Spring Security e JWT

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Considerações iniciais
- Entendendo Json Web Token (JWT)
- Estrutura de classes
- Laboratório

Entendendo tokens JWT

Entendendo tokens JWT (Json Web Token)

- O JWT é um padrão de mercado (RFC-7519) que define como transmitir e armazenar objetos JSON de forma compacta e segura entre diferentes aplicações usando a arquitetura REST ou Microserviços sendo RESTful ou JSON
- Os dados nele contidos podem ser validados a qualquer momento, pois o token é assinado digitalmente e sua validação é feita a cada requisição

JWT: Definição

- O token JWT é definido por três seções:
  - **Header**
    - Define informações sobre o tipo do token, neste caso JWT
  - **Payload**
    - Contém informações da entidade autorizada no caso do usuário que fez login
  - **Signature**
    - A assinatura é a junção de todas as partes somadas a uma chave de assinatura ou certificado, e tudo é codificado em Base64
  - **Vantagens**
    - Permite a comunicação segura entre diferentes sistemas e integrações, também evita que os dados sejam capturados e manipulados de forma errada por alguém com más intenções

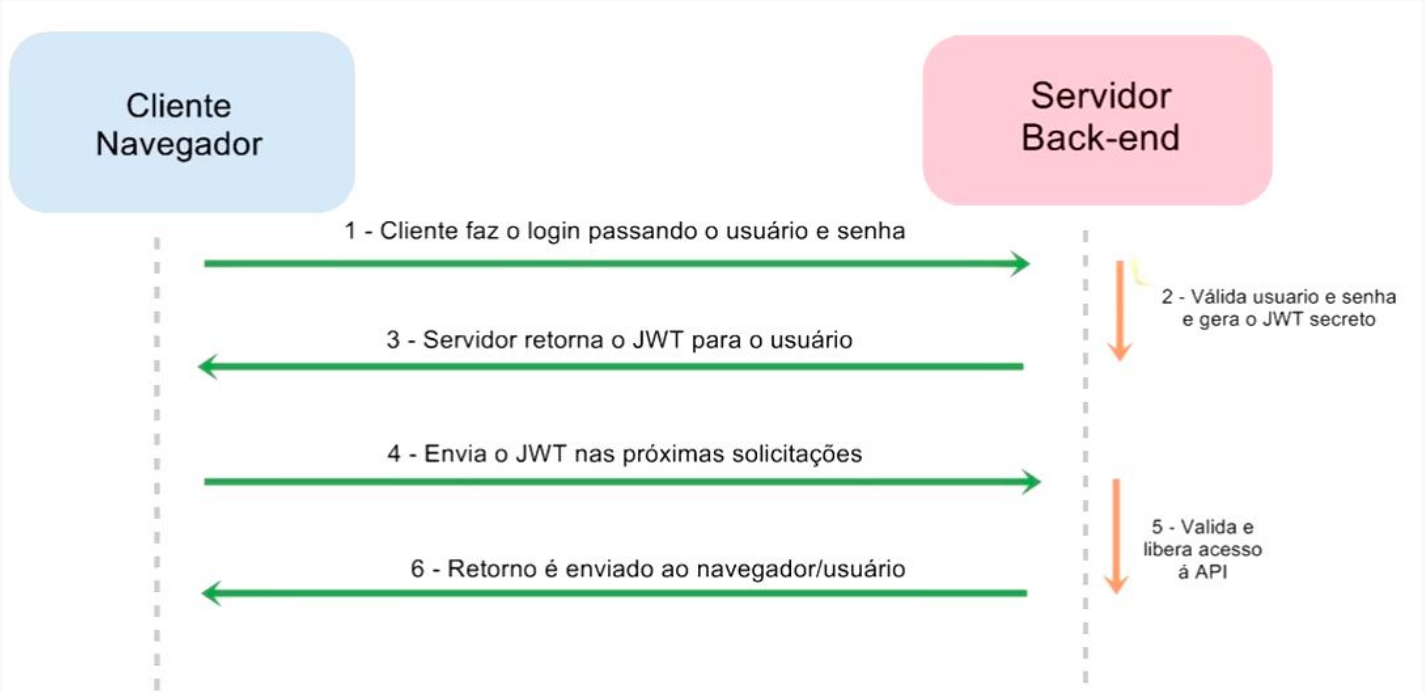
JWT: Aparência e padrão de uso

- O padrão utilizado acompanha a nomenclatura de **Authorization** e **Bearer** que é o padrão de mercado e do Http para JWT
- Esse é o JWT gerado para um login e autenticação usando o usuário brunomoura com a senha 123, podemos ver as 3 partes Header, Payload e Signature divididas por ponto e todas essa junção é codificada usando o padrão de chaves HS512 mas existem outros padrões também

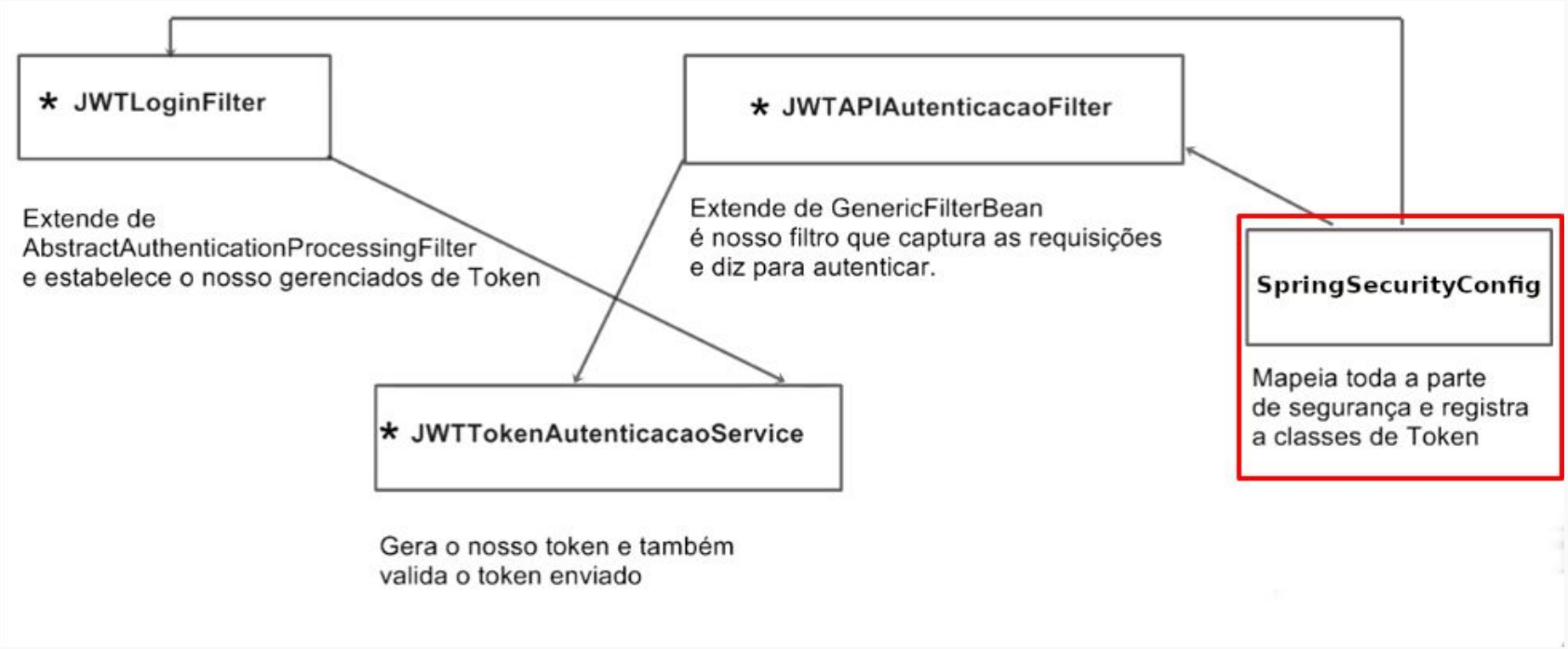
JWT: Aparência e padrão de uso

- Praticamente todos os sistemas mais novos possuem esse modo de autenticação, por exemplo, os bancos, caso você use o aplicativo do seu banco pode ter certeza que ele usa JWT em suas autenticações
- O Token ou JWT é a sua chave e ninguém pode saber dele, além de você mesmo e o servidor
- Muitas integrações de API fornecem o JWT o TOKEN para sua autenticação como forma de provar que você é quem realmente que diz ser

JWT: Funcionamento e estrutura



JWT: Estrutura de Classes



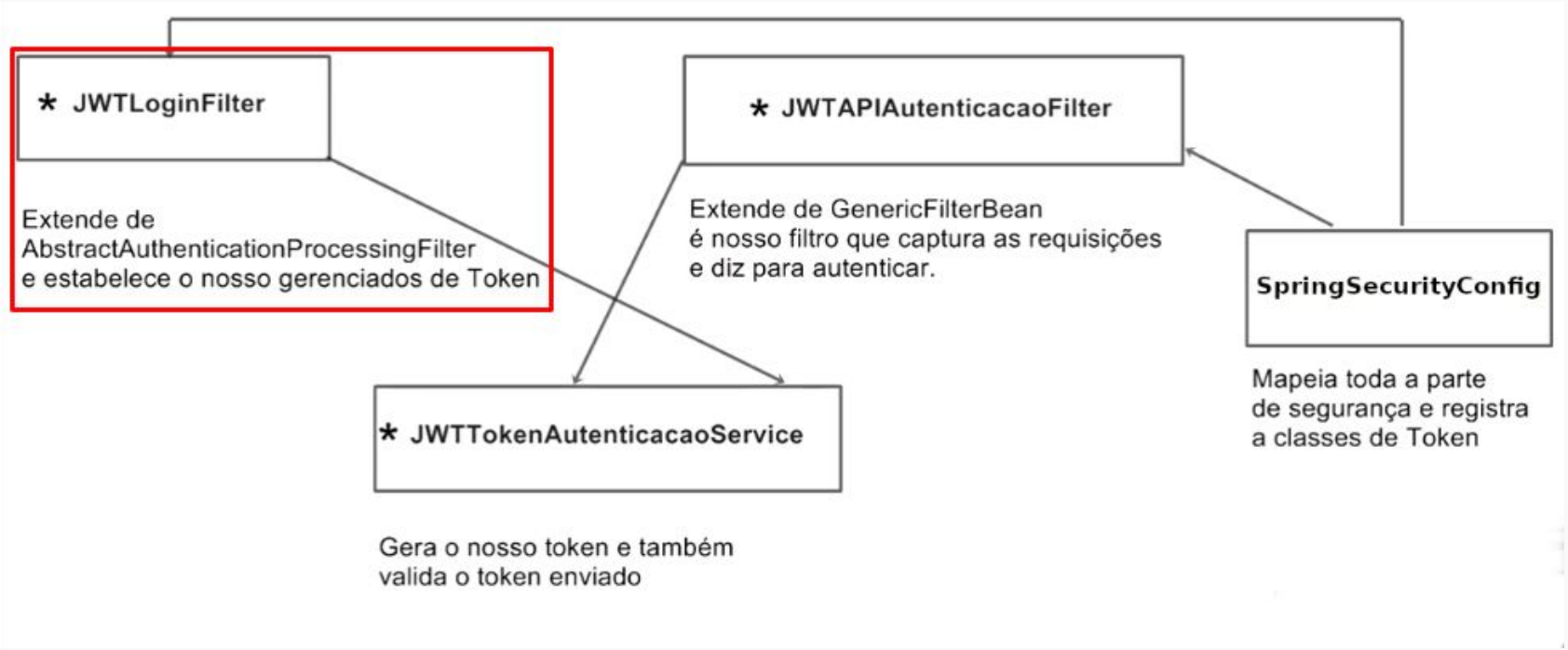
JWT: Classes e suas responsabilidades

- **SpringSecurityConfig**
  - Classe central da configuração do Spring Security, quando rodamos nosso projeto esta classe será executada pelo Spring carregando todas as configurações que definimos

Etapas de Implementação: SpringSecurityConfig

- Após fazer todos ajustes na parte do usuário (aula anterior)
- É a classe central de configuração onde serão mapeados: endereços, URLs, autorizações e bloqueios de acesso a URLs
- Sobrescrever os métodos da classe **WebSecurityConfigurerAdapter**:
  - **configure(HttpSecurity http)**: onde iremos aplicar o mapeamento dos filtros de autorizações e bloqueios necessários
  - **configure(AuthenticationManagerBuilder auth)**: onde será informado o provedor de autenticação
  - **configure(WebSecurity web)**: podemos usar para configurar acessos que não passaram por filtros

JWT: Estrutura de Classes



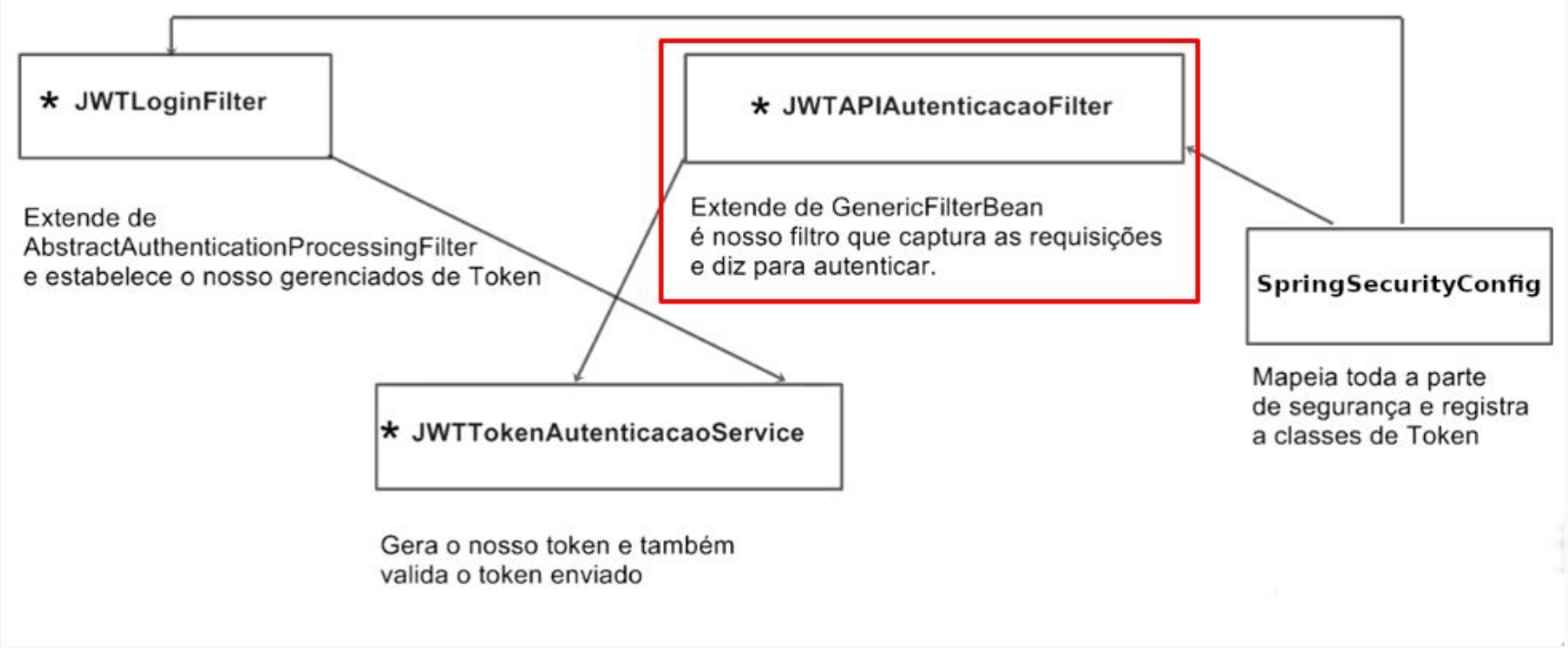
JWT: Classes e suas responsabilidades

- **JWTLoginFilter**
  - Classe que irá estender da AbastractAuthenticationProcessingFilter, e estabelece o gerenciador de token, o Spring tem o gerenciador de token dele, mas precisamos estender desta parte abstrata para se conectarmos com todo o núcleo do Spring

Etapas de Implementação: JwtLoginFilter

- Classe responsável por fazer o gerenciamento do token
  - criar o construtor **JwtLoginFilter(String url, AuthenticationManager authenticationManager)** que tem a função de forçar a autenticação de URL
  - sobrescrever os métodos:
    - **attemptAuthentication(HttpServletRequest request, HttpServletResponse response)**: onde pegamos o token do usuário para validar e retornamos os acessos
    - **successfulAuthentication(HttpServletRequest request,...)**: caso sucesso na autenticação retornamos o token do usuário

JWT: Estrutura de Classes



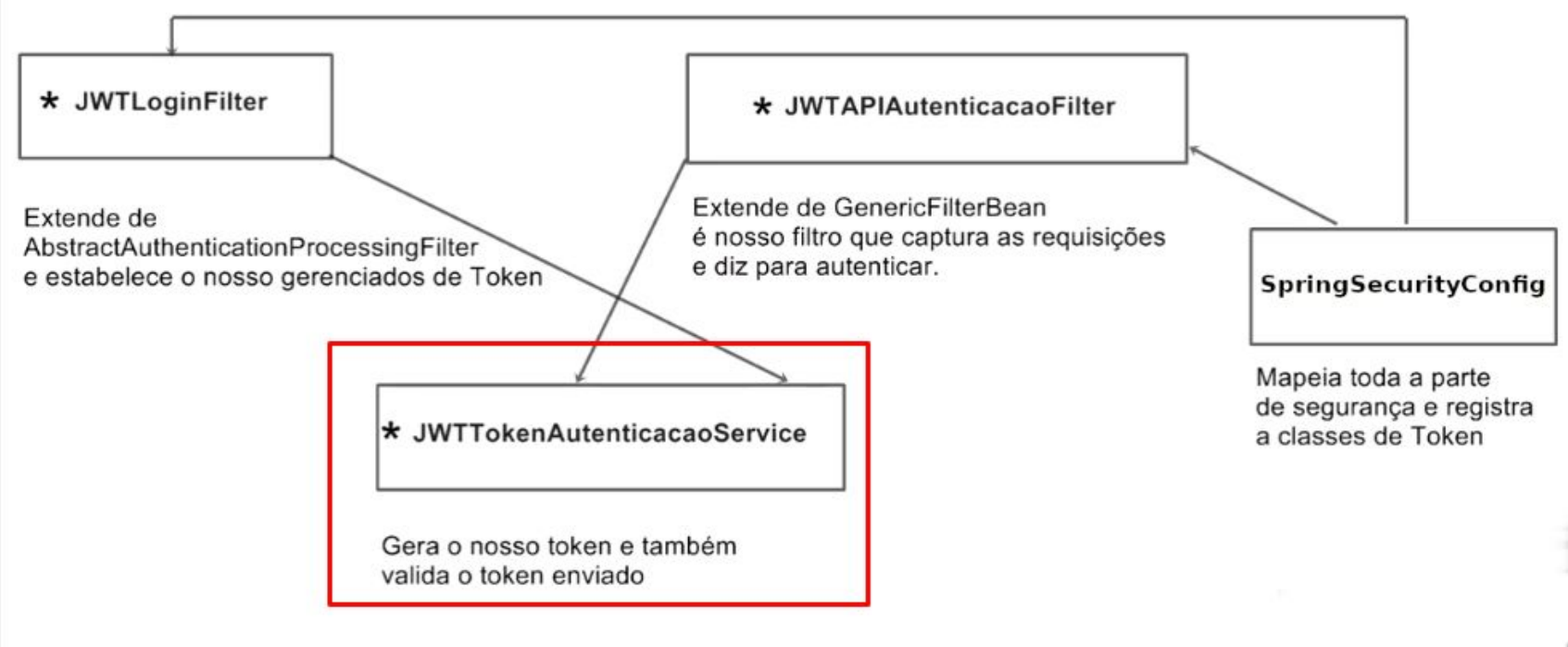
JWT: Classes e suas responsabilidades

- **JWTAPIAutenticacaoFilter**
  - Classe que irá estender de GenericFilterBean, é nosso filtro que captura as requisições e diz para autenticar com token

Etapas de Implementação: JwtApiAutenticacaoFilter

- Classe responsável por filtrar requisições para autenticar, deve herdar da GenericFilterBean
  - Sobrescrever o método **doFilter(ServletRequest request,...):**
    - onde será estabelecida a autenticação para requisição
    - adicionado o processo de autenticação no spring security
    - informar para processo continuar seu fluxo

JWT: Estrutura de Classes



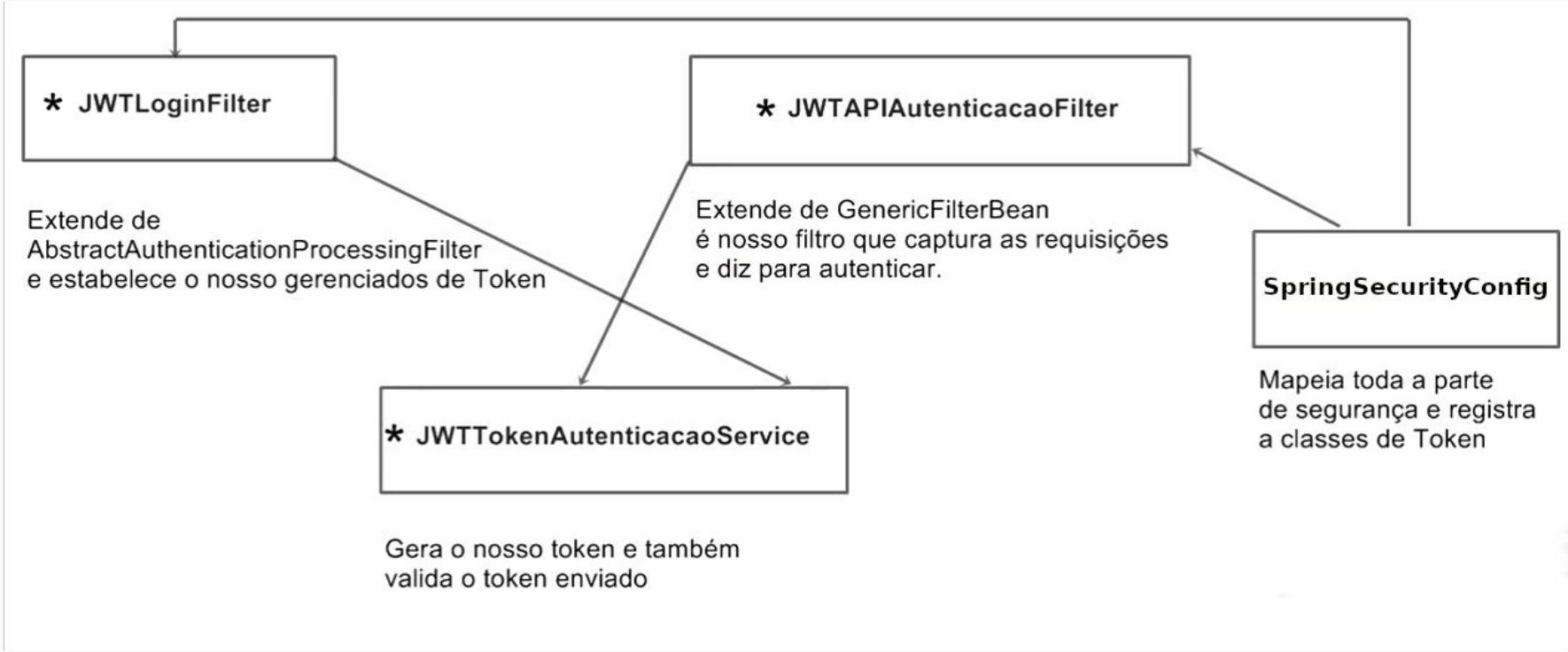
JWT: Classes e suas responsabilidades

- As duas classes JWTLoginFilter e JWTAPIAutenticacaoFilter irão usar em conjunto o serviço provido pela JWTTokenAutenticacaoService
- **JWTTokenAutenticacaoService**
  - Classe que será responsável por gerar e validar o token

Etapas de Implementação: JwtTokenAutenticacaoService

- Classe responsável por gerar e validar o token, considerar os seguinte métodos:
  - **addAuthentication(HttpServletResponse response, String username)**: onde será gerado o token de autenticação e adicionado no cabeçalho de resposta HTTP
  - **getAuthentication(HttpServletRequest request):** método que retorna retornará o usuário válido com token, caso seja inválido retorna null

JWT: Estrutura de Classes



Laboratório

- Projeto Spring-Security
  - **Façam backup dos seus fontes antes de começar**
  - Finalizando a configuração do Spring Security
    - Vídeo 1 - (https://youtu.be/BNjmHzDcnFc)
  - Ajustando a configuração da classe central e realizando testes
    - Vídeo 2 - (https://youtu.be/AjzfYIOqxuo)
  - Repositório Github do projeto desenvolvido em aula
    - https://github.com/DEVin-Clamed/modulo3-semana2

INTERVALO DE AULA

DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

Início: 20:20
Retorno: 20:40

AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

Clique aqui ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.

DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!

SENAI

<LAB365>