

Curso Técnico Superior Profissional em: Tecnologias e Programação de Sistemas de Informação

2º Ano/1º Semestre

Unidade Curricular: Aplicações Centradas em Redes

Docente: Michael Silva / Hugo Perdigão

INTRODUÇÃO ÀS LINGUAGENS PARA O DESENVOLVIMENTO DE APLICAÇÕES CENTRADAS EM REDES

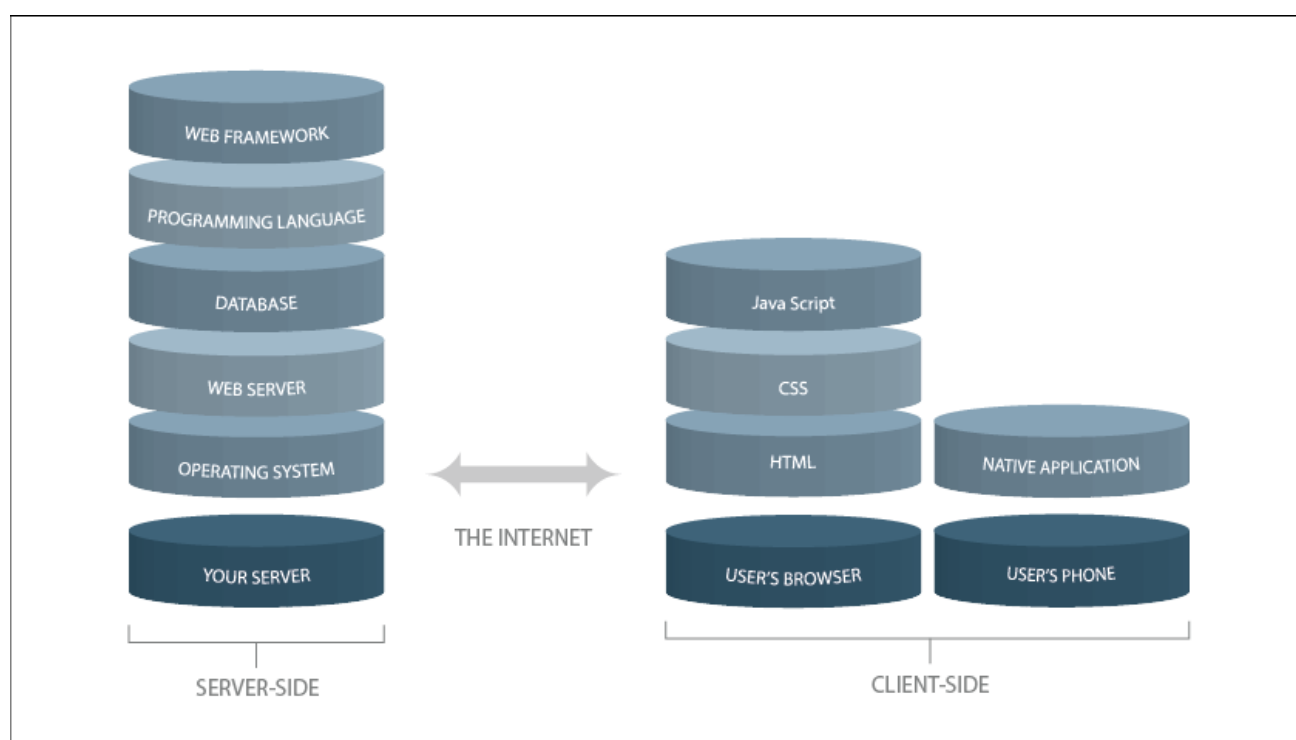


Figure 1 – Exemplo genérico de Stack de Aplicação Web

1.1. Linguagens de apresentação e formatação

- HTML
- CSS
 - SASS

Cofinanciado por:

1.2. Linguagens de script (Front-End)

1.2.1. JavaScript

- jQuery
 - JS mais fácil de usar na web
 - Dominou front end durante anos, ainda é muito usado
 - Bom para apps simples (websites)
 - Desenvolvimento Rápido
- SPA (Single page Application / Aplicativo de página única)
 - Corre no browser, não requer reloading ao usar
 - Renderiza o conteúdo em resposta às ações de navegação (ex: clicar link), sem fazer request ao servidor para receber novo HTML
 - Router para navegação no front-end/cliente
 - Chamam o servidor para receber dados dinâmicos
 - Vantagens:
 - Performance/rapidez: 1º request mais demorado, depois é muito rápida pois interface é separado do servidor
 - Separação do servidor (simplicidade, equipa separada, mesmo backend para apps mobile)

Frameworks mais populares:

Angular

- Criado pela Google, open source
- Muito usado no meio enterprise
- AngularJS (version 1.x.x)
 - 2010
 - Principais inovações:
 - Data binding
 - router (SPAs)
- Angular (version 2+)

React

- Criado pelo Facebook, open source
- 2013
- Biblioteca, pode ser usada como Framework
- JSX: extensão da sintaxe do JavaScript, parecido a XML

Cofinanciado por:



- Framework React Native: permite criar aplicações mobile nativas

Vue

- 2014
- Junta o melhor do Angular e React
- Compreensão mais fácil
- Biblioteca, pode ser usada como Framework
- Single file componentes

1.2.2. TypeScript

- 2010
- Microsoft
- “Is a typed superset of JavaScript that compiles to plain JavaScript”
- Objectivo é simplificar o desenvolvimento de apps de larga escala escritas em JS
- Exemplo de sintaxe extra, definição de tipos:
 - JS: `let lines = 20;`
 - TypeScript: `let lines: number = 20;`
- Usado por bibliotecas e frameworks modernas

1.3. Linguagens de script (Back-End)

1.3.1. Node.js

- 2009
- Javascript no servidor
- Muito popular atualmente
- Programação Assíncrona
- Apropriado para aplicações em tempo real (ex: Chat, partilha e sincronização: editores na cloud, documentos na cloud. apps do mercado financeiro)
- Leve e eficiente: apps de grande escala e carga

1.3.2. PHP

- 1994
- Muito usado na web. Wordpress
- Usada por várias empresas na Madeira
- Framework populares: Laravel, Code Igniter, Yii, Symfony

Cofinanciado por:



1.3.3. Ruby

- 1995
- “Foco na simplicidade e na produtividade... sintaxe elegante de leitura natural e fácil escrita”. Funções sem parêntesis (usa end)
- Ruby on Rails Framework (MVC), 2005

```
# The Greeter class
class Greeter
  def initialize(name)
    @name = name.capitalize
  end

  def salute
    puts "Hello #{@name}!"
  end
end

# Create a new object
g = Greeter.new("world")

# Output "Hello World!"
g.salute
```

1.3.4. Python

- 1991
- Usado em diversas áreas: web, matemática, Inteligência artificial, IoT
- Sintaxe simples, menos linhas, indentação obrigatória em vez de parêntesis
- Django Web Framework 2005, Flask 2010

1.3.5. ASP.NET

- 2002
- Microsoft
- Foi e é muito usada em Portugal

Golang / GO (Linguagem Compilada)

- Criada pela Google
- 2009
- Programas para servidores, web
- Performance
- Eficiente a lidar com concorrência
- Netflix e Uber substituiu partes em Node.js por GO, Dropbox Python -> GO

Cofinanciado por:



Outras linguagens compiladas: Java e C#

Métodos de escolha de linguagem:

- Requisitos da app
- Conhecimento/experiência com a linguagem
- Gosto pessoal

1.4. Linguagens para aplicações nativas e para aplicações híbridas (dispositivos móveis)

1.4.1. Android - Nativas

- Java
- Kotlin
 - 2011/2016
 - JVM
 - Mais fácil, sintaxe mais limpa, ideias de programação funcional
- C/C++ , usando algum Java
- Unity (motor de jogos multi-plataforma)

1.4.2. iOS - Nativas

- Objective C (1984)
 - Usado pela Apple mas não criado
- Swift (2014)

Vantagens das Nativas:

- Performance
- UI específico
- Uso fácil da câmara, acelerômetro, bússola, microfone

Desvantagens:

- Mais que um código base
- Tempo/custo
- Developers especializados

Cofinanciado por:



1.4.3. Nativas multi-plataforma (iOS, android)

100% nativo, acedem à API nativa

- React Native (React)
- NativeScript (Angular, Vue.js, JavaScript ou TypeScript)
- Xamarin (C#, +windows, Microsoft)

1.4.4. Aplicações híbridas

- Algumas funcionalidades nativas

Frameworks

- Ionic (Angular 2+)
- Onsen UI (jQuery, Angular 1 e 2+, React, Vue.js)
- Framework7 (JS, React, Vue.js)
- Cordova

Vantagens:

- Mesmo código para várias plataformas. Pode haver partes específicas no código.
- Tempo / custo

Desvantagens:

- Performance
- Problemas com partes específicas de cada plataforma

Cofinanciado por:



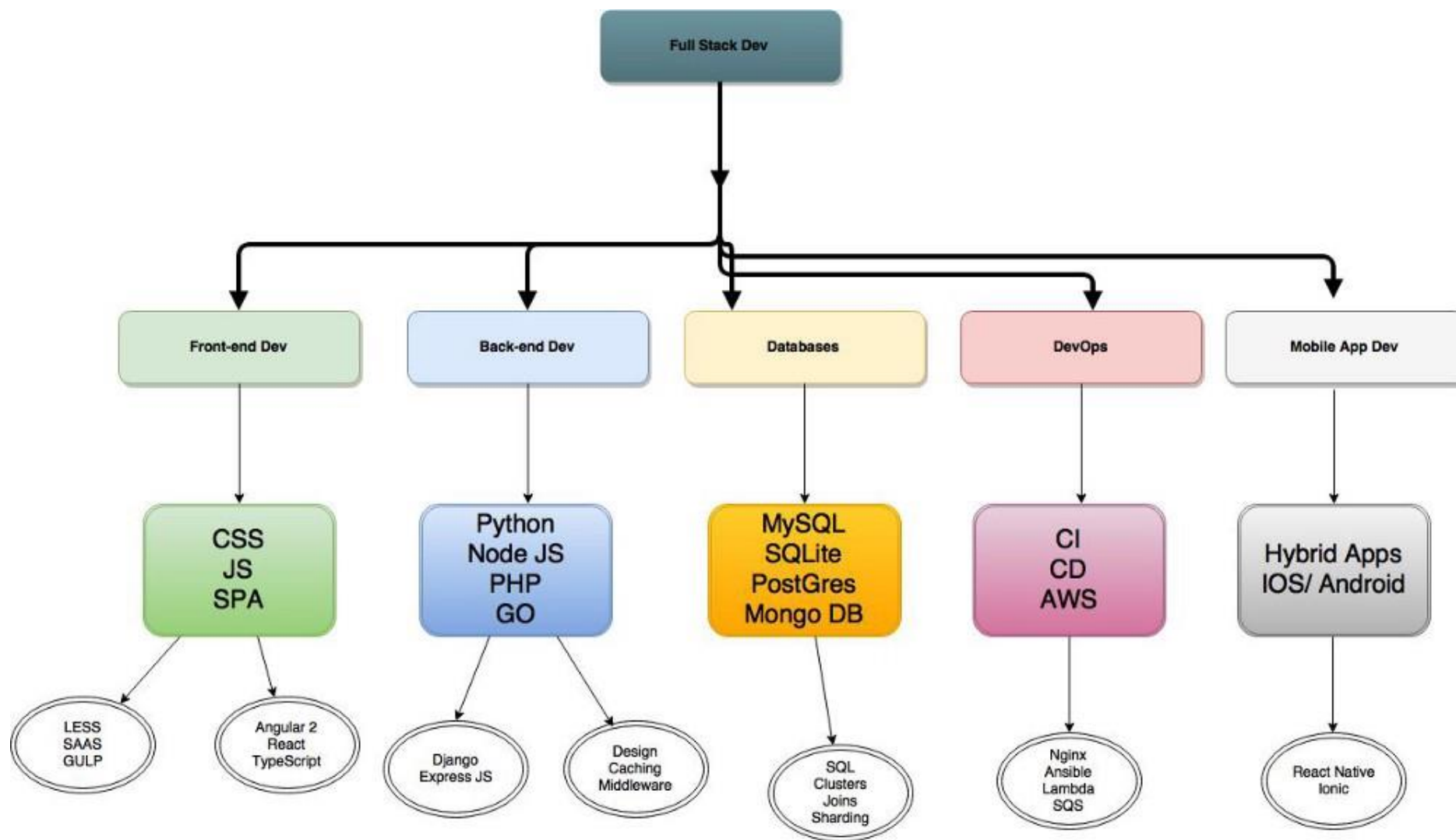


Figure 2 – Exemplos de Linguagens e Frameworks por camadas

Cofinanciado por:

