

microcurso

# Construindo uma API Rest com Spring Boot, Kotlin, JUnit e PostgreSQL

autor

Welyab Paula



# Formato

- Apresentação
- Exemplos de código



# Lógica de programação

- Desvios condicionais (if, switch)
- Estruturas de repetição (for, while)
- Operadores lógicos (>, <, =, ...)
- Variáveis e tipos de dados



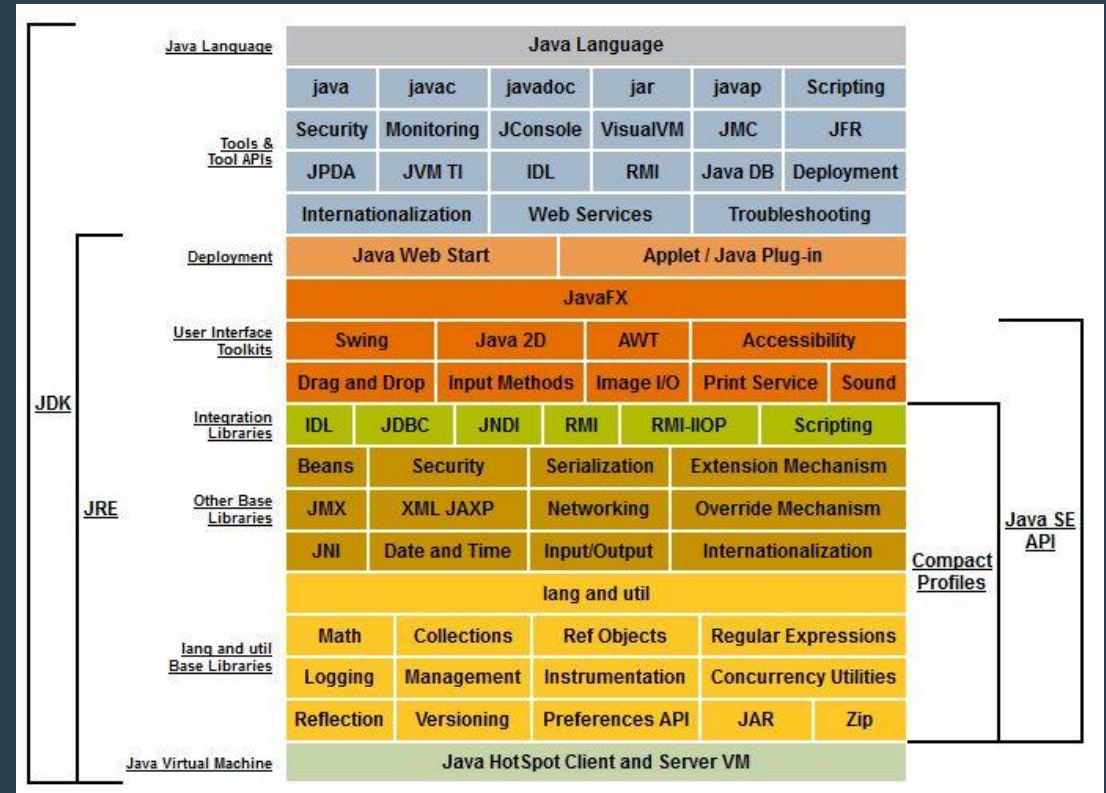
# Algoritmos e estruturas de dados

- Ordenação
- Busca binária
- Listas
- Pilhas
- Árvores
- Mapas
- Conjuntos



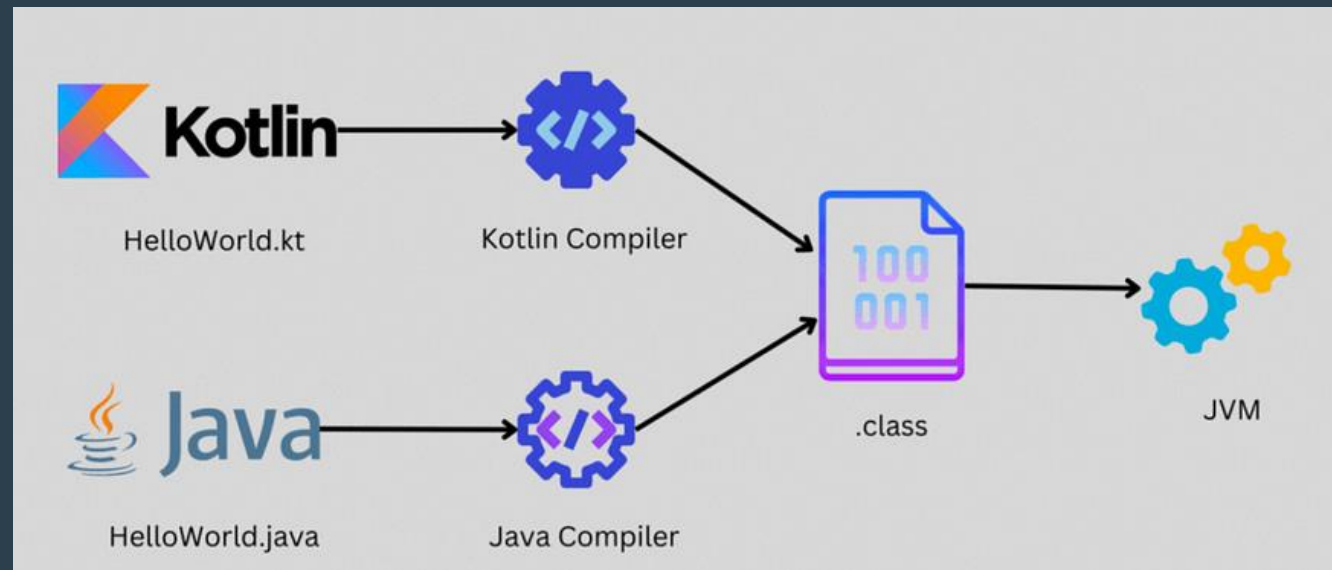
# Linguagem vs Plataforma

- Linguagem Java
- Plataforma Java
- Linguagem Kotlin, Groovy, Clojure, Jython...

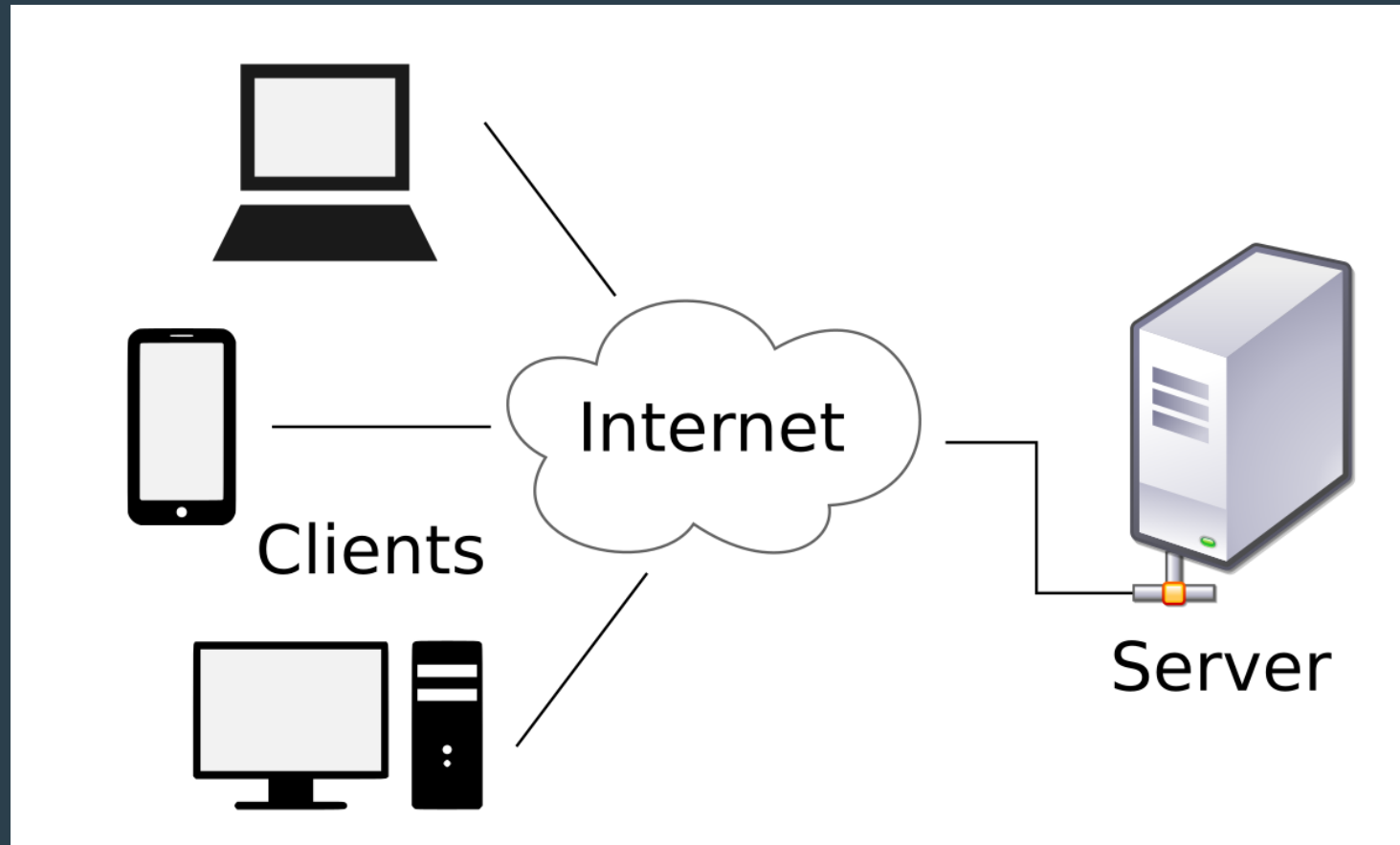




# Bytecode e Java Virtual Machine



# Modelo Cliente/Servidor





# Modelo Cliente/Servidor

The image shows a web browser's developer tools interface, specifically the 'Network' tab. The left pane lists the resources loaded on the page, including the search query and various image files. The right pane shows the details of the selected request, including the URL, method, status, and response headers.

**Nome**

- search?q=modelo+cliente+servidor&oq=mod...
- googlelogo\_color\_92x30dp.png
- tia.png
- 4UaGrENHsxJIGDuGo1OIL3Owp4.woff2
- tia.png
- desktop\_searchbox\_sprites318\_hr.webp
- data:image/png;base...
- data:image/png;base...
- data:image/jpeg;base...
- data:image/png;base...
- data:image/gif;base...
- data:image/svg+xml,...
- data:image/jpeg;base...
- data:image/jpeg;base...
- data:image/gif;base...
- data:image/jpeg;base...
- data:image/png;base...
- data:image/jpeg;base...
- data:image/png;base...
- data:image/png;base...
- data:image/png;base...
- data:image/png;base...
- data:image/png;base...

**Cabeçalhos** Payload Visualização Resposta Iniciador Tempo Cookies

**▼ Geral**

URL Da Solicitação: [https://www.google.com/search?q=modelo+cliente+servidor&oq=modelo+cliente&gs\\_lcrp=EgZjaHJvbWUqBggAEEUYOzIGCAAQRrg7MgYIARBFGDnSAQg3MDMzajBqMagCALACAA&sourceid=chrome&ie=UTF-8](https://www.google.com/search?q=modelo+cliente+servidor&oq=modelo+cliente&gs_lcrp=EgZjaHJvbWUqBggAEEUYOzIGCAAQRrg7MgYIARBFGDnSAQg3MDMzajBqMagCALACAA&sourceid=chrome&ie=UTF-8)

Método De Solicitação: GET

Código De Status: 200 OK

Endereço Remoto: [2800:3f0:4004:800::2004]:443

Política Do Referenciador: strict-origin-when-cross-origin

**▼ Cabeçalhos de resposta**

Accept-Ch: Sec-CH-UA-Platform

Accept-Ch: Sec-CH-UA-Platform-Version

Accept-Ch: Sec-CH-UA-Full-Version

Accept-Ch: Sec-CH-UA-Arch

Accept-Ch: Sec-CH-UA-Model

Accept-Ch: Sec-CH-UA-Bitness

Accept-Ch: Sec-CH-UA-Full-Version-List

Accept-Ch: Sec-CH-UA-WoW64

Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000

Cache-Control: private, max-age=0

Content-Encoding: br

Content-Security-Policy: object-src 'none';base-uri 'self';script-src 'nonce-J6U9VLI1uXDXbZVOj8dwjg' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http;report-uri



# Códigos de Status HTTP

1. Informational responses ( 100 – 199 )
2. Successful responses ( 200 – 299 )
3. Redirection messages ( 300 – 399 )
4. Client error responses ( 400 – 499 )
5. Server error responses ( 500 – 599 )

# Códigos de Status HTTP - 200 Ok

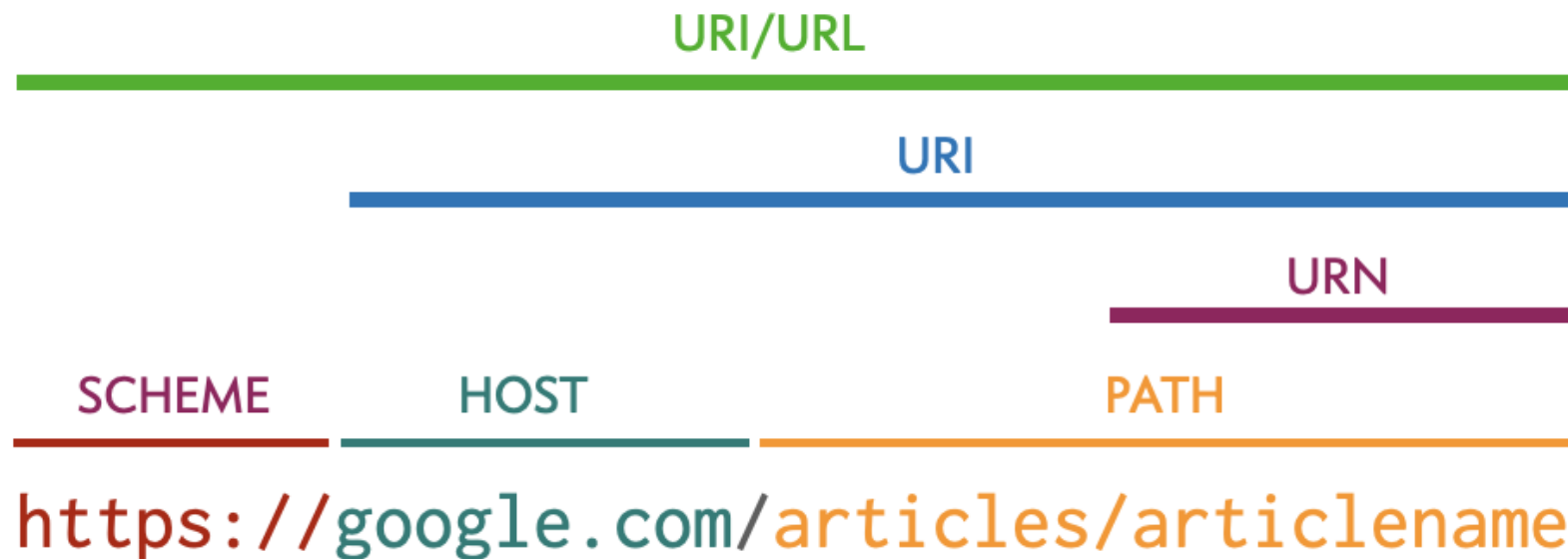
- Indica sucesso na solicitação
  - GET - o recurso foi carregado e transmitido como resposta
  - HEAD - a resposta é transmitida sem o corpo
  - POST - o recurso que descreve o resultado da operação é devolvido do corpo da resposta



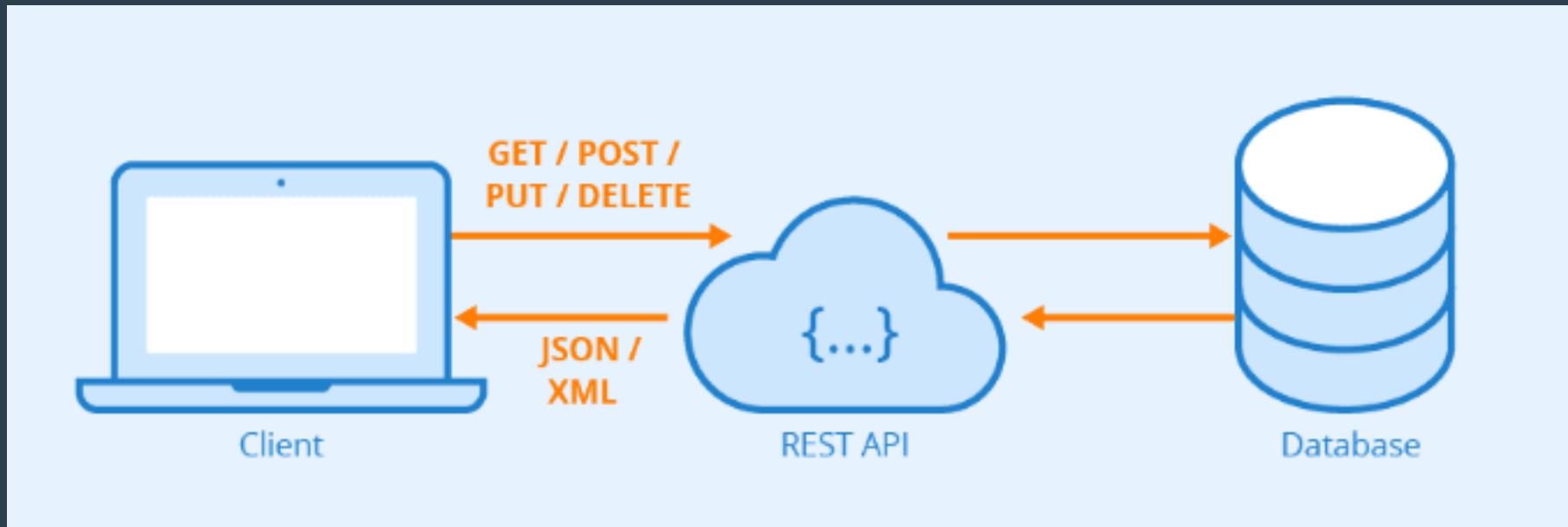
# Códigos de Status HTTP - 3xx/4xx

- 301 Moved Permanently – o recurso não se encontra mais no local indicado
- 404 Not Found - Indica que o servidor não pode encontrar o recurso solicitado naquele momento
- 410 Gone – O recurso foi permanentemente removido

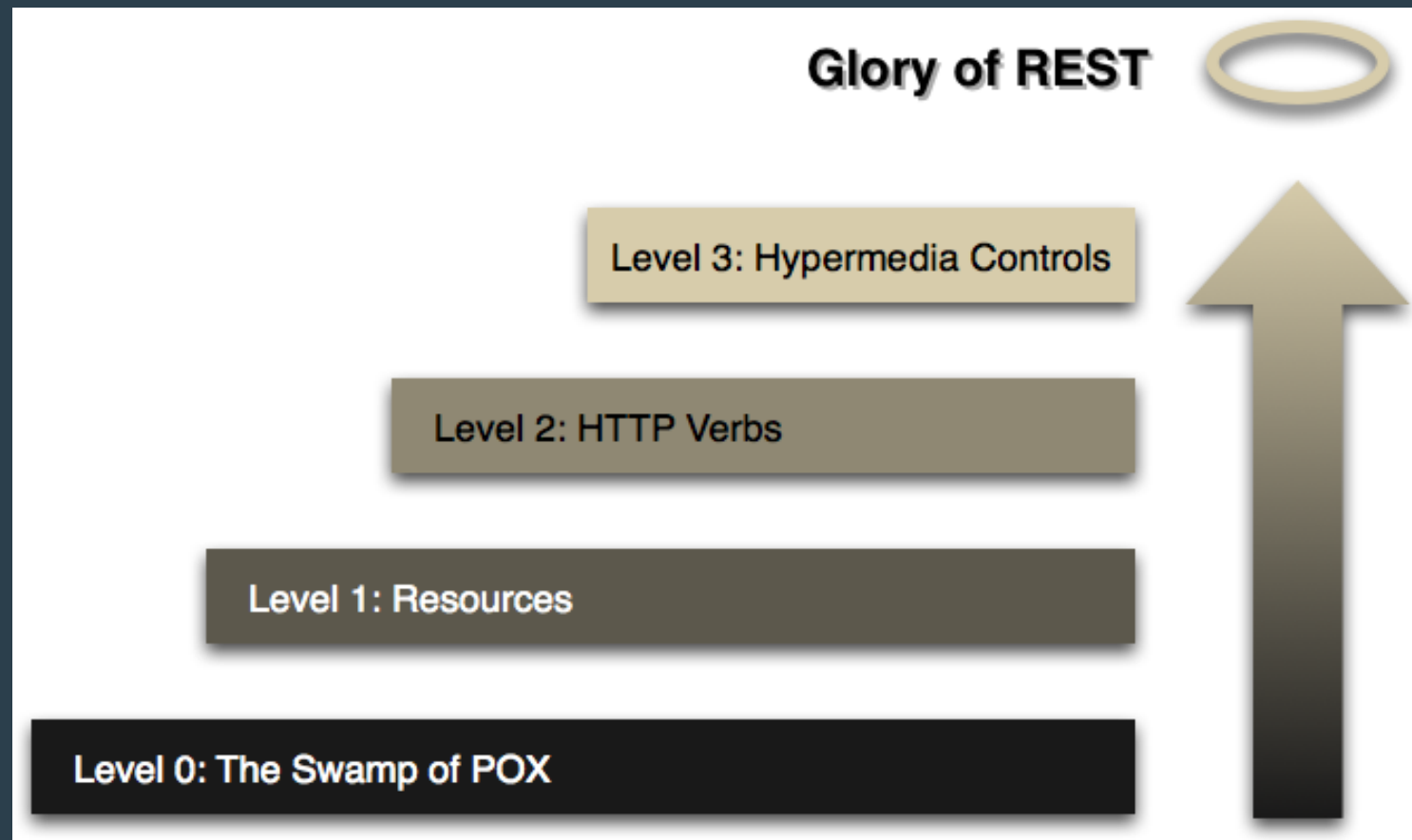
# URI - Uniform Resource Identifier



# API Rest



# Richardson Maturity Model



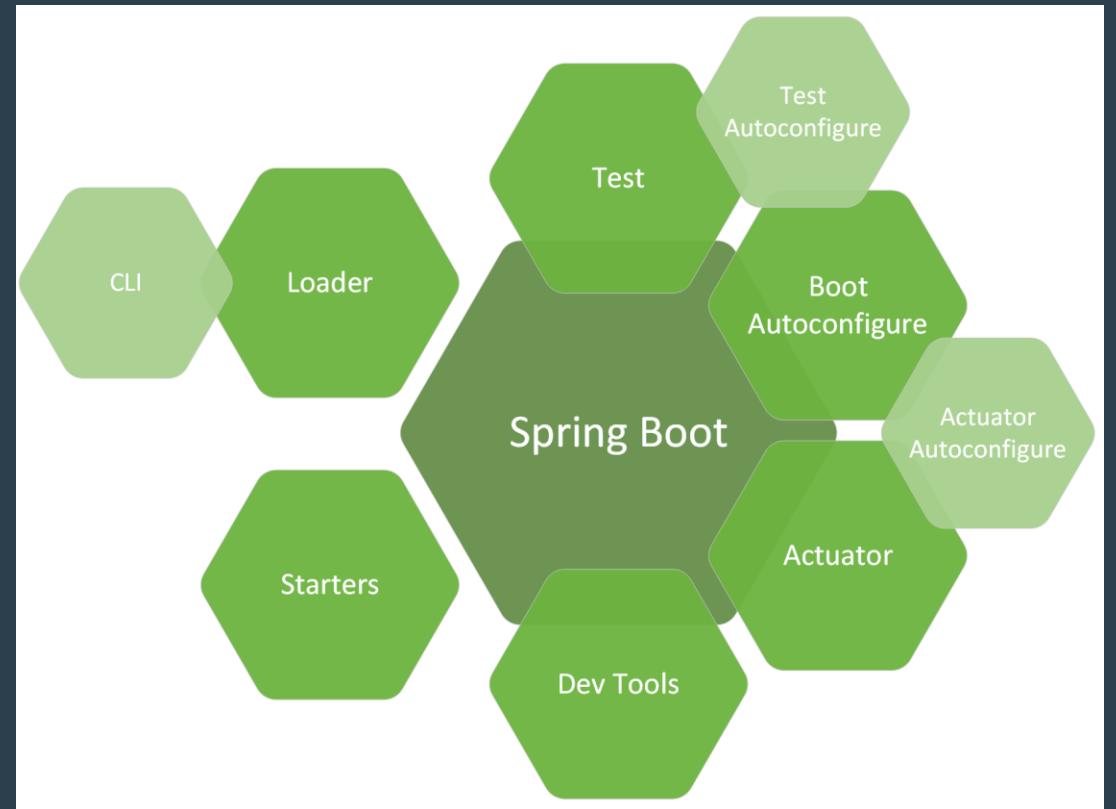


# Outros Protocolos sobre o HTTP

- SOAP – Simple Object Access Protocol
- RRS – transmissão de atualizações de conteúdo
- GraphQL – consulta declarativa de informações

# Spring Framework / Spring Boot

- Configuração sobre configuração



# Gradle

- Ferramenta para automação de build de projetos de software

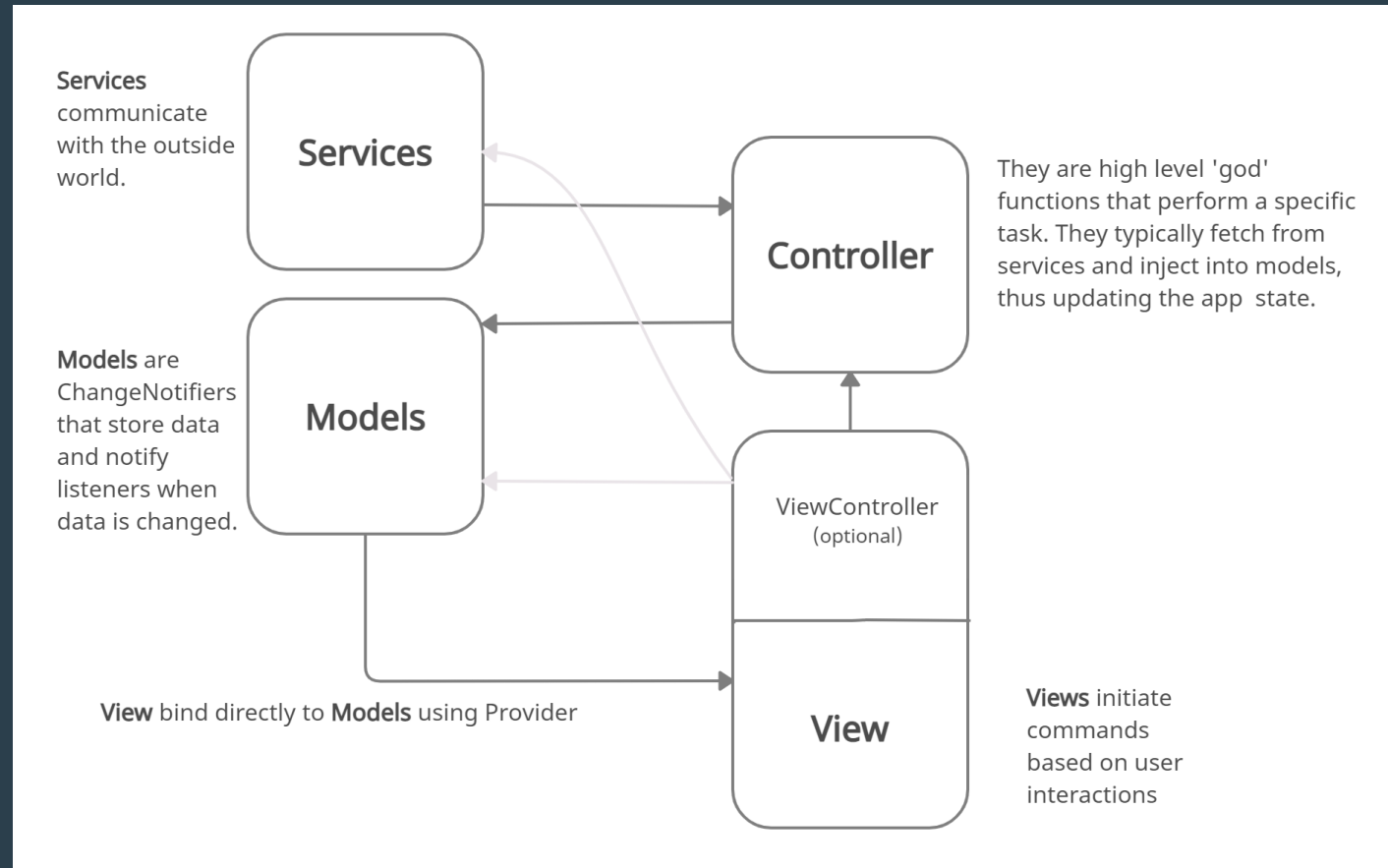


# Ecommerce - Carrinho de Compras

- Listar produtos
- Listar produtos do carrinho
- Adicionar e remover produtos
- Pagamento



# MVCS – Model View Controller Service



# Revisão

- Lógica de programação e algoritmos
- Bom embasamento teórico
- Conhecimento dos padrões de mercado
- Familiaridade com as ferramentas de desenvolvimento