



# WEB ACADEMY

## Programação Avançada Front-end

Daniel Augusto Nunes da Silva

# **Apresentação**



# Ementa

- HTML5: novos elementos e atributos. APIs na especificação do HTML5 (canvas, audio e video, geolocation e storage). Précompiladores: Less e Sass. Introdução ao Bootstrap. Design de Componentes Bootstrap com CSS. Componentes Javascript do Bootstrap. Componentes Bootstrap do AngularJS. Especificação do Google Material Design e sua aplicação. Introdução ao SVG para sintetização de imagens vetoriais. Resolução de problemas de Acessibilidade.

# Objetivos

- **Geral:** Desenvolver **competências avançadas em programação web full-stack**, baseado nos recursos dos frameworks **Spring Boot** e **Angular**, bem como em boas práticas de programação, padrões de projeto e usabilidade.
- **Específicos:**
  - Aprimorar conhecimentos de UI/UX, acessibilidade e otimização da experiência do usuário.
  - Habilitar o aluno a desenvolver sistemas seguros com recursos de autenticação e autorização.
  - Discutir e aplicar práticas avançadas no desenvolvimento web.

# Conteúdo programático

## Introdução

- Revisão sobre arquitetura e funcionamento de aplicações web.

## UI/UX

- Resolução de problemas de Acessibilidade;
- Pré-processadores CSS;
- Melhoria da experiência do usuário em aplicações web.

## Autenticação e Segurança

- Autenticação de usuários e autorização de acesso com Spring Security;
- Autenticação de usuários no Angular;
- Controle de acesso por rotas no Angular.

## Práticas Avançadas

- Padrão de projeto DTO;
- Paginação e ordenação de resultados.

# Bibliografia



## JavaScript e JQuery: desenvolvimento de interfaces web interativas.

Jon Duckett  
1ª Edição – 2016  
Editora Alta Books  
ISBN 9781118871652



## The TypeScript Handbook

Microsoft  
<https://www.typescriptlang.org/docs/handbook/intro.html>



## Engenharia de Software Moderna

Marco Tulio Valente  
<https://engsoftmoderna.info/>



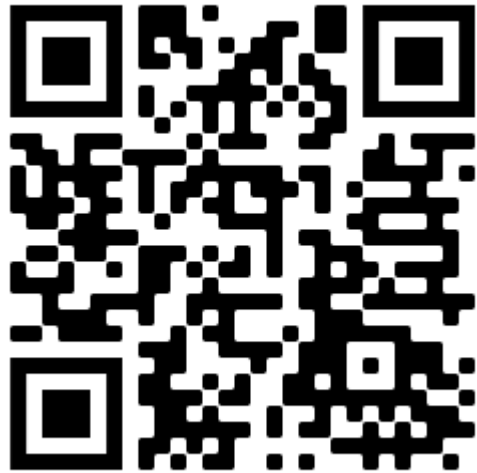
## Spring in Action

Craig Walls  
6ª Edição – 2021  
Editora Manning  
ISBN 9781617297571

# Sites de referência

- Sass Documentation: <https://sass-lang.com/documentation/>
- Angular Docs: <https://angular.io/docs>
- TypeScript Documentation: <https://www.typescriptlang.org/docs/>
- Spring Boot Reference Documentation: <https://docs.spring.io/spring-boot/docs/current/reference/html/index.html>
- MDN Web Docs: Aprendendo desenvolvimento web: <https://developer.mozilla.org/pt-BR/docs/Learn>
- Swagger Documentation: <https://swagger.io/docs/>

# Contato

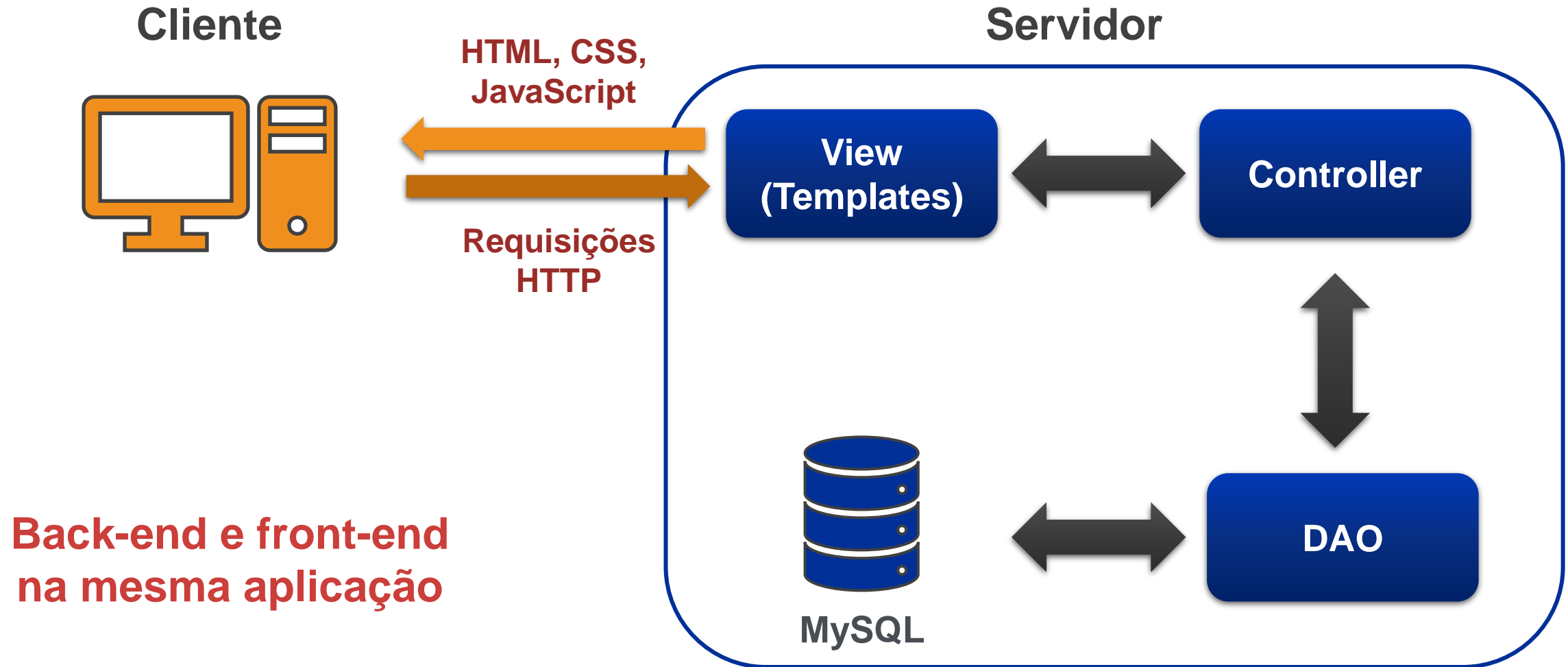


<https://linkme.bio/danielnsilva/>

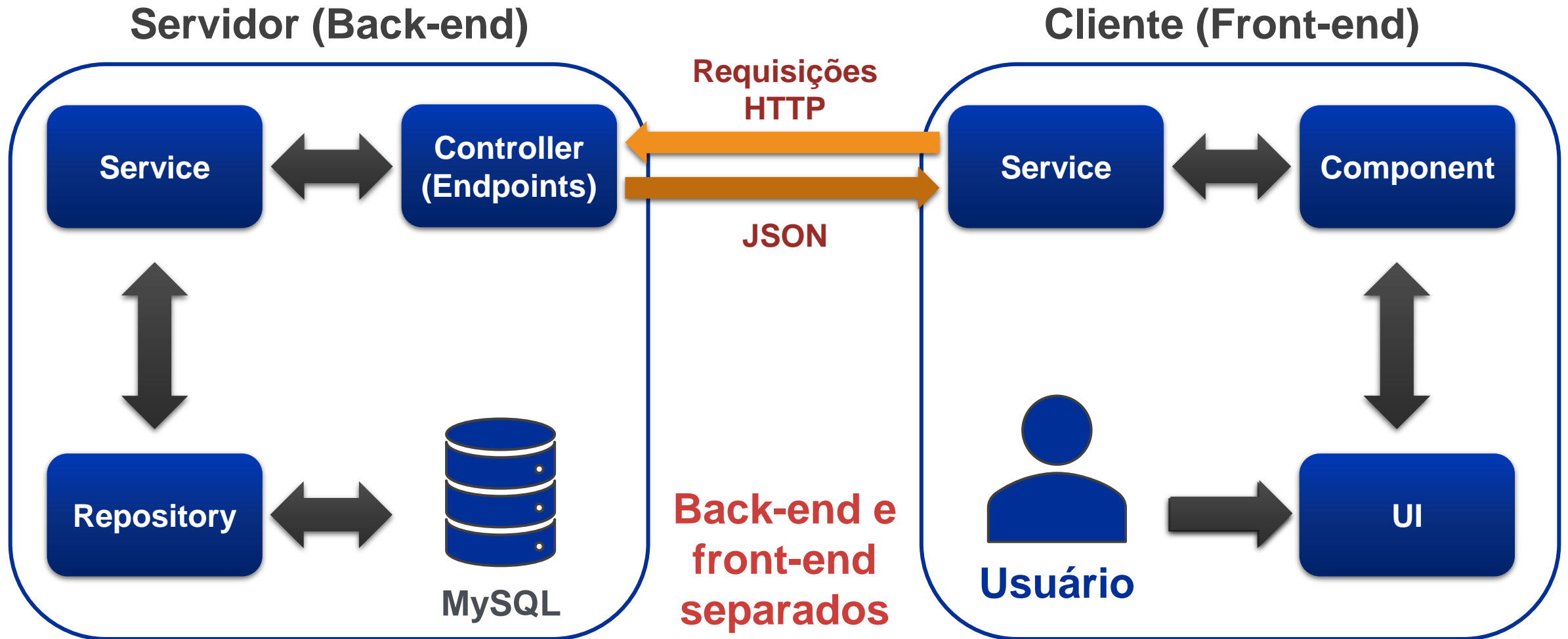


# Introdução

# Arquitetura de uma aplicação web



# Arquitetura de uma aplicação front-end



**UI/UX**

# Acessibilidade na web

- **Acessibilidade** na web significa permitir que o maior número possível de pessoas possa usar a web, independente da sua limitação, **removendo barreiras que possam impedir esse acesso**. Exemplos de barreiras:
  - Imagens que não possuem texto alternativo.
  - Formulários que não podem ser navegados em uma sequência lógica ou que não estão rotulados.
  - Páginas com tamanhos de fontes absoluta, que não podem ser aumentadas ou reduzidas facilmente.
- Os **padrões web** representam o básico para uma página web acessível.
- É também importante acrescentar aos padrões web as técnicas de acessibilidade associadas ao **WCAG** e suas recomendações.

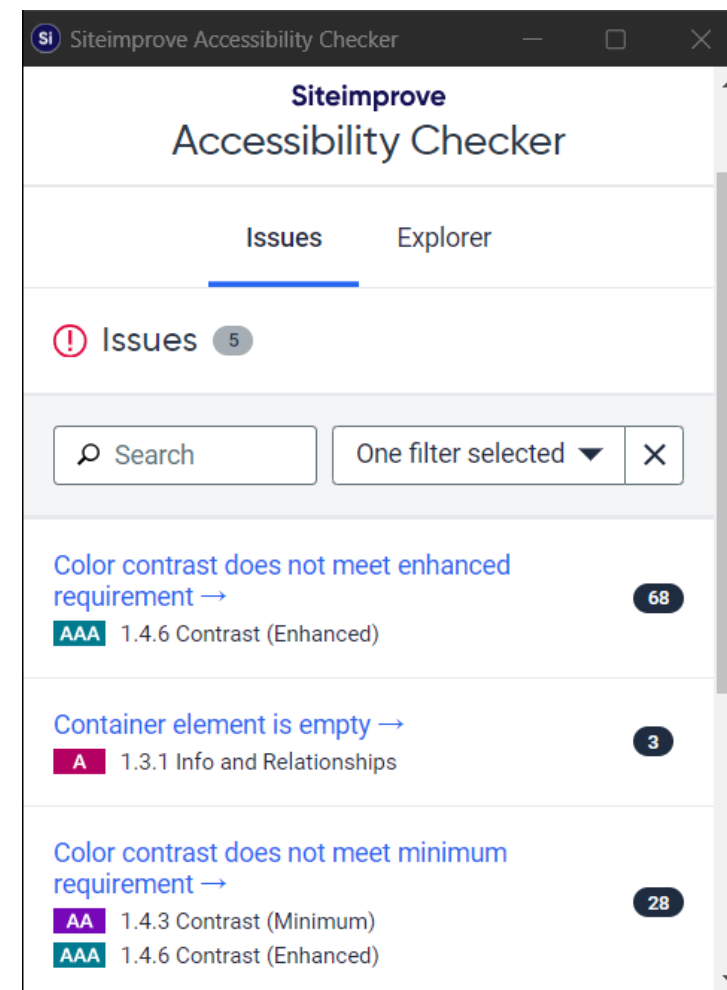


# WCAG

- **WCAG** (**W**eb **C**ontent **A**ccessibility **G**uidelines) são padrões internacionais definidos pelo W3C para tornar o conteúdo web mais acessível, sobretudo para pessoas com algum tipo de limitação.
- Possui 3 níveis para representar o grau de acessibilidade:
  - Nível **A** (básico): Requisitos mínimos para ser acessível.
  - Nível **AA** (intermediário): Padrão recomendado para a maioria dos casos.
  - Nível **AAA** (avançado): O padrão mais elevado, mas nem sempre prático para todos os conteúdos.
- Um exemplo de problema de acessibilidade com requisitos diferentes nos 3 níveis é o **controle de contraste de texto em relação ao seu fundo**.

# Resolução de problemas de Acessibilidade

- Identificar e resolver problemas de acessibilidade é um processo que envolve uma combinação de recursos:
  - Ferramentas de avaliação automática;
  - Testes manuais;
  - Envolvimento do público alvo (usuários com diversos tipos de limitações).
- Siteimprove Accessibility Checker (Extensão Chrome):
  - <https://chromewebstore.google.com/detail/djcglbmbegflehmbleechkjhmedcopn>



# Pré-processadores CSS

- **Estendem a funcionalidade padrão do CSS** com recursos adicionais como variáveis, *mixins* (blocos de código reutilizáveis), funções, herança e aninhamento de regras.
  - Pré-processadores resolvem certas limitações do CSS, projetada para ser simples e declarativa. No entanto, **atualizações mais recentes já incorporam recursos inspirados nos pré-processadores, como o suporte a variáveis.**
- Exemplos: **Sass/Scss**, Less e Stylus.
- **O código precisa ser compilado** para gerar o CSS (suportado pelo navegador).



# Sass/Scss

## SCSS

```
$color: blue;

.button {
  background-color: $color;
  &:hover {
    background-color: darken($color, 10%);
  }
}
```



## CSS

```
.button {
  background-color: blue;
}

.button:hover {
  background-color: #0000cc;
}
```

<https://sass-lang.com/playground>

# Migrando para SCSS no Angular

1. Renomear arquivos CSS no diretório `src/app`.
  2. Alterar o parâmetro `styleUrls` nos componentes.
  3. Atualiza a configuração de estilo padrão para novos componentes no arquivo `angular.json`
  4. Renomear todas as referências ao arquivo `styles.css` para `styles.scss` no arquivo `angular.json`.
- SCSS-Migrate:
    - `ng add schematics-scss-migrate`





# Melhoria da experiência do usuário em webapps

- UX é sobre encontrar as melhores soluções para atender determinado problema e garantir uma boa experiência para os usuários.
- **O que é um problema de UX?** São necessidades implícitas dos usuários, nem sempre identificáveis por eles próprios, exigindo análise mais aprofundada.
- **Técnicas para identificar problemas:** análise de dados (ex: métricas de desempenho), coleta de feedback dos usuários (ex: formulários, entrevistas, etc.), testes de usabilidade, dentre outros.
- Além da identificação, existem técnicas que auxiliam a definir e escrever problemas em UX, como a **Técnica dos 5 Ws**: What, Who, Where, When, Why.

# Técnica dos 5 Ws

- Funciona como um guia para a coleta informações completas sobre necessidades do usuário perguntando: o quê, quem, onde, quando e por quê, para definir claramente o problema.
  - Qual é o problema? (What)
  - Quem é impactado pelo problema? (Who)
  - Onde acontece o problema? (Where)
  - Quando o problema acontece? (When)
  - Por que o problema acontece? (Why)

# Exemplo de aplicação da Técnica dos 5 Ws

- **What (O quê):** Usuários estão achando difícil realizar o checkout no aplicativo de e-commerce.
- **Who (Quem):** O problema afeta principalmente novos usuários que estão tentando fazer a primeira compra.
- **Where (Onde):** A dificuldade ocorre na página de pagamento, especificamente no preenchimento de informações de cartão de crédito.
- **When (Quando):** O problema surge principalmente durante promoções de alta demanda, quando os usuários estão com pressa para concluir suas compras.
- **Why (Por quê):** Talvez o formulário de pagamento seja muito longo, as instruções sejam confusas ou a UI não seja intuitiva, levando a uma experiência de usuário ruim.

# Técnica dos 5 Ws aplicada ao SGCM

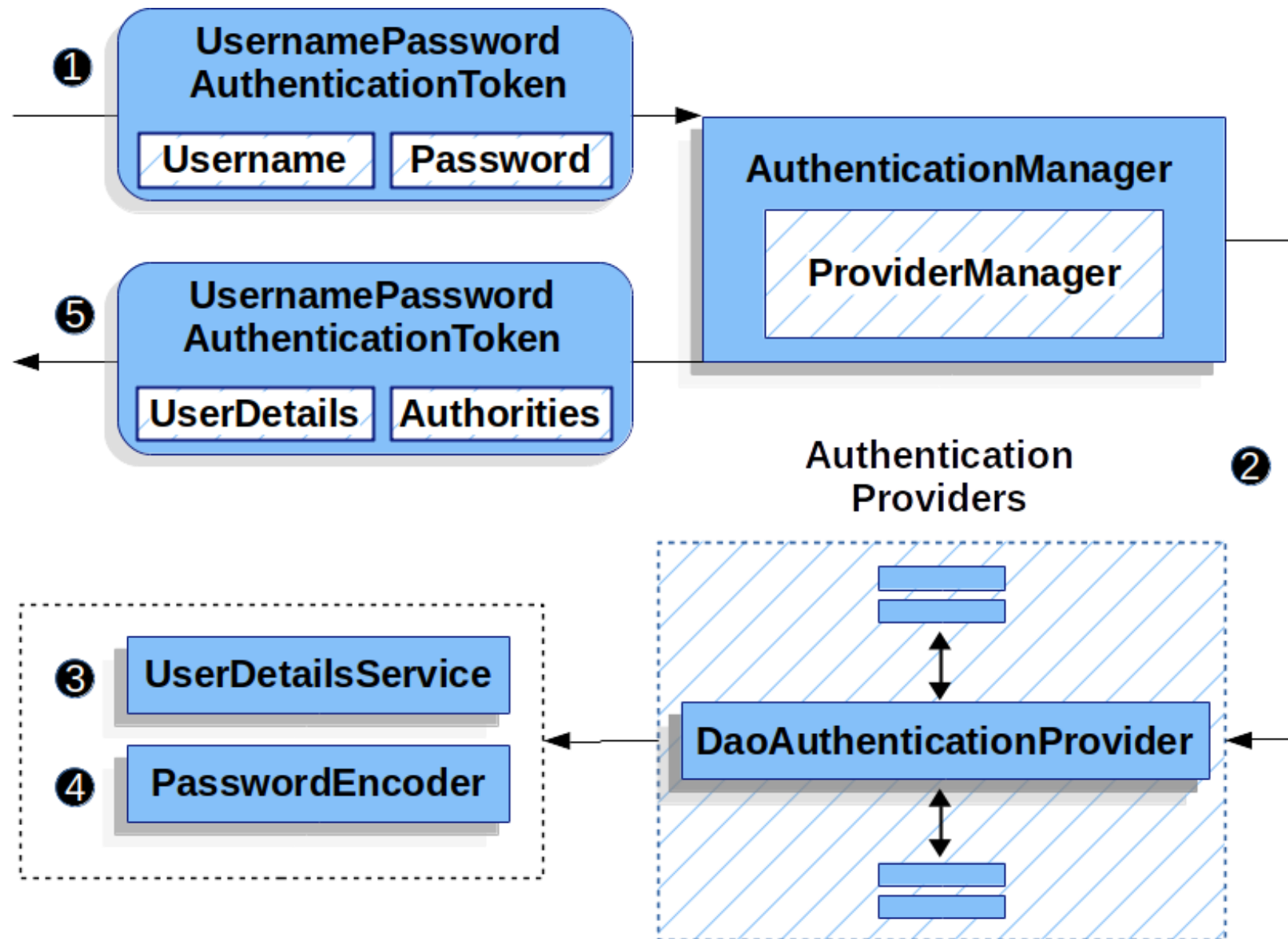
- **What (O quê):** Usuários não recebem qualquer indicação de que um processo está em andamento quando um recurso está sendo carregado, levando à impressão de que o sistema está inoperante.
- **Who (Quem):** Impacta usuários não familiarizados com o tempo de resposta esperado do sistema.
- **Where (Onde):** O problema é observado na interface do usuário onde ações de carregamento ou atualização de dados ocorrem.
- **When (Quando):** Acontece cada vez que um usuário realiza uma ação que requer carregamento ou processamento de dados, sem uma indicação visual imediata.
- **Why (Por quê):** Falta de uma interface de usuário adequada que forneça feedback durante operações de carregamento, como spinners de carregamento, barras de progresso ou mensagens informativas.

# Autenticação e Segurança



# Autenticação de usuários

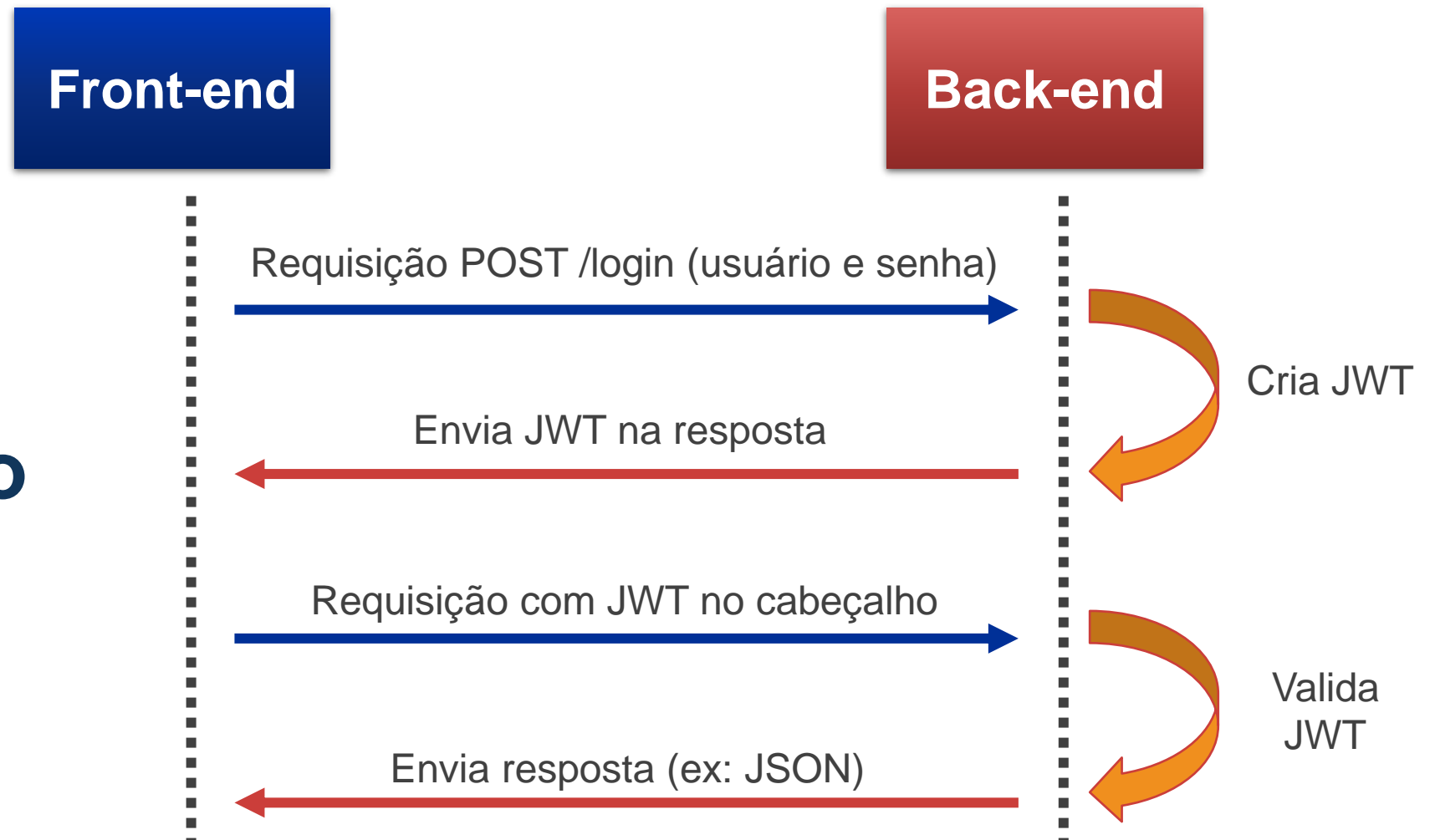
**Spring Security:** fornece recursos para **autenticação**, **autorização de acesso** e proteção contra ataques comuns.



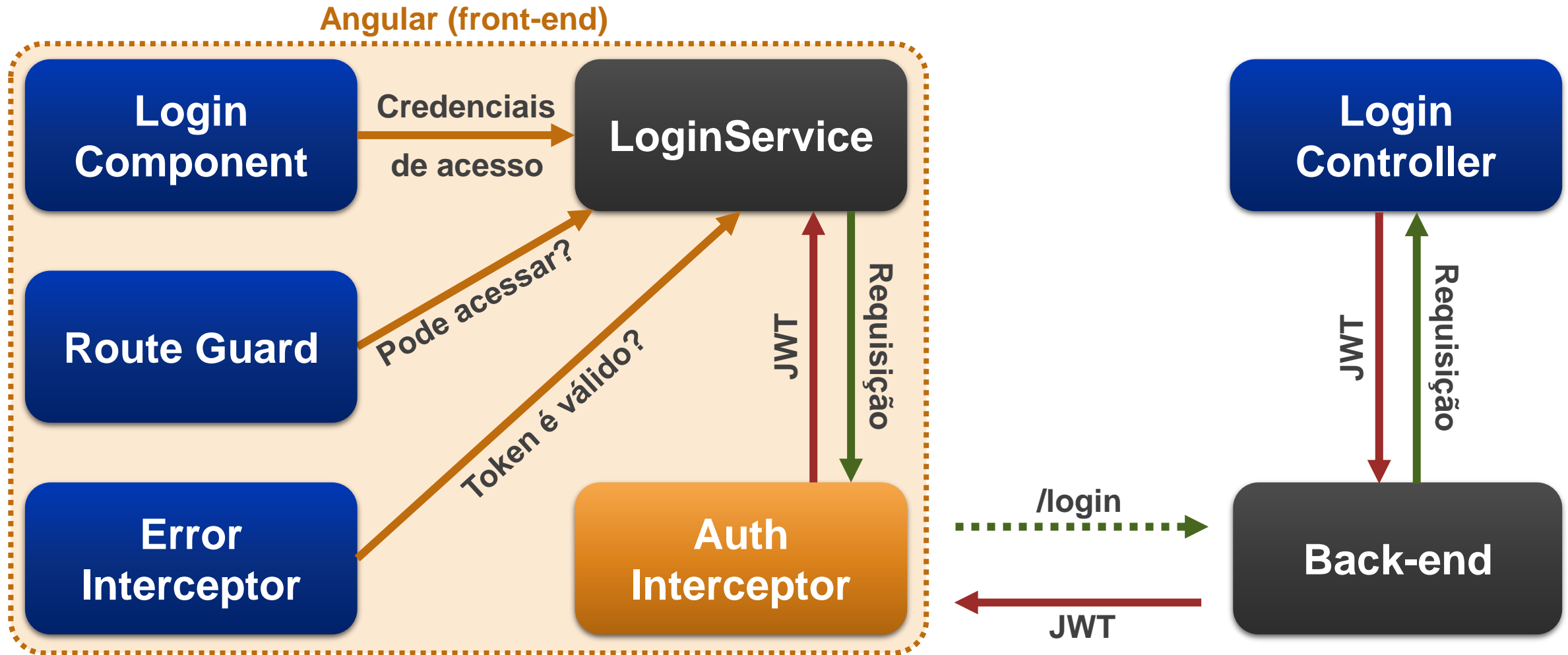
Fonte: <https://docs.spring.io/spring-security/site/docs/5.5.x-SNAPSHOT/reference/html5/>

# Autenticação JWT

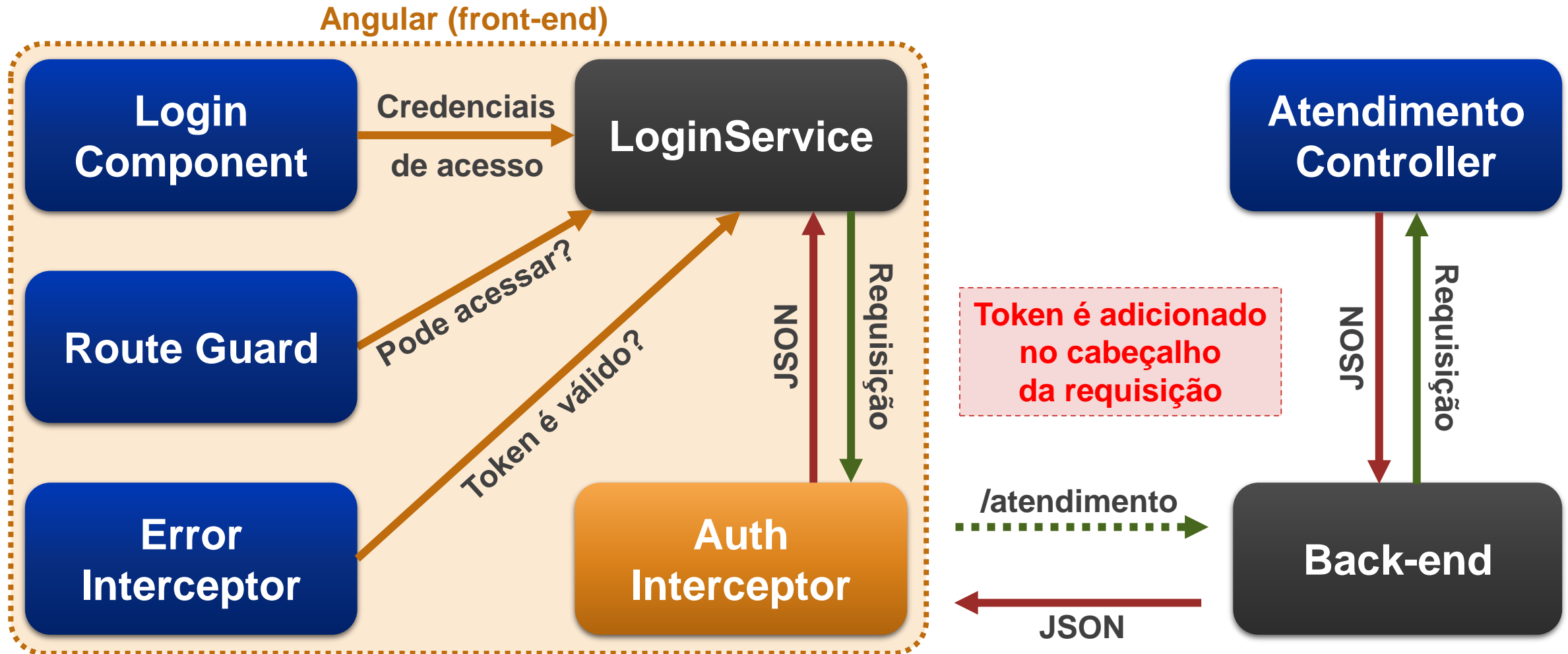
JSON Web Token



# Autenticação e autorização no front-end



# Autenticação e autorização no front-end



# Controle de acesso por rotas

- **Route Guards** são um conjunto de funções que permitem controlar o acesso às rotas, impedindo acessos não autorizados.

```
{  
  path: 'agenda',  
  component: AgendaComponent,  
  canActivate: [funcaoGuard]  
}
```

Tipo	Descrição
<b>canActivate</b>	Controla se uma rota pode ser acessada.
<b>canActivateChild</b>	Controla se uma rota “filha” pode ser acessada.
<b>canDeactivate</b>	Controla se o usuário pode sair de uma rota.
<b>canMatch</b>	Controla se uma rota pode ser carregada. Não impede que outras “versões” da rota sejam acessadas.

- Exemplos de uso: <https://angular.io/guide/router#preventing-unauthorized-access>



# Práticas Avançadas

# Padrão de projeto DTO

- O padrão **DTO** (*Data Transfer Object*) representa um **formato utilizado na transferência de dados** entre sistemas diferentes (ou entre camadas de um mesmo sistema).
- Não é o mesmo que classes de camada de modelo: **DTO não representa um objeto que será persistido.**

```
public class UsuarioDto {  
    // Mesmos atributos da camada  
    // de modelo, mas sem a senha.  
    private Long id;  
    private String nomeCompleto;  
    private String nomeUsuario;  
    private String papel;  
    private boolean ativo;  
    // Getters e Setters  
}
```

# DTO com Java Records

```
public class UsuarioDto {  
    private Long id;  
    private String nomeCompleto;  
    private String nomeUsuario;  
    private String papel;  
    private boolean ativo;  
  
    // Getters e Setters  
  
}
```

```
public record UsuarioDto(  
    Long id,  
    String nomeCompleto,  
    String nomeUsuario,  
    String papel,  
    boolean ativo  
) {}
```

```
// Acessar um atributo  
dto.nomeCompleto();
```

**Java Records são imutáveis**, pois uma vez criado, o estado de um objeto não podem ser alterado. Não há como definir valor para os atributos após criação do objeto (sem *setters*).

# Paginação e ordenação de resultados

- **Paginação de resultados** é uma técnica comum em aplicações que acessam bancos de dados, com objetivo de melhorar a **performance**, **usabilidade**, além de **reduzir a carga sobre o servidor da aplicação**.
- O **Spring Data** fornece a interface **Pageable**, que facilita a implementação de **paginação e ordenação** em consultas a banco de dados.

```
public ResponseEntity<Page<Atendimento>> get(Pageable page) {  
    Page<Atendimento> registros = servico.get(page);  
    return new ResponseEntity<>(registros, HttpStatus.OK);  
}
```

- Exemplo de requisição: `GET /atendimento/?page=0&size=10&sort=data,desc&sort=hora,asc`

**Fim!**



# Referências

- DUCKETT, Jon. **Javascript e JQuery: desenvolvimento de interfaces web interativas**. 1. ed. [S. I.]: Alta Books, 2016. 640 p.
- GOOGLE (ed.). **Angular Docs**. [S. I.], 2023. Disponível em: <https://angular.io/docs>.
- MARCO TULIO VALENTE. **Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade**, 2020. Disponível em: <https://engsoftmoderna.info/>
- MICROSOFT (ed.). **The TypeScript Handbook**. [S. I.], 2023. Disponível em: <https://www.typescriptlang.org/docs/handbook/intro.html>.

# Referências

- MOZILLA (ed.). **MDN Web Docs: Aprendendo desenvolvimento web**. [S. l.], 2023. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn>.
- SPRING (ed.). **Spring Boot Reference Documentation**. [S. l.], 2023. Disponível em: <https://docs.spring.io/spring-boot/docs/current/reference/html/index.html>.
- WALLS, Craig. **Spring in Action**. 6. ed. Shelter Island: Manning, 2021. 520 p.
- W3C. Cartilha de Acessibilidade na Web. [S.l.], 2015. Disponível em: <https://www.w3c.br/GT/GrupoAcessibilidade>.