error>
  </mat-form-field>
</td>
...
```

# Validando o Formulário Aulas



src/app/courses/course-form/course-form.component.html

```
...
<td style="width: 35%;">
  <mat-form-field class="fill-width">
    <mat-label>Link da aula</mat-label>
    <input matInput formControlName="linkLesson">
    <mat-error *ngIf="getLessonsFormArray()[i].get('linkLesson')">{{{
      formUtils.getFormArrayFieldErrorMessage(
        form, "lessons", "linkLesson", i)
    }}</mat-error>
  </mat-form-field>
</td>
...
...
```



# CRUD em aula

src/main/java/com/uneb/labweb/spring\_project/dto/mapper/CourseMapper.java

```
public class CourseMapper {

    public CourseDTO toEntity(Course course) {
        ...
        List<Lesson> lessons = courseDTO.lessons().stream().map((lessonDTO) -> {
            var lesson = new Lesson();
            lesson.setId(lessonDTO.id());
            lesson.setName(lessonDTO.name());
            lesson.setLinkLesson(lessonDTO.linkLesson());
            lesson.setCourse(course);
            return lesson;
        }).collect(Collectors.toList());
        course.setLessons(lessons);
        return course;
    }
    ...
}
```



# CRUD em aula

src/main/java/com/uneb/labweb/spring\_project/service/CourseService.java

```
public class CourseService {  
  
    public CourseDTO update(@NotNull @Positive Long id,  
                           @Valid @NotNull CourseDTO courseDTO){  
        return courseRepository.findById(id)  
            .map(recordFound -> {  
                Course course = courseMapper.toEntity(courseDTO);  
                recordFound.setName(courseDTO.name());  
                recordFound.setCategory(this.courseMapper.converterCategoryValue(  
                    courseDTO.category()));  
                recordFound.getLessons().clear();  
                course.getLessons().forEach(lesson ->  
                    recordFound.getLessons().add(lesson));  
                return courseMapper.toDTO(courseRepository.save(recordFound));  
            })  
            .orElseThrow(() -> new RecordNotFoundException(id));  
    }  
    ...  
}
```



# Refatorando Projeto

## src/pom.xml (remoção)

```
...
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <optional>true</optional>
</dependency>
...
```



# Refatorando Projeto

src/main/java/com/uneb/labweb/spring\_project/model/Lesson.java

src/main/java/com/uneb/labweb/spring\_project/model/Course.java

```
import lombok.Data;
```

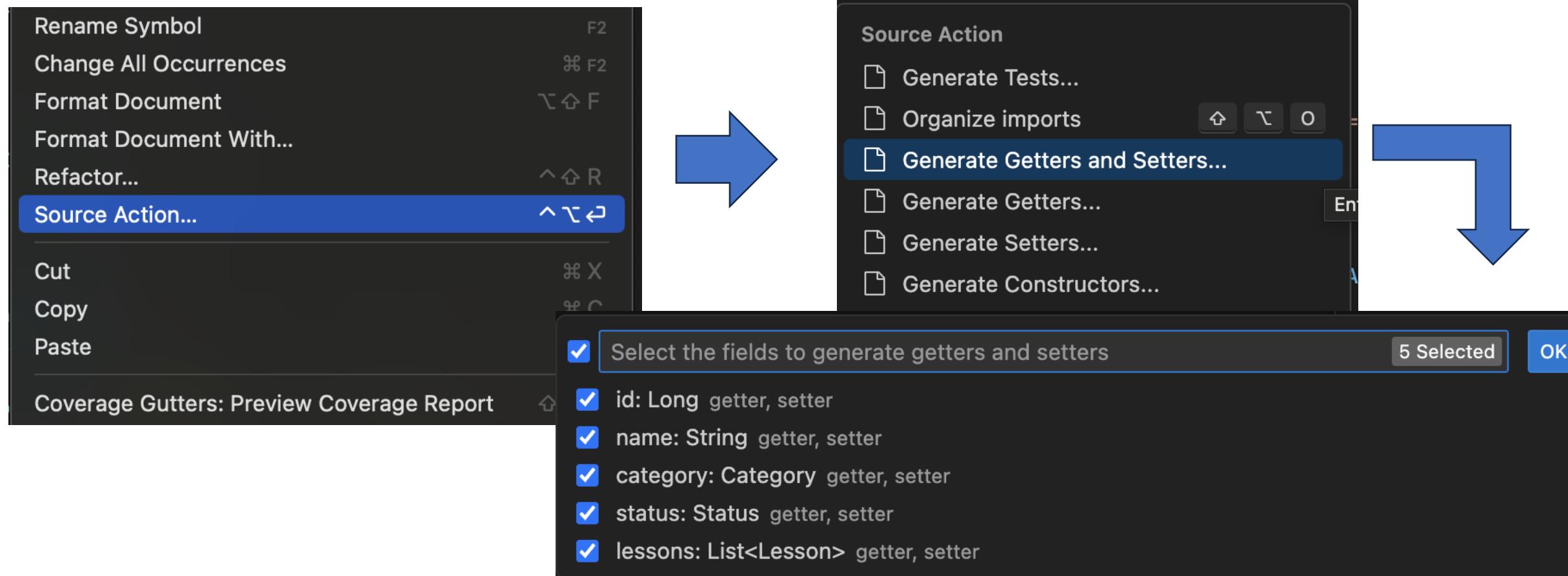
```
@Data
```



# Criando Getters e Setters

src/main/java/com/uneb/labweb/spring\_project/model/Lesson.java

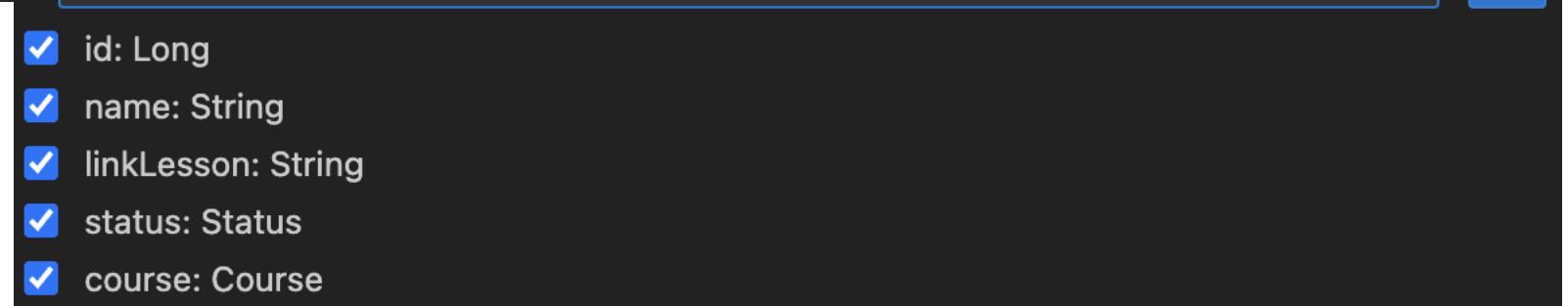
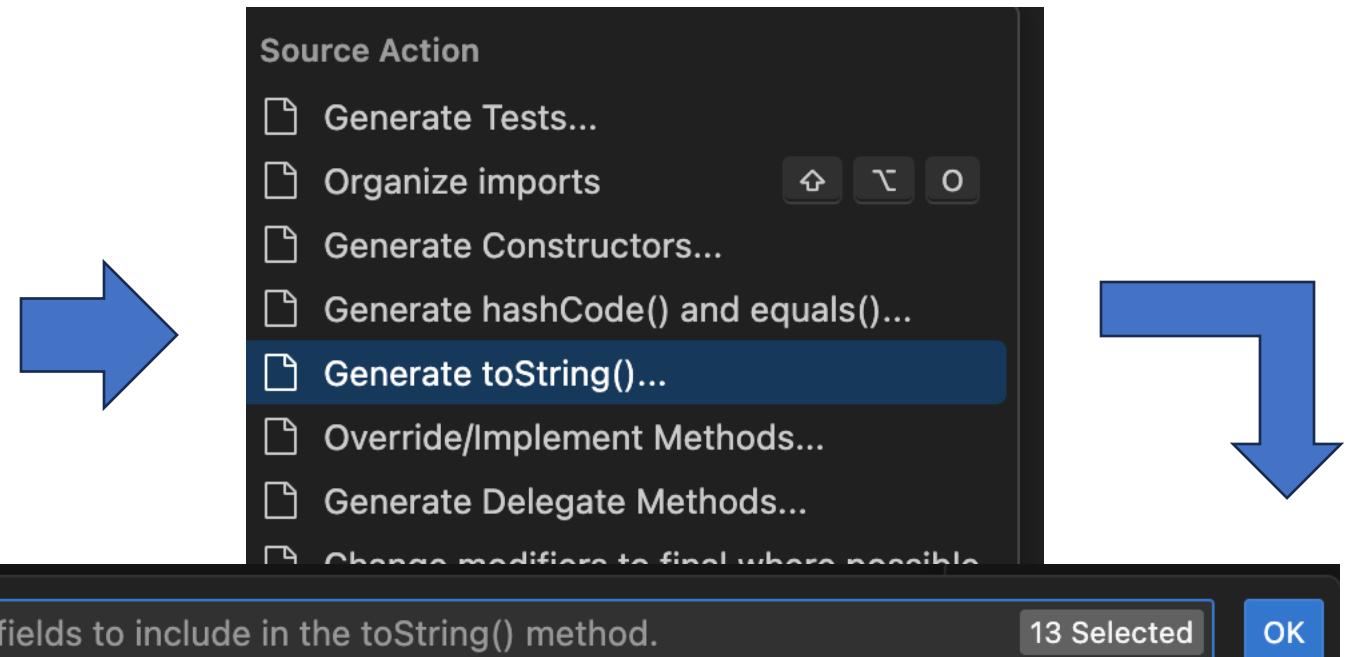
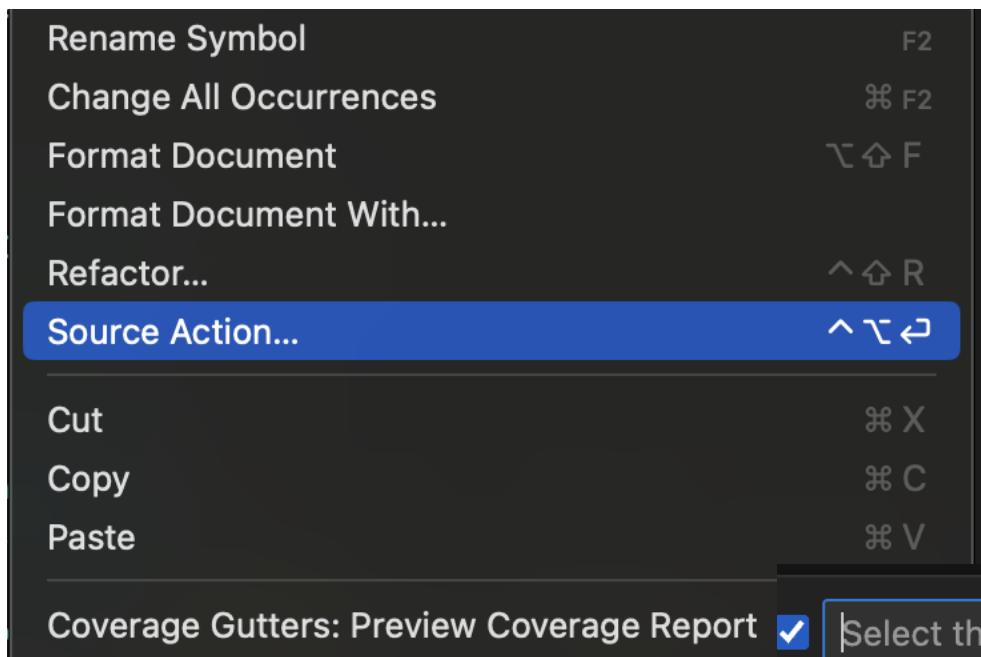
src/main/java/com/uneb/labweb/spring\_project/model/Course.java





# Criando `toString`

`src/main/java/com/uneb/labweb/spring_project/model/Lesson.java`





# Adicionando Validações em Aulas

src/main/java/com/uneb/labweb/spring\_project/model/Lesson.java

```
public class Lesson {  
    ...  
    @NotBlank  
    @NotNull  
    @Length(min = 5, max = 100)  
    @Column(length = 100, nullable = false)  
    private String name;  
  
    @NotBlank  
    @NotNull  
    @Length(min = 5, max = 100)  
    @Column(length = 100, nullable = false)  
    private String linkLesson;  
    ...  
}
```



# Adicionando Validações em Cursos e Aulas

src/main/java/com/uneb/labweb/spring\_project/model/Lesson.java

```
public class Lesson {  
    ...  
    @NotNull  
    @ManyToOne(fetch = FetchType.LAZY, optional=false)  
    @JoinColumn(name = "course_id", nullable = false)  
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)  
    private Course course;  
    ...  
}
```



# Adicionando Validações em Cursos e Aulas

src/main/java/com/uneb/labweb/spring\_project/model/Course.java

```
public class Course {  
    ...  
    @NotNull  
    @NotEmpty  
    @Valid  
    @OneToMany(cascade= CascadeType.ALL, orphanRemoval=true, mappedBy="course")  
    private List<Lesson> lessons = new ArrayList<>();  
    ...  
}
```



# Adicionando Validações em Cursos e Aulas

src/main/java/com/uneb/labweb/spring\_project/dto/CourseDTO.java

```
public record CourseDTO(  
    Long id,  
    @NotBlank @NotNull @Length(min = 5, max = 100) String name,  
    @NotNull @Length(max = 10) @ValueOfEnum(enumClass = Category.class) String category,  
    @NotNull @NotEmpty @Valid List<LessonDTO> lessons)  
{  
}
```



# Adicionando Validações em Cursos e Aulas

src/main/java/com/uneb/labweb/spring\_project/dto/LessonDTO.java

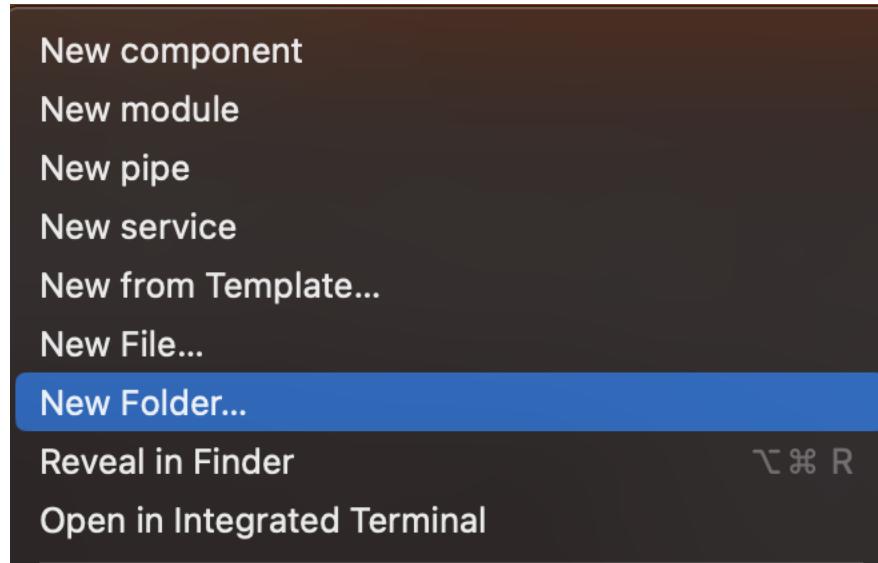
```
public record LessonDTO(  
    Long id,  
    @NotNull @NotBlank @Length(min = 5, max = 100) String name,  
    @NotNull @NotBlank @Length(min = 5, max = 100) String linkLesson)  
{}
```

# Adicionando Validações em Cursos e Aulas

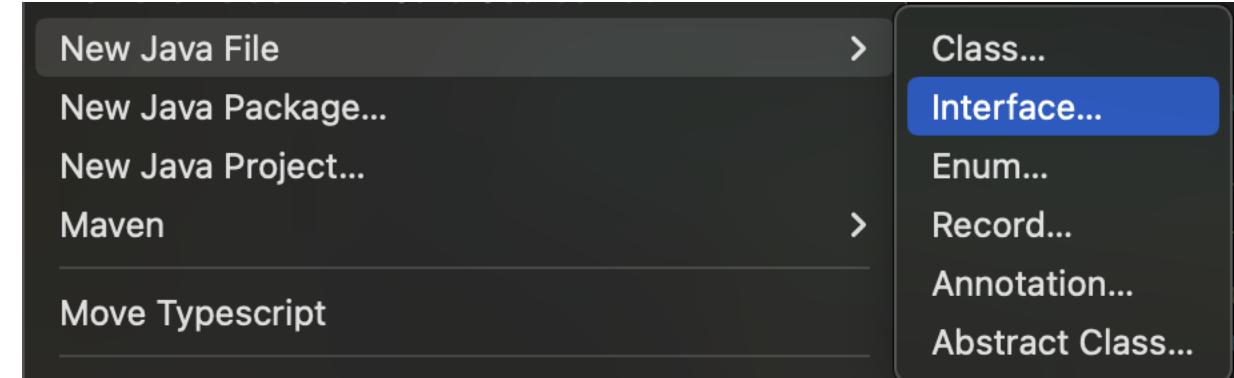


src/main/java/com/uneb/labweb/spring\_project/enums/

Criar uma pasta validation



Criar um interface ValueOfEnum.java

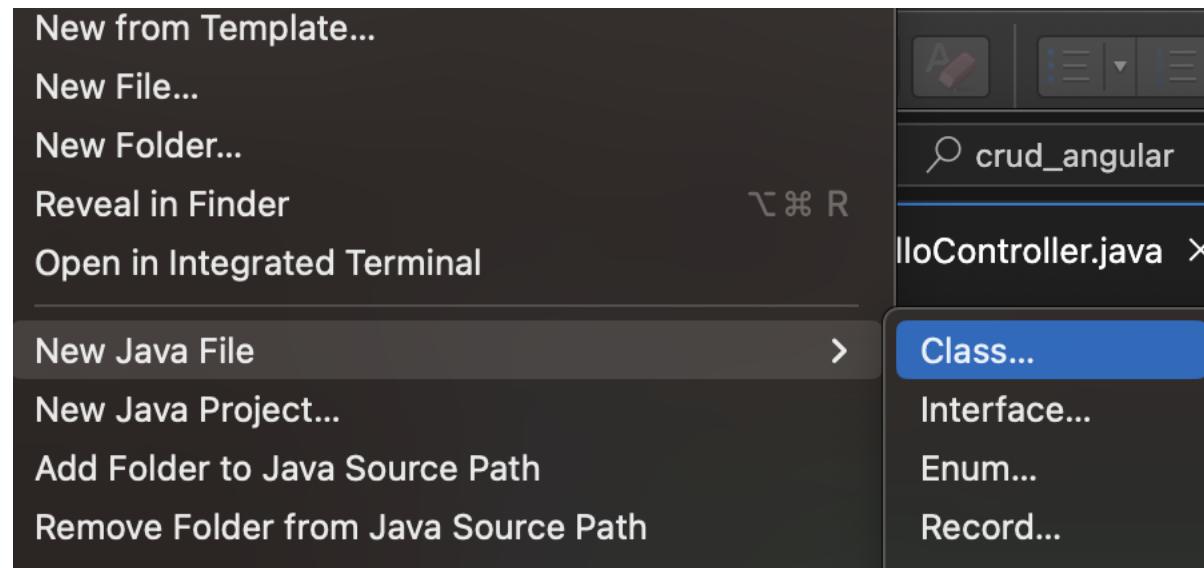


# Adicionando Validações em Cursos e Aulas



src/main/java/com/uneb/labweb/spring\_project/enum/validation

Criar um class ValueOfEnumValidator.java





# Adicionando Validações em Cursos e Aulas

src/main/java/com/uneb/labweb/spring\_project/emums/validation/ValueOfEnum.java

```
@Target({ METHOD, FIELD, ANNOTATION_TYPE, CONSTRUCTOR, PARAMETER, TYPE_USE })
@Retention(RUNTIME)
@Documented
@Constraint(validatedBy = ValueOfEnumValidator.class)
public @interface ValueOfEnum {
    Class<? extends Enum<?>> enumClass();

    String message() default "must be any of enum {enumClass}";

    Class<?>[] groups() default {};

    Class<? extends Payload>[] payload() default {};
}
```



# Adicionando Validações em Cursos e Aulas

src/main/java/com/uneb/labweb/spring\_project/emums/validation/ValueOfEnumValidator.java

```
public class ValueOfEnumValidator implements
    ConstraintValidator<ValueOfEnum, CharSequence> {
    private List<String> acceptedValues;

    @Override
    public void initialize(ValueOfEnum annotation) {
        acceptedValues = Stream.of(annotation.enumClass().getEnumConstants())
            .map(Enum::toString)
            .collect(Collectors.toList());
    }

    @Override
    public boolean isValid(CharSequence value, ConstraintValidatorContext context) {
        if (value == null) {
            return true;
        }
        return acceptedValues.contains(value.toString());
    }
}
```



# Tratamento de Erros da API

src/main/java/com/uneb/labweb/spring\_project/controller/ApplicationControllerAdvice.java

```
public class ApplicationControllerAdvice() {  
    ...  
    @ExceptionHandler(MethodArgumentNotValidException.class)  
    @ResponseStatus(HttpStatus.BAD_REQUEST)  
    public String handleMethodArgumentNotValidException(  
        MethodArgumentNotValidException ex) {  
        return ex.getBindingResult().getFieldErrors().stream()  
            .map(error -> error.getField() + " " + error.getDefaultMessage())  
            .reduce("", (acc, error) -> acc + error + "\n");  
    }  
    @ExceptionHandler(ConstraintViolationException.class)  
    @ResponseStatus(HttpStatus.BAD_REQUEST)  
    public String handleConstraintViolationException(ConstraintViolationException ex) {  
        return ex.getConstraintViolations().stream()  
            .map(error -> error.getPropertyPath() + " " + error.getMessage())  
            .reduce("", (acc, error) -> acc + error + "\n");  
    }  
    ...  
}
```



# Tratamento de Erros da API

src/main/java/com/uneb/labweb/spring\_project/controller/ApplicationControllerAdvice.java

```
public class ApplicationControllerAdvice() {  
    ...  
    @SuppressWarnings("null")  
    @ExceptionHandler(MethodArgumentTypeMismatchException.class)  
    @ResponseStatus(HttpStatus.BAD_REQUEST)  
    public String handleMethodArgumentTypeMismatchException(  
        MethodArgumentTypeMismatchException ex){  
        if (ex != null && ex.getRequiredType() != null) {  
            String type = ex.getRequiredType().getName();  
            String[] typeParts = type.split("\\.");  
            String typeName = typeParts[typeParts.length - 1];  
            return ex.getName() + " should be of type " + typeName;  
        }  
        return "Argument type not valid";  
    }  
}
```



# Exclusão logica em Aulas

src/main/java/com/uneb/labweb/spring\_project/model/Lesson.java

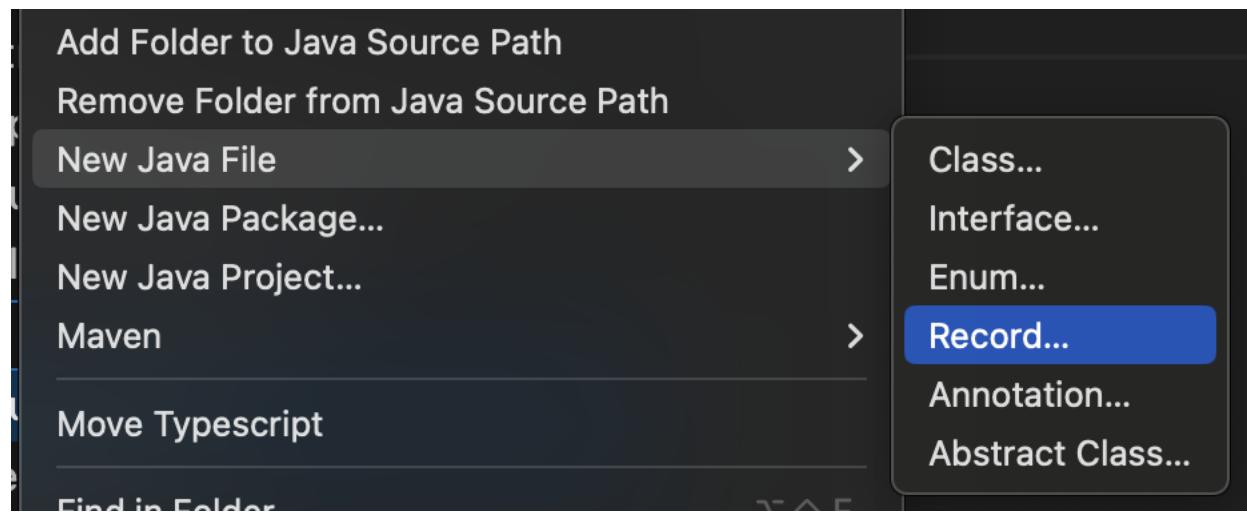
```
@SQLDelete(sql = "UPDATE Lesson SET status = 'Inativo' WHERE id = ?")
@SQLRestriction("status = 'Ativo'")
public class Lesson {
    ...
    @NotNull
    @Column(length = 10, nullable = false)
    @Convert(converter = StatusConverter.class)
    private Status status = Status.ACTIVE;
    ...
}
```



# Paginação de Cursos

src/main/java/com/uneb/labweb/spring\_project/dto/

Criar um record CoursePageDTO.java





# Paginação de Cursos

src/main/java/com/uneb/labweb/spring\_project/dto/CoursePageDTO.java

```
public record CoursePageDTO(  
    List<CourseDTO> courses, long totalElements, int totalPages) {  
}
```



# Paginação de Cursos

src/main/java/com/uneb/labweb/spring\_project/service/CourseService.java

```
public class CourseService {  
    ...  
    public CoursePageDTO list(@PositiveOrZero int pageNumber,  
        @Positive @Max(100) int pageSize){  
        Page<Course> page = courseRepository.findAll(  
            PageRequest.of(pageNumber, pageSize));  
        List<CourseDTO> courses = page.get()  
            .map(courseMapper::toDTO)  
            .collect(Collectors.toList());  
        return new CoursePageDTO(courses, page.getTotalElements(), page.getTotalPages());  
    }  
    ...  
}
```



# Paginação de Cursos

src/main/java/com/uneb/labweb/spring\_project/controller/CourseController.java

```
public class CourseController {  
    ...  
    @GetMapping  
    public CoursePageDTO list(  
        @RequestParam(name = "page", defaultValue = "0")  
        @PositiveOrZero int pageNumber,  
        @RequestParam(name = "pageSize", defaultValue = "10")  
        @Positive @Max(100) int pageSize){  
        return courseService.list(pageNumber, pageSize);  
    }  
    ...  
}
```



# Paginação de Cursos

src/main/java/com/uneb/labweb/spring\_project/SpringProjectApplication.java

```
...
public class SpringProjectApplication {
...
    CommandLineRunner initDatabase(CourseRepository courseRepository){
        return args -> {
            courseRepository.deleteAll();
            for (int i = 0; i < 20; i++) {
                Course c = new Course();
                c.setName("Curso Angular " + i);
                ...
            }
        };
    }
}
```



# Paginação de Cursos

src/main/resources/application.properties

```
...
spring.data.web.pageable.default-page-size=5
spring.data.web.pageable.max-page-size=10
```



# Paginação de Cursos

```
ng g interface courses/model/coursePage
```

```
vagner@Vagners-MBP angular-project % ng g interface courses/model/coursePage
CREATE src/app/courses/model/course-page.ts (32 bytes)
```

```
src/app/courses/model/course-page.ts
```

```
export interface CoursePage {
  courses : Course[];
  totalElements: number;
  totalPages: number
}
```

# Paginação de Cursos



src/app/courses/services/courses.service.ts

```
...
export class CoursesService {
  ...
  list(){
    return this.httpClient.get<CoursePage>(this.API)
      .pipe(
        first(),
        tap(courses => console.log(courses))
      );
  }
  ...
}
```

# Paginação de Cursos



src/app/courses/courses/courses.component.ts

```
...
export class CoursesComponent implements OnInit {
  courses$: Observable<CoursePage> | null = null;

  ...
  refresh(){
    this.courses$ = this.courseService.list()
      .pipe(
        catchError(error => {
          this.onError('Erro ao carregar a lista de cursos');
          return of({ courses: [], totalElements: 0, totalPages: 0 });
        })
      );
  }
  ...
}
```

# Paginação de Cursos



src/app/courses/courses/courses.component.html

```
...
<div class="column-flex mat-elevation-z8">
  <app-courses-list
    *ngIf="courses$ | async as courses; else loading"
    [courses]="courses.courses"
    (add)="onAdd()"
    (edit)="onEdit($event)"
    (remove)="onRemove($event)"
  ></app-courses-list>
...
</div>
...
```

# Paginação de Cursos



src/app/courses/courses/courses.component.ts

```
import { Component, OnInit, ViewChild } from '@angular/core';
...
import {MatPaginator, MatPaginatorModule } from '@angular/material/paginator';
...
@Component({
  selector: 'app-courses',
  standalone: true,
  imports: [MatTableModule, MatCardModule, MatToolbarModule,
    CommonModule, MatProgressSpinnerModule, MatDialogModule,
    MatButtonModule, MatIconModule, CategoryPipe, CoursesListComponent,
    MatPaginatorModule],
  templateUrl: './courses.component.html',
  styleUrls: ['./courses.component.scss'
})
...
```

# Paginação de Cursos



src/app/courses/courses/courses.component.ts

```
...
export class CoursesComponent implements OnInit {
  ...
  @ViewChild(MatPaginator) paginator!: MatPaginator;
  pageIndex = 0;
  pageSize = 10;
  ...
}
```

# Paginação de Cursos



src/app/courses/courses/courses.component.ts

```
...
export class CoursesComponent implements OnInit {
...
refresh(pageEvent: PageEvent = { length: 0, pageIndex: 0, pageSize: 10 }){
  this.courses$ = this.courseService.list(pageEvent.pageIndex, pageEvent.pageSize)
    .pipe(
      tap(() => {
        this.pageIndex = pageEvent.pageIndex;
        this.pageSize = pageEvent.pageSize;
      }),
      catchError(error => {
        this.onError('Erro ao carregar a lista de cursos');
        return of({ courses: [], totalElements: 0, totalPages: 0 });
      })
    );
}
...
}
```

# Paginação de Cursos



src/app/courses/services/courses.service.ts

```
...
export class CoursesService {
...
  list(page = 0, pageSize = 10){
    return this.httpClient.get<CoursePage>(this.API, {params: { page, pageSize }})
      .pipe(
        first(),
        tap(courses => console.log(courses))
      );
  }
...
}
```

# Paginação de Cursos



src/app/courses/courses/courses.component.html

```
...
<div class="column-flex mat-elevation-z8">
...
<mat-paginator
  [pageSizeOptions]=["5, 10, 15"]
  showFirstLastButtons
  [pageIndex]=["pageIndex"]
  [pageSize]=["pageSize"]
  [length]=["courses.totalElements"]
  (page)="refresh($event)"
  aria-label="Select page of courses">
</mat-paginator>
...
</div>
...
```

# Migração para Standalone Components



ng generate @angular/core:standalone

- vagner@192 angular-project % ng generate @angular/core:standalone
  - ? Choose the type of migration: (Use arrow keys)
  - › Convert all components, directives and pipes to standalone
    - Remove unnecessary NgModule classes
    - Bootstrap the application using standalone APIs

```
vagner@192 angular-project % ng generate @angular/core:standalone
? Choose the type of migration: Convert all components, directives and pipes to standalone
? Which path in your project should be migrated? ./
  🎉 Automated migration step has finished! 🎉
  IMPORTANT! Please verify manually that your application builds and behaves as expected.
  See https://angular.dev/reference/migrations/standalone for more information.
Nothing to be done.
```

# Migração para Standalone Components



ng generate @angular/core:standalone

```
vagner@192 angular-project % ng generate @angular/core:standalone
? Choose the type of migration:
  Convert all components, directives and pipes to standalone
❯ Remove unnecessary NgModule classes
  Bootstrap the application using standalone APIs
```

- vagner@192 angular-project % ng generate @angular/core:standalone  
? Choose the type of migration: Remove unnecessary NgModule classes  
? Which path in your project should be migrated? ./  
🎉 Automated migration step has finished! 🎉  
IMPORTANT! Please verify manually that your application builds and behaves as expected.  
See <https://angular.dev/reference/migrations/standalone> for more information.  
Nothing to be done.

# Migração para Standalone Components



ng generate @angular/core:standalone

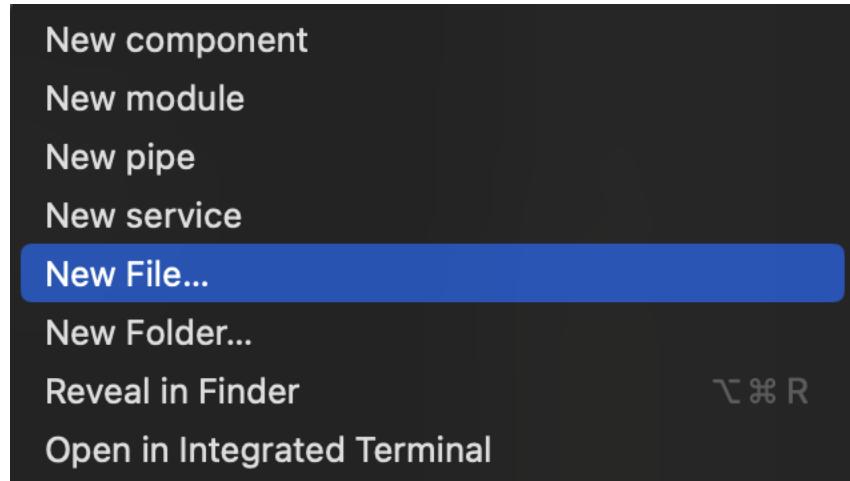
```
vagner@192 angular-project % ng generate @angular/core:standalone
? Choose the type of migration:
  Convert all components, directives and pipes to standalone
  Remove unnecessary NgModule classes
❯ Bootstrap the application using standalone APIs
```

```
vagner@192 angular-project % ng generate @angular/core:standalone
? Choose the type of migration: Bootstrap the application using standalone APIs
? Which path in your project should be migrated? ./
🎉 Automated migration step has finished! 🎉
IMPORTANT! Please verify manually that your application builds and behaves as expected.
See https://angular.dev/reference/migrations/standalone for more information.
Nothing to be done.
```

# Migração para Standalone Components



src/app/courses/courses-routes.ts



# Migração para Standalone Components



src/app/courses/courses-routes.ts

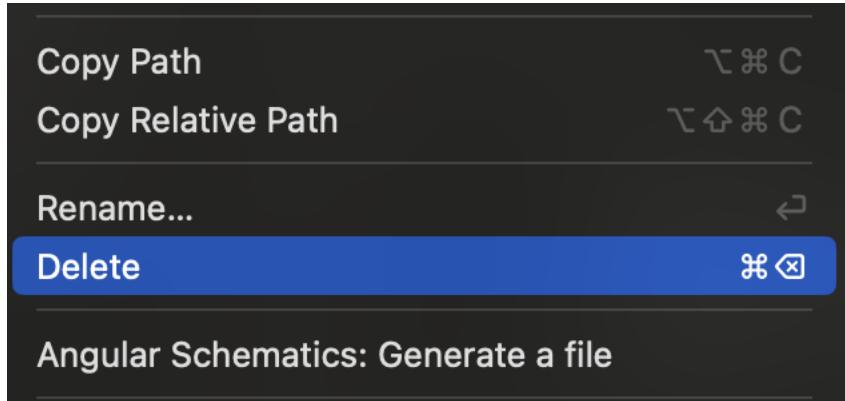
```
import { Routes } from "@angular/router";
import { CoursesComponent } from "./courses/courses.component";
import { CourseFormComponent } from "./course-form/course-form.component";
import { CourseResolver } from "./guards/course.resolver";

export const COURSE_ROUTES : Routes = [
  { path : "", component : CoursesComponent },
  { path : 'new', component : CourseFormComponent,
    resolve: { course: CourseResolver } },
  { path : 'edit/:id', component: CourseFormComponent,
    resolve: { course: CourseResolver } }
];
```

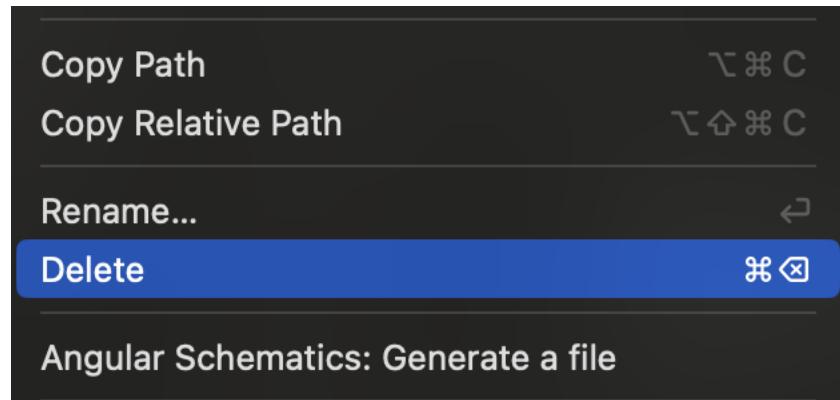
# Migração para Standalone Components



src/app/courses/courses-routing.module.ts



src/app/courses/courses.module.ts



# Migração para Standalone Components



src/app/app.routes.ts

```
import { Routes } from '@angular/router';

export const routes: Routes = [
  { path: '', pathMatch: 'full', redirectTo: 'courses' },
  {
    path: 'courses',
    loadChildren: () => import('./courses/courses.routes').then(m =>
      m.COURSE_ROUTES)
  }
];
```

# Migração para Novo Control Flow



ng generate @angular/core:control-flow

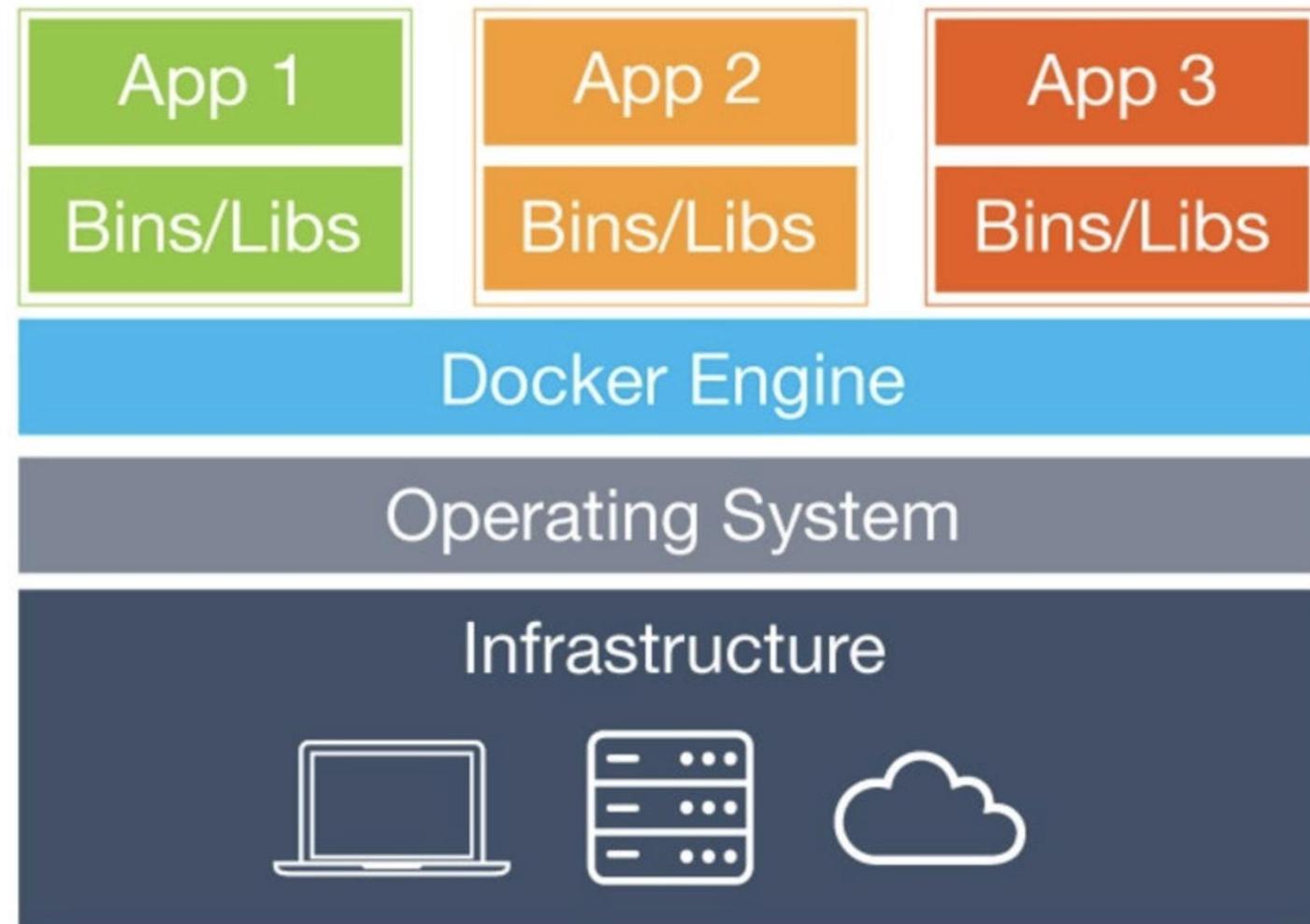
```
● vagner@192 angular-project % ng generate @angular/core:control-flow
? Which path in your project should be migrated? ./
? Should the migration reformat your templates? yes
UPDATE src/app/courses/course-form/course-form.component.html (3140 bytes)
UPDATE src/app/courses/courses/courses.component.html (874 bytes)
UPDATE src/app/courses/course-form/course-form.component.ts (3578 bytes)
```

# Migração para Novo Control Flow

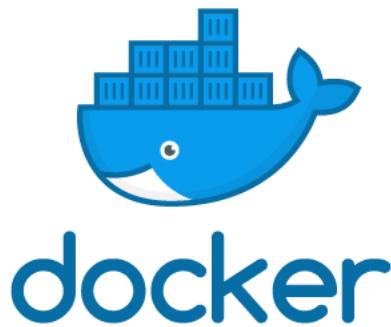


src/app/courses/courses/courses.component.html (remover)

```
<ng-template #loading>
  <div class="loading-spinner">
    <mat-spinner></mat-spinner>
  </div>
</ng-template>
```



# Instalando docker



```
sudo apt-get update
```

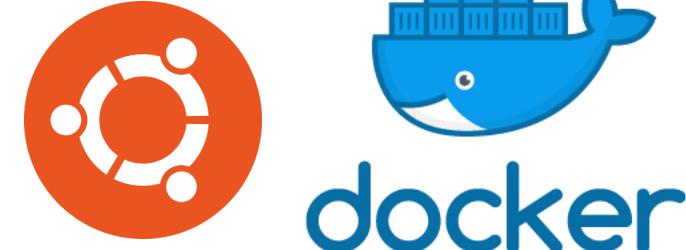
```
sudo apt-get install ca-certificates curl
```

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o  
/etc/apt/keyrings/docker.asc
```

```
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

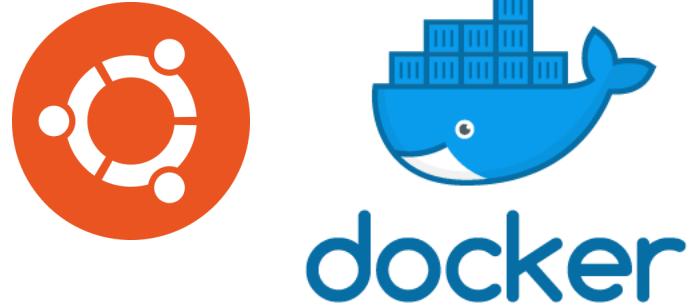
# Instalando docker



# Add the repository to Apt sources:

```
echo "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \ $(. /etc/os-release &&
echo "$VERSION_CODENAME") stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

# Instalando docker



```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

```
sudo docker run hello-world
```

# Instalando docker composer

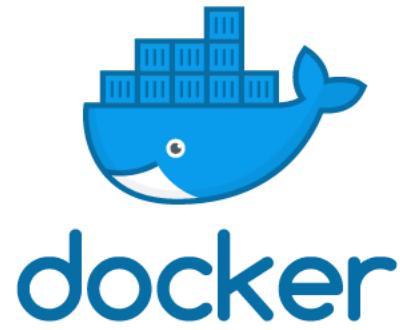


```
wget https://desktop.docker.com/linux/main/amd64/docker-desktop-amd64.deb
```

```
sudo apt-get update
```

```
sudo apt-get install ./docker-desktop-amd64.deb
```

# Instalando docker



dockerdocs

Get started   Guides   **Manuals**   Reference

Manuals

Docker Build

Docker Build Cloud

Docker Compose

Docker Desktop

Setup

Install

Mac

Understand permission require...

Windows

Understand permission require...

Enterprise deployment

Linux

Run Docker Desktop for Window...

[Home](#) / [Manuals](#) / [Docker Desktop](#) / [Setup](#) / [Install](#) / [Windows](#)

## Install Docker Desktop on Windows

**Docker Desktop terms**

Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than \$10 million USD in annual revenue) requires a [paid subscription](#).

This page contains the download URL, information about system requirements, and instructions on how to install Docker Desktop for Windows.

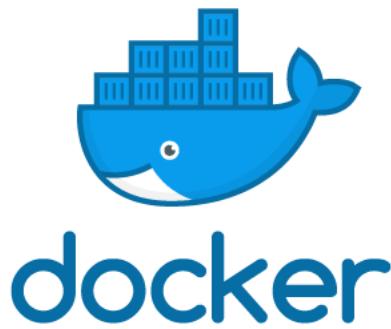
[Docker Desktop for Windows - x86\\_64](#)

[Docker Desktop for Windows - Arm \(Beta\)](#)

*For checksums, see [Release notes](#)*

## System requirements

# Instalando docker



[dockerdocs](#) Get started Guides Manuals Reference

Manuals

- Docker Build
- Docker Build Cloud
- Docker Compose
- Docker Desktop
  - Setup
  - Install
    - Mac
    - Understand permission require...
    - Windows
    - Understand permission require...
    - Enterprise deployment
    - Linux
    - Run Docker Desktop for Window...

[Home](#) / [Manuals](#) / [Docker Desktop](#) / [Setup](#) / [Install](#) / Mac

## Install Docker Desktop on Mac

**Docker Desktop terms**

Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than \$10 million USD in annual revenue) requires a [paid subscription](#).

This page contains download URLs, information about system requirements, and instructions on how to install Docker Desktop for Mac.

[Docker Desktop for Mac with Apple silicon](#)

[Docker Desktop for Mac with Intel chip](#)

*For checksums, see [Release notes](#).*

## System requirements

# Adicionando MySQL

pom.xml

The screenshot shows the Eclipse IDE interface with the Docker Compose Support extension open. A blue arrow points from the 'Add Starters...' option in the Command Palette to the search bar of the extension's interface.

**Command Palette:**

- Coverage Gutters: Preview Coverage Report
- Coverage Gutters: Display Coverage
- Coverage Gutters: Watch
- Coverage Gutters: Remove Watch
- Coverage Gutters: Remove Coverage
- Refactor Spring Boot Project...
- Upgrade Spring Boot Version...
- Add Starters...
- Command Palette...
- Red Hat Dependency Analytics Report...

**Docker Compose Support Interface:**

- Search Bar:** mysql
- Result:** ✓ MySQL Driver SQL  
MySQL JDBC driver.
- Search Bar:** docker
- Result:** Docker Compose Support Developer Tools  
Provides docker compose support for enhanced development experience.
- Message:** Select dependencies to add.
- Status:** ⚡ Selected 8 dependencies  
Press <Enter> to continue.



# Migrando para MySQL

pom.xml

```
...
<dependencies>
...
<dependency>
  <groupId>com.mysql</groupId>
  <artifactId>mysql-connector-j</artifactId>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-docker-compose</artifactId>
  <scope>runtime</scope>
</dependency>
...
</dependencies>
...
```



# Adicionando MySQL



**Project**

Gradle - Groovy    Gradle - Kotlin  
 Maven

**Language**

Java    Kotlin    Groovy

**Spring Boot**

3.4.0 (SNAPSHOT)    3.4.0 (RC1)    3.3.6 (SNAPSHOT)    3.3.5  
 3.2.12 (SNAPSHOT)    3.2.11

**Project Metadata**

Group com.example

Artifact demo

Name demo

Description Demo project for Spring Boot

Package name com.example.demo

Packaging  Jar    War

Java  23    21    17

**Dependencies**

**MySQL Driver** SQL  
MySQL JDBC driver.

**Docker Compose Support** DEVELOPER TOOLS  
Provides docker compose support for enhanced development experience.

**GENERATE** ⌘ + ↩   **EXPLORE** CTRL + SPACE   **SHARE...**



<https://start.spring.io/>

# Adicionando MySQL



demo.zip

- .gitattributes
- .gitignore
- .mvn
- HELP.md
- compose.yaml**
- mvnw
- mvnw.cmd
- pom.xml
- src

services:  
mysql:  
image: 'mysql:latest'  
environment:  
- 'MYSQL\_DATABASE=mydatabase'  
- 'MYSQL\_PASSWORD=secret'  
- 'MYSQL\_ROOT\_PASSWORD=verysecret'  
- 'MYSQL\_USER=myuser'  
ports:  
- '3306'

DOWNLOAD COPY

# Adicionando MySQL

## compose.yaml

```
services:  
  mysql:  
    image: 'mysql:latest'  
    environment:  
      - 'MYSQL_DATABASE=labweb'  
      - 'MYSQL_PASSWORD=verysecurity'  
      - 'MYSQL_ROOT_PASSWORD=verysecret'  
      - 'MYSQL_USER= labweb'  
    ports:  
      - '3306'
```



# Migrando para MySQL

src/main/resources/application.properties

```
# spring.application.name=spring-project
# spring.datasource.url=jdbc:h2:mem:testdb
# spring.datasource.driverClassName=org.h2.Driver
# spring.datasource.username=sa
# spring.datasource.password=12345678
# spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
# spring.jpa.show-sql=true

spring.datasource.url=jdbc:mysql://localhost:3606/labweb
spring.datasource.username=labweb
spring.datasource.password=verysecurity
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=create

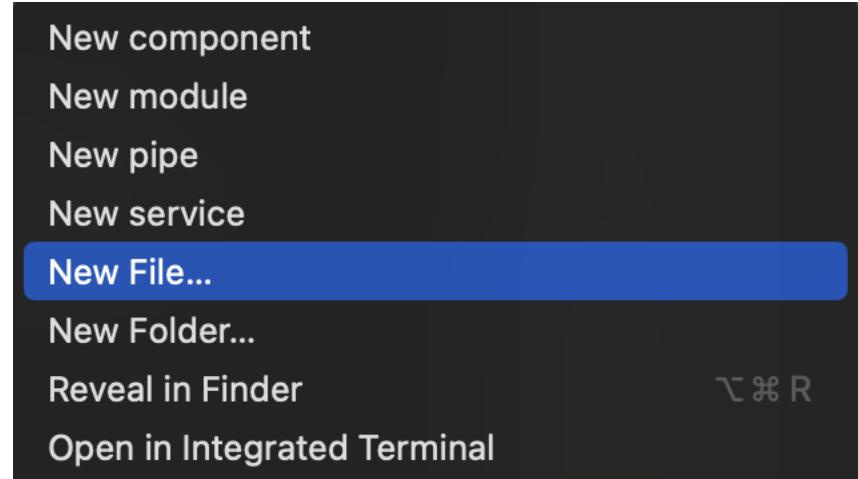
...
```



# Adicionando o Spring em um container Docker



## Dockerfile



# Adicionando o Spring em um container Docker

## Dockerfile

```
FROM openjdk:22-jdk-slim
ARG JAR_FILE=target/spring-project-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} spring-project.jar
EXPOSE 8080
ENTRYPOINT ["java", "-jar", "/spring-project.jar"]
```



# Adicionando o Spring em um container Docker

## compose.yaml



```
services:  
  mysql:  
    image: 'mysql:8.0'  
    environment:  
      - 'MYSQL_DATABASE=labweb'  
      - 'MYSQL_PASSWORD=verysecurity'  
      - 'MYSQL_ROOT_PASSWORD=verysecret'  
      - 'MYSQL_USER=labweb'  
    ports:  
      - '3306'  
    healthcheck:  
      test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]  
      interval: 10s  
      timeout: 5s  
      retries: 3  
...  
...
```

# Adicionando o Spring em um container Docker

## compose.yaml



```
services:  
...  
spring-app:  
  build:  
    context: .  
    dockerfile: Dockerfile  
  container_name: spring-courses  
  ports:  
    - '8080:8080'  
  environment:  
    SPRING_DATASOURCE_URL: jdbc:mysql://mysql:3306/labweb  
    SPRING_DATASOURCE_USERNAME: labweb  
    SPRING_DATASOURCE_PASSWORD: verysecurity  
  depends_on:  
    mysql:  
      condition: service_healthy
```

# Migrando para MySQL

src/main/resources/application.properties

```
# spring.application.name=spring-project
# spring.datasource.url=jdbc:h2:mem:testdb
# spring.datasource.driverClassName=org.h2.Driver
# spring.datasource.username=sa
# spring.datasource.password=12345678
# spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
# spring.jpa.show-sql=true

spring.datasource.url=${SPRING_DATASOURCE_URL}
spring.datasource.username=${SPRING_DATASOURCE_USERNAME}
spring.datasource.password=${SPRING_DATASOURCE_PASSWORD}
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=create

...
```

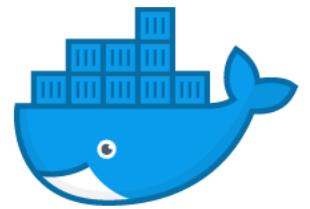


# Adicionando o Spring em um container Docker



```
mvn clean package -DskipTests  
docker-compose up --build
```

# Adicionando o Angular em um container Docker



docker

## Dockerfile

- New component
- New module
- New pipe
- New service
- New File...**
- New Folder...
- Reveal in Finder ⌘ R
- Open in Integrated Terminal

## compose.yaml

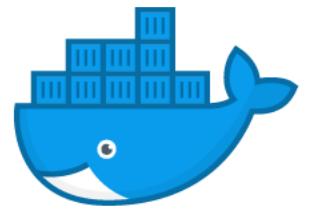
- New component
- New module
- New pipe
- New service
- New File...**
- New Folder...
- Reveal in Finder ⌘ R
- Open in Integrated Terminal

# Adicionando o Angular em um container Docker



## Dockerfile

```
FROM node:20
WORKDIR /angular
COPY ..
RUN npm install
RUN npm install -g @angular/cli
EXPOSE 4200
CMD ["npm", "run", "start", "--", "--host", "0.0.0.0"]
```

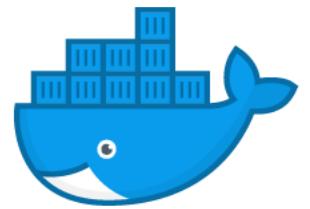


docker

# Adicionando o Angular em um container Docker

## compose.yaml

```
services:  
  angular-app:  
    build:  
      context: .  
      dockerfile: Dockerfile  
    container_name: angular-courses  
    ports:  
      - "4200:4200"  
    command: ["npm", "run", "start", "--", "--host", "0.0.0.0"]
```

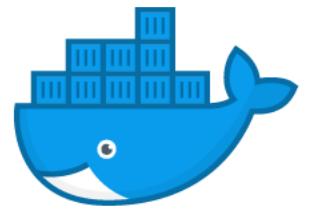


docker

# Adicionando o Angular em um container Docker

## proxy.conf.js

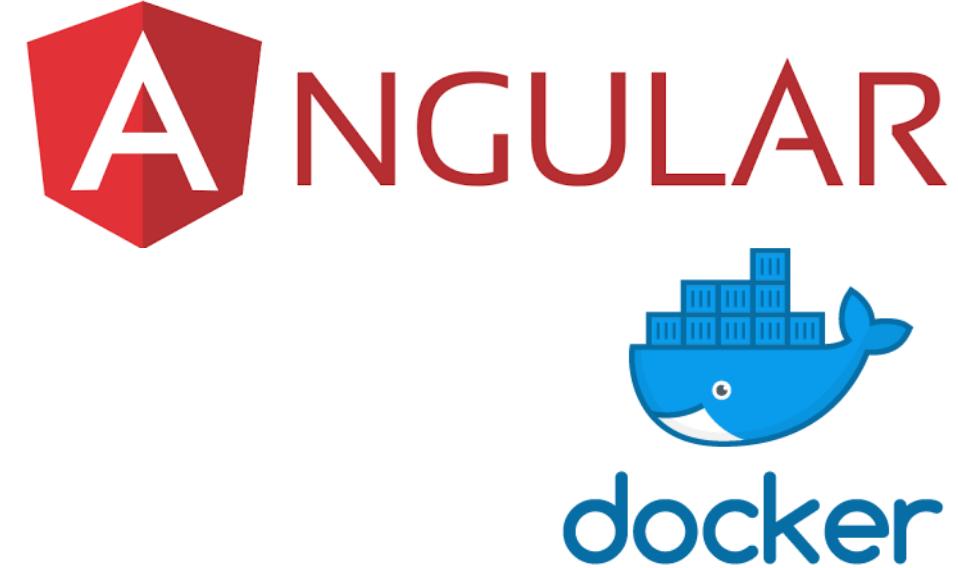
```
const PROXY_CONFIG = [  
  {  
    context: ['/api'],  
    target: 'http://spring-courses:8080/',  
    secure: false,  
    logLevel: 'debug'  
  }  
];  
  
module.exports = PROXY_CONFIG;
```



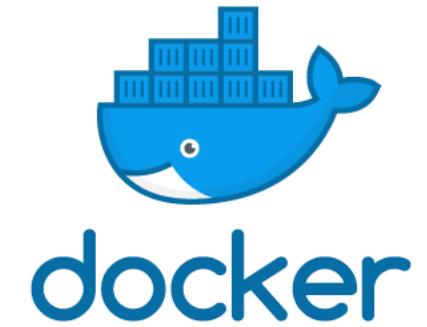
docker

# Adicionando o Angular em um container Docker

`docker-compose up --build`



# Adicionando o Spring e o Angular em uma mesma rede no Docker



```
docker network create courses-projects
```

```
docker network connect courses-projects angular-courses
```

```
docker network connect courses-projects spring-courses
```