



Certified Tech Developer

The Ultimate Degree

Protocolo HTTP

Como você já sabe, este protocolo é orientado a transações. O cliente faz um pedido, enviando ao servidor uma “mensagem” (**request**) com um determinado formato. Posteriormente, ele (servidor) envia a você uma “mensagem” de resposta (**response**). Por exemplo, quando você insere o endereço `www.facebook.com` em seu navegador, você —como cliente— está enviando uma solicitação pela Internet para o servidor ou servidores do Facebook; e este, após receber o referido pedido, processa-o e envia-lhe uma resposta como resultado, por exemplo: mostrar-lhe a página inicial para que possa inserir o seu usuário e senha.

Vamos ver um resumo do que foi visto até agora.

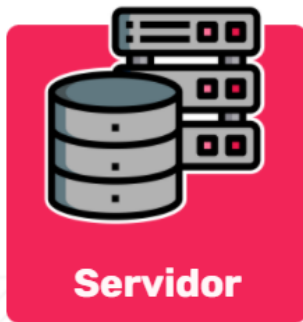
Resumo



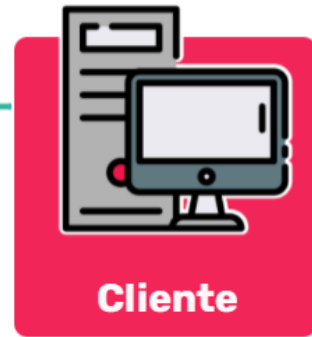
Ver passos



Passos

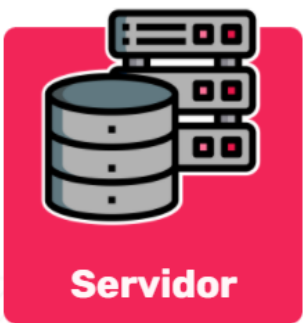


Solicitação/Request



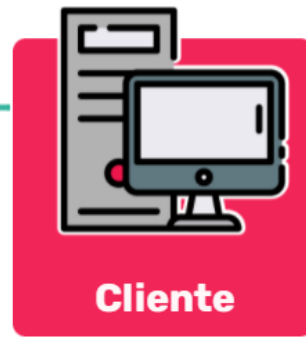
Um dos clientes envia uma solicitação ao servidor.

Passos



Solicitação/Request

O servidor recebe o pedido, realiza as verificações necessárias e, se tudo estiver correto, a processa.



Passos



Por que é importante conhecer este fluxo request - response?

No mundo do desenvolvimento web, a maioria das aplicações tem duas frentes: o front-end e o back-end. Como você pode imaginar, tudo o que acontece no front-end é o que acontece no lado do cliente, enquanto o back-end é tudo o que acontece no lado do servidor (funcionamento lógico e interno do site).

Como são seus formatos?

Cada uma das mensagens entre cliente e servidor possui uma estrutura com a qual são enviadas pela rede.

Request

Method: indica qual método está sendo usado para enviar a mensagem.

URL: endereço para onde o pedido está sendo enviado.

Header (cabeçalho): atributos gerais ou essenciais incluídos. Por exemplo, Auth.

Body: corpo da mensagem utilizada nas solicitações de transmissão de dados.

Response

O **status** é um código que representa o que acontece com uma solicitação. Existem diferentes códigos de status, sendo:

- **1xx** - Respostas informativas indicando que a solicitação foi recebida e está sendo processada;
- **2xx** - Respostas de sucesso, indicam que a solicitação foi processada corretamente, por exemplo: 200 Success;
- **3xx** - Respostas de redirecionamento;
- **4xx** - Erros causados pelo cliente, por exemplo: 404 Not Found;
- **5xx** - Erros causados pelo servidor, por exemplo: 500 Internal Server Error.

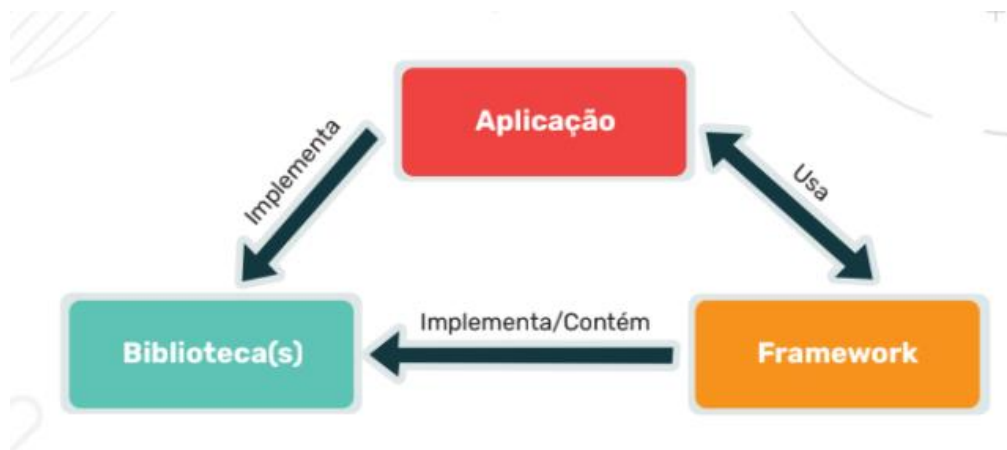
O **header** (cabeçalho) contém atributos gerais ou essenciais;

O **body** (corpo) corresponde à resposta da mensagem.

Então, o que são bibliotecas e frameworks?

Revisando, lembre-se que as bibliotecas são um conjunto de funcionalidades que atendem às necessidades específicas de um determinado projeto e podem ser empacotadas para serem reutilizadas no futuro.

Por outro lado, os frameworks são um conjunto de arquivos e diretrizes que definem a estrutura de como desenvolver um projeto de software. Isso ajuda a economizar tempo e esforços dos programadores.



Spring

Vamos prosseguir para entender mais sobre o Spring e, em seguida, conhecer mais sobre o Spring Boot em detalhes.



O que é o Spring Platform?

O Spring Platform é um conjunto de projetos open source desenvolvidos em Java com o objetivo de agilizar o desenvolvimento de aplicações. Possui uma grande variedade de ferramentas que facilitam nosso trabalho desde o acesso ao banco de dados, infraestrutura, criação de aplicações web, micro serviços, etc.

São eles:

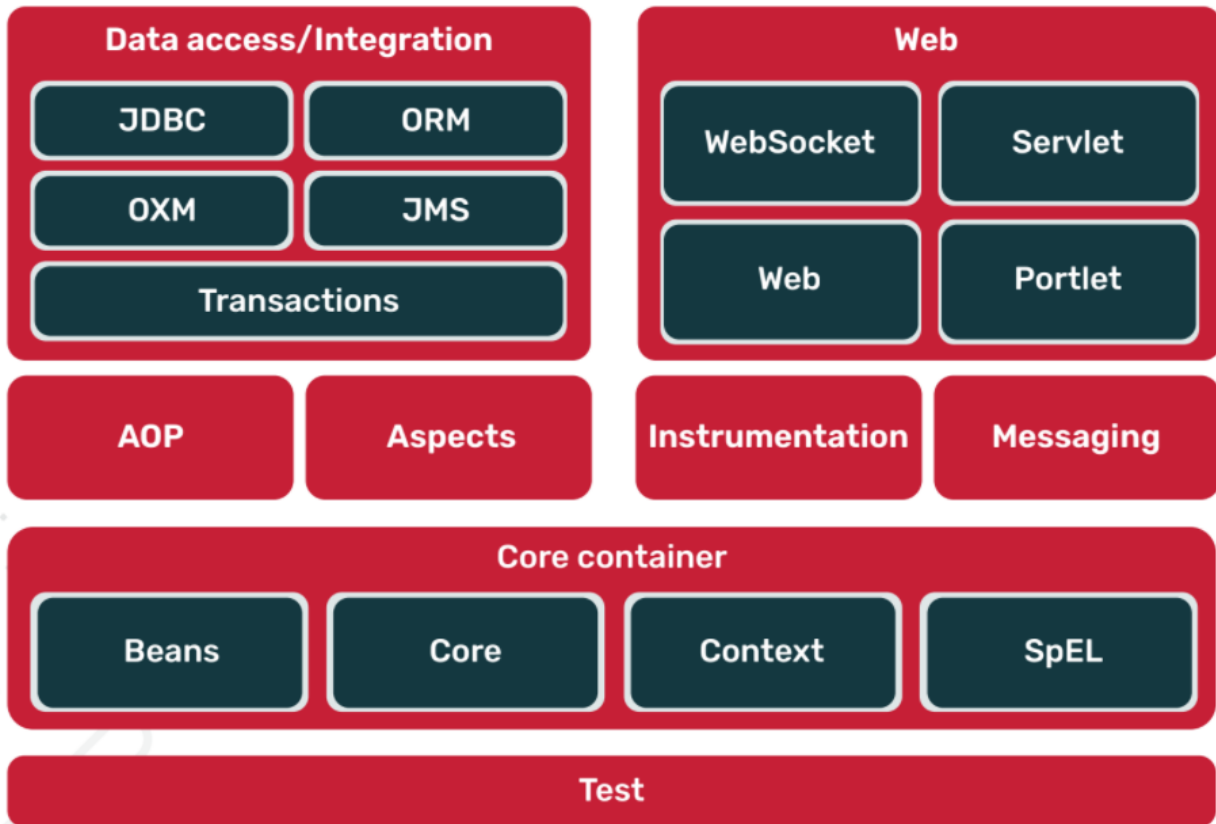
Spring Boot	Spring HATEOAS
Spring Framework	Spring REST Docs
Spring Data	Spring Batch
Spring Cloud	Spring AMQP
Spring Cloud Data Flow	Spring CredHub
Spring Security	Spring Flo
Spring Integration	Spring for Apache Kafka
Spring LDAP	Spring Shell
Spring Roo	Spring Statemachine
Spring Vault	Spring Web Flow
Spring Web Services	

Para saber mais, convidamos você a entrar no [link](#).

O que é o Spring Framework?

Como vimos anteriormente, trata-se de um framework para o desenvolvimento de aplicações e contêineres de inversão de controle. Pode ser utilizado em qualquer aplicação desenvolvida em Java.

O Spring Framework consiste em diferentes módulos que estão agrupados no Core Container, Data Access / Integration, Web, AOP, Instrumentation e Messaging.



O que é o Spring Boot?

O Spring Boot é uma **extensão** do **Spring Framework** que permite a criação fácil e rápida de **aplicações web prontas para produção** com o conceito de just run (apenas executar).

Ele requer uma **mínima configuração** e é complementado por muitos projetos da Spring Platform e bibliotecas de terceiros.

Concluindo, podemos mencionar que permite reduzir muito o tempo de desenvolvimento e aumenta a produtividade.

O Spring Boot nos permite deixar de lado algumas questões de configuração que com o Spring podem ser um tanto entediante. Vamos conhecer suas principais características!

Características do Spring Boot

Web server incorporado

O Spring Boot traz o servidor Tomcat já incorporado, sem a necessidade de instalação prévia. Também é possível utilizar o Jetty e o Undertow.

Porta 8080

Por padrão, o servidor incorporado escuta as solicitações HTTP na porta 8080. Isso significa que quando iniciamos o servidor, devemos acessar a url <http://localhost:8080> para ver nossa aplicação. Podemos configurar outra porta e propriedades diferentes daquelas que são definidas por padrão. Para isso, devemos colocá-las diretamente no arquivo de configuração: **application.properties**.

Spring Boot's starters

Os starters (iniciadores) são usados para diminuir a quantidade de configuração manual das dependências que devemos fazer. Basicamente são dependências do Maven que estão registradas no arquivo POM.xml.

Ao instalar os starters, o Spring Boot se encarregará de fazer com que as dependências sejam ajustadas de forma que possam ser usadas naturalmente em nossa solução com suas versões correspondentes. Eles começam com **spring-boot-starter-*** (onde * é o tipo de aplicação que queremos desenvolver).

Os mais populares:

- **spring-boot-starter-web** é usado para desenvolver web services RESTful com Spring MVC e Tomcat como contêiner de aplicações incorporado.
- **spring-boot-starter-jdbc** é usado para o agrupamento de conexões JDBC. Baseia-se na implementação do conjunto de conexões JDBC do Tomcat.

