


# DESENVOLVIMENTO AVANÇADO PARA A WEB

*Do front-end ao back-end*



HTML  
CSS  
JAVASCRIPT  
BOOTSTRAP-VUE  
VUE.JS  
VUE CLI  
VUE ROUTER  
VUEX  
FETCH  
GITHUB PAGES  
REST  
JWT  
NODE.JS  
EXPRESS  
MONGOOSE  
MONGODB  
GITHUB  
HEROKU

Ricardo Queirós  
Filipe Portela



### **EDIÇÃO**

FCA – Editora de Informática, Lda.  
Av. Praia da Vitória, 14 A – 1000-247 Lisboa  
Tel: +351 213 511 448  
fca@fca.pt  
www.fca.pt

### **DISTRIBUIÇÃO**

Lidel – Edições Técnicas, Lda.  
Rua D. Estefânia, 183, R/C Dto. – 1049-057 Lisboa  
Tel: +351 213 511 448  
lidel@lidel.pt  
www.lidel.pt

### **LIVRARIA**

Av. Praia da Vitória, 14 A – 1000-247 Lisboa  
Tel: +351 213 511 448  
livraria@lidel.pt

Copyright © 2020, FCA – Editora de Informática, Lda.  
ISBN edição impressa: 978-972-722-915-4  
1.ª edição impressa: agosto 2020

Impressão e acabamento: Tipografia Lousanense, Lda. – Lousã  
Depósito Legal n.º 472417/20  
Capa: José M. Ferrão – *Look-Ahead*

---

Marcas Registadas de FCA – Editora de Informática, Lda. –



Todos os nossos livros passam por um rigoroso controlo de qualidade, no entanto aconselhamos a consulta periódica do nosso *site* ([www.fca.pt](http://www.fca.pt)) para fazer o *download* de eventuais correções.

Não nos responsabilizamos por desatualizações das hiperligações presentes nesta obra, que foram verificadas à data de publicação da mesma.

Os nomes comerciais referenciados neste livro têm patente registada.



Reservados todos os direitos. Esta publicação não pode ser reproduzida, nem transmitida, no todo ou em parte, por qualquer processo eletrónico, mecânico, fotocópia, digitalização, gravação, sistema de armazenamento e disponibilização de informação, *síto Web*, *blogue* ou outros, sem prévia autorização escrita da Editora, exceto o permitido pelo CDADC, em termos de cópia privada pela AGECOP – Associação para a Gestão da Cópia Privada, através do pagamento das respetivas taxas.

## **Dedicatória**

*Às filhas mais novas dos autores: a Sofia e a Diana!*

Ricardo Queirós e Filipe Portela

## **Agradecimentos**

Em primeiro lugar, gostaríamos de enaltecer o trabalho de qualidade das revisões feitas pela Ana Correia e Laura Faia, da FCA.

Gostaríamos também de agradecer à equipa ioTeam, nomeadamente à Bruna Ferreira pelo desenho do logotipo e ao Daniel Carneiro pelo apoio no desenvolvimento da API incluída no projeto apresentado neste livro.

Finalmente, um agradecimento à Sandra Correia e ao Eng.º Frederico Annes, da FCA, pela simpatia e pelo apoio no decorrer desta nossa aventura editorial.

Ricardo Queirós e Filipe Portela



<b>Os AUTORES .....</b>	<b>XI</b>
<b>PREFÁCIO.....</b>	<b>XIII</b>
<b>0. INTRODUÇÃO.....</b>	<b>1</b>
0.1. O que posso encontrar neste livro?.....	1
0.2. Público-alvo .....	2
0.3. Convenções.....	2
0.4. Organização do livro.....	2
0.5. Suporte .....	3
<b>PARTE I – PROGRAMAÇÃO AVANÇADA PARA A WEB .....</b>	<b>5</b>
<b>1. ROTEIRO DA WEB MODERNA .....</b>	<b>7</b>
1.1. Introdução.....	7
1.2. Roteiro Web: <i>Front-end</i> .....	8
1.2.1. <i>Cascading style sheets</i> (CSS).....	9
1.2.2. JavaScript.....	12
1.2.3. <i>Frameworks</i> Web.....	14
1.2.4. Testes.....	17
1.2.5. <i>Progressive Web Apps</i> .....	19
1.2.6. Ferramentas de <i>workflow</i> .....	23
1.3. Roteiro Web: <i>Back-end</i> .....	24
1.3.1. Documentação .....	25
1.3.2. Testes.....	27
1.3.3. Novas abordagens.....	28
<b>2. O PROJETO ANIMALEC .....</b>	<b>33</b>
2.1. Introdução.....	33
2.2. Análise de requisitos .....	34
2.3. Prototipagem .....	35
2.4. Ferramentas .....	37
<b>PARTE II – FRONT-END.....</b>	<b>39</b>
<b>3. VUE.JS.....</b>	<b>41</b>
3.1. Introdução.....	41
3.1.1. Instalação do Vue .....	42
3.1.2. Instalação de ferramentas auxiliares.....	43
3.1.3. A minha primeira aplicação Vue.....	44
3.2. Instâncias.....	45
3.2.1. Ciclo de vida .....	46
3.3. <i>Templates</i> .....	49
3.3.1. Interpolações.....	49
3.3.2. Diretivas.....	51

3.4. Propriedades calculadas .....	53
3.5. Classes e estilos .....	55
3.5.1. Interligação de classes HTML .....	55
3.5.2. Interligação de estilos <i>inline</i> .....	56
3.6. Renderização .....	57
3.6.1. Renderização condicional.....	57
3.6.2. Renderização cíclica .....	59
3.7. Eventos .....	63
3.7.1. Manipuladores de eventos .....	63
3.7.2. Modificadores de eventos.....	64
3.7.3. Eventos de teclado e rato .....	64
3.8. Formulários.....	65
3.9. Componentes.....	67
3.9.1. Registo de componentes .....	68
3.9.2. Implementação de componentes .....	69
3.9.3. Comunicação de componentes .....	71
3.10. Vue CLI .....	78
3.10.1. Instalação .....	78
3.10.2. Criação de projeto.....	80
3.10.3. Estrutura do projeto .....	85
3.10.4. <i>Build</i> do projeto .....	90
3.10.5. Exemplo prático .....	92
3.11. Vue Router .....	94
3.11.1. <i>Single Page Applications</i> (SPA) .....	94
3.11.2. Criação de uma aplicação .....	96
3.11.3. <i>Matching</i> dinâmico de rotas.....	100
3.11.4. Rotas aninhadas .....	102
3.11.5. Navegação .....	103
3.11.6. Rotas nomeadas .....	105
3.11.7. Redirecionamentos e <i>aliases</i> .....	105
3.11.8. Proteções de navegação .....	106
3.12. Vuex .....	109
3.12.1. Criação de uma aplicação .....	111
3.12.2. Estado.....	112
3.12.3. <i>Getters</i> .....	113
3.12.4. Mutações.....	115
3.12.5. Ações .....	117
3.12.6. Módulos .....	118
3.12.7. Estruturação de uma aplicação.....	119
3.13. Boas-práticas.....	120
3.13.1. Componentes .....	121
3.13.2. Diretivas de ciclo .....	123
<b>A. IMPLEMENTAÇÃO DO PROJETO: <i>FRONT-END</i> .....</b>	<b>125</b>
A.1. Introdução.....	125
A.1.1. Criação do projeto .....	127

A.1.2. Infraestrutura .....	128
A.1.3. Gestão de estado .....	131
A.1.4. API Animalec .....	133
A.1.5. Rotas .....	135
A.1.6. Configuração do projeto .....	137
A.2. Desenvolvimento <i>front-end</i> da aplicação Animalec .....	139
A.2.1. Página principal .....	140
A.2.2. Barra de navegação .....	144
A.2.3. Autenticação e registo .....	146
A.2.4. Perfil do utilizador .....	153
A.2.5. Catálogo de animais .....	160
A.2.6. Detalhe de animal .....	167
A.2.7. Quiz .....	172
A.2.8. Lista de quizzes .....	178
A.2.9. Contactos .....	180
A.3. <i>Back-office</i> da aplicação Animalec .....	181
A.3.1. Gestão de utilizadores .....	183
A.3.2. Gestão de animais .....	189
A.3.3. Gestão de quizzes .....	193
A.3.4. Gestão de questões .....	195
A.4. Distribuição da aplicação .....	198

## PARTE III – BACK-END ..... 203

### 4. REVISÃO DE CONCEITOS ..... 205

4.1. Introdução .....	205
4.2. Padrão <i>Model-View-Controller</i> (MVC) .....	205
4.2.1. Estrutura do MVC .....	206
4.2.2. Iteração do MVC .....	206
4.2.3. Exemplo de estrutura MVC .....	207
4.3. Programação do lado do servidor .....	208
4.3.1. Node.js .....	209
4.3.2. Módulos .....	210
4.3.3. <i>Node Package Manager</i> (npm) .....	212
4.3.4. Comandos npm .....	212
4.3.5. <i>Promises</i> .....	213
4.3.6. Ficheiro <i>Package.json</i> .....	215
4.3.7. <i>Framework</i> Express .....	216
4.3.8. <i>Middleware</i> .....	217
4.3.9. Rotas .....	218
4.3.10. Variáveis globais .....	220
4.4. Paradigma CRUD .....	220
4.5. <i>Application Programming Interface</i> (API) .....	222
4.6. <i>Representational State Transfer</i> (REST) .....	223
4.7. <i>HTTP Status Codes</i> .....	224
4.8. Boas-práticas .....	225

<b>5. WEB SEGURA E DINÂMICA .....</b>	<b>229</b>
5.1. Introdução.....	229
5.2. Cibersegurança.....	229
5.2.1. Confidencialidade, integridade e disponibilidade.....	230
5.2.2. Vulnerabilidade, ameaça, controlo, ataque e risco.....	230
5.3. Segurança na Web.....	231
5.3.1. <i>Injection</i> .....	232
5.3.2. Quebras de autenticação e gestão de sessões.....	232
5.3.3. Utilização de <i>scripts</i> de <i>site</i> cruzados (XSS).....	234
5.3.4. Pedidos externos (CSRF) .....	235
5.3.5. <i>Cross-Origin Resource Sharing</i> (CORS).....	235
5.3.6. Mecanismos de segurança.....	236
5.4. Mecanismos de armazenamento.....	239
5.4.1. <i>Cookies</i> .....	239
5.4.2. <i>Sessions</i> (sessões).....	240
5.4.3. “Tokenização” .....	241
5.5. Métodos de autenticação.....	243
5.5.1. Autenticação baseada no conhecimento.....	244
5.5.2. Autenticação baseada na propriedade.....	244
5.5.3. Autenticação baseada na característica.....	245
5.6. Tratamento de erros e exceções.....	246
5.6.1. Erros .....	246
5.6.2. Exceções.....	246
5.6.3. <i>Promises</i> .....	247
5.7. Validação e verificação.....	247
5.8. <i>Templates</i> .....	249
5.8.1. <i>Templating</i> (mecanismos de modelo) .....	249
5.8.2. <i>Handlebars</i> .....	251
5.8.3. <i>Mustache</i> vs. <i>Handlebars</i> .....	258
5.9. Boas-práticas.....	259
<b>6. BASES DE DADOS .....</b>	<b>261</b>
6.1. Introdução.....	261
6.2. Bases de dados não estruturadas (NoSQL) .....	262
6.2.1. Chave-valor .....	264
6.2.2. Documentos .....	264
6.2.3. Colunas .....	264
6.2.4. Grafos.....	264
6.3. MongoDB.....	265
6.3.1. Estrutura .....	266
6.3.2. Esquema ( <i>schema</i> ) .....	266
6.3.3. Método <i>find</i> .....	268
6.3.4. Projeções.....	269
6.3.5. Método <i>update</i> .....	270
6.3.6. Método <i>insert</i> .....	271
6.3.7. Método <i>DELETE</i> .....	271



6.3.8. Cursor .....	272
6.3.9. Índices .....	272
6.3.10. Conexões MongoDB e Node.js .....	273
6.4. Mongoose.....	275
6.4.1. <i>Object-Documents Mapping</i> (ODM) .....	278
6.4.2. Criar documentos .....	278
6.4.3. Procurar registos.....	278
6.4.4. Procurar e alterar registos .....	279
6.4.5. Apagar e remover registos .....	280
6.5. Ferramentas e interfaces gráficas .....	280
6.5.1. MongoDB Shell .....	280
6.5.2. MongoDB Compass .....	281
6.5.3. MongoDB Robo 3T .....	281
6.6. Mapeamento MySQL e MongoDB.....	282
6.7. Boas-práticas.....	284
<b>B. IMPLEMENTAÇÃO DO PROJETO: BACK-END .....</b>	<b>287</b>
B.1. Introdução .....	287
B.2. Bases de dados .....	287
B.2.1. Esquema ( <i>schema</i> ).....	287
B.2.2. Estrutura.....	288
B.3. Padrão <i>model-view-controller</i> (MVC) .....	289
B.4. Página inicial.....	290
B.5. Servidor.....	291
B.5.1. Aplicação.....	291
B.5.2. Ficheiro DB.js.....	292
B.5.3. Middleware.....	292
B.5.4. Router .....	293
B.6. Rotas.....	293
B.6.1. animal.....	294
B.6.2. auth.....	295
B.6.3. question.....	295
B.6.4. quiz.....	296
B.6.5. user.....	297
B.6.6. userlevel.....	297
B.6.7. email .....	298
B.6.8. home.....	298
B.7. Controladores .....	299
B.7.1. Controlador Animal .....	300
B.7.2. Controlador auth.....	304
B.7.3. Controlador Question.....	307
B.7.4. Controlador Quiz.....	308
B.7.5. Controlador User .....	309
B.7.6. Controlador UserLevel.....	311
B.7.7. Controlador email .....	311
B.7.8. Controlador Home.....	312

B.8. Modelos .....	313
B.8.1. Modelo animal.....	313
B.8.2. Modelo question.....	315
B.8.3. Modelo quiz.....	315
B.8.4. Modelo user .....	316
B.8.5. Modelo userlevel.....	318
B.9. Pasta Views .....	318
B.9.1. <i>Template</i> Email .....	318
B.9.2. <i>Template</i> Home.....	319
B.10. Outros ficheiros e pastas.....	320
B.10.1. Pasta assets.....	321
B.10.2. Pasta messages .....	321
B.10.3. Pasta config.....	323
B.10.4. Ficheiro Procfile.....	324
B.10.5. Ficheiro .gitignore .....	324
B.10.6. Documentação dos <i>endpoints</i> .....	325

<b>ÍNDICE REMISSIVO.....</b>	<b>327</b>
------------------------------	------------

## OS AUTORES

**Ricardo Queirós** ([www.ricardoqueiros.com](http://www.ricardoqueiros.com)) – Doutorado em Ciências de Computadores pela Faculdade de Ciências da Universidade do Porto (FCUP). Docente na Escola Superior de Media Artes e Design (ESMAD) do Politécnico do Porto (P.PORTO), sendo responsável por disciplinas na área da Programação de Computadores, focada para o desenvolvimento de aplicações e serviços para os ambientes Web e os dispositivos móveis. Membro efetivo do *Center for Research in Advanced Computing Systems* (CRACS), uma unidade de investigação do laboratório associado INESC TEC, onde desenvolve atividade científica nas áreas de Integração de Sistemas, Interoperabilidade entre Sistemas de E-learning e Gamificação em Ambientes de Aprendizagem de Programação de Computadores. Membro da Comissão Científica da Unidade de investigação da ESMAD (uniMAD).

A nível de produção científica, destaca-se o trabalho em vários projetos nacionais e internacionais e mais de uma centena de publicações. Integra várias comissões científicas e de revisão de várias conferências e revistas de especialidade, bem como a equipa de júris do Concurso Internacional de Desenvolvimento de Aplicações Móveis (IEEEmadC), promovido pelo *Institute of Electrical and Electronics Engineers* (IEEE).

Autor dos seguintes livros relacionados com programação de aplicações para dispositivos móveis e para a Web, publicados pela FCA: *Android – Introdução ao Desenvolvimento de Aplicações*, *Desenvolvimento de Aplicações Profissionais em Android*, *Introdução ao Desenvolvimento de Jogos em Android* (em coautoria), *Android – Bases de Dados e Geolocalização*, *Android – Desenvolvimento de Aplicações com o Android Studio*, *Criação Rápida de Sites Responsivos com o Bootstrap*, *Android Profissional – Desenvolvimento Moderno de Aplicações e Introdução ao Desenvolvimento Moderno para a Web – Do front-end ao back-end: uma visão global!* (em coautoria).

**Filipe Portela** ([www.filipeportela.com](http://www.filipeportela.com)) – Licenciado em Tecnologias e Sistemas de Informação (2007), mestre em Engenharia e Gestão de Sistemas de Informação (2009) e doutorado em Tecnologias e Sistemas de Informação (2013) pela Universidade do Minho (UM). Professor Auxiliar convidado do Departamento de Sistemas de Informação da Escola de Engenharia da UM, onde tem lecionado e supervisionado vários alunos de mestrado, sendo responsável por unidades curriculares na área da Programação Web.

Investigador Integrado do Centro de Investigação Algoritmi, onde desenvolveu o seu trabalho de investigação de pós-doutoramento sob o tópico Sistemas de Apoio à Decisão Inteligentes e *Pervasive*. A nível científico, possui diversas publicações indexadas nos seguintes tópicos de investigação: Sistemas de Informação *Pervasive*, Apoio à Decisão, Sistemas Inteligentes, *Big Data*, *Data Science*, Inteligência Artificial, *Business Intelligence*, *Data Mining* e *Knowledge Discovery*. Coeditor de diversos livros e revistas, coorganizador

de várias conferências e *workshops* e revisor/membro dos comités de reconhecidas revistas, livros e conferências.

A sua experiência e o seu conhecimento têm potenciado o convite/participação em diversos eventos, projetos e júris técnico-científicos, contribuindo assim para a disseminação do conhecimento junto da sociedade e dos cidadãos. Este facto teve o ponto alto em janeiro de 2018, altura em que fundou a *startup* tecnológica IOtech – Innovation on Technology ([www.iotech.pt](http://www.iotech.pt)).

Coautor do livro *Introdução ao Desenvolvimento Moderno para a Web – Do front-end ao back-end: uma visão global!*, publicado pela FCA.

## PREFÁCIO

Um livro em português para apoiar quem quer aprender a desenvolver aplicações Web avançadas era premente. O vulcão da capa é reflexo da natureza telúrica da experiência de quem se inicia nesta área e se sente, após uma introdução, sobre solo frágil, quente, instável.

De facto, o programador que se inicia no desenvolvimento Web, após dominar a programação convencional, tem de abraçar todo um conjunto de conceitos adicionais: código dividido entre a máquina do utilizador (onde é executado o navegador) e a máquina servidora; linguagens e tecnologias diferentes num caso e noutro; controlo técnico de uma (o servidor) mas não da outra (o computador, telemóvel ou outro dispositivo onde é executado o navegador); meio comunicacional instável e vulnerável entre estas duas partes da aplicação...

Tantos pormenores, tantas transformações do terreno – por isso telúricas – face à programação clássica, face ao foco clássico das ciências da computação e da algoritmia, tão confortável nos seus contextos habitualmente bem delimitados e controlados, que, no fim de qualquer livro de iniciação, fica um vazio, uma frustração: o domínio das transformações não permite encarar o produto final com o orgulho de ser parecido aos sítios Web modernos, às aplicações Web que utilizadores e programadores se habituaram a esperar e a ambicionar.

Este livro dedicado ao desenvolvimento avançado para a Web vem dar robustez e meios a quem se lança por este ambiente vulcânico ativo. Vem combater esse vazio e essa frustração. Parte do contexto atual de desenvolvimento Web apoiado em ferramentas diversas, como *frameworks*, gestores de pacotes, automatizadores de tarefa e outras. A partir desse contexto prossegue pelos territórios diferenciados do *front-end* e do *back-end*, mas esse trajeto é feito com todo o equipamento moderno: as ferramentas diversificadas, as práticas de desenvolvimento e testes, mas também a noção das necessidades balístico-explosivas, das técnicas preventivas e de pancrácio digital tão necessárias ao faroeste da Web atual.

A metodologia pedagógica é talvez o instrumento mais relevante nesse trajeto. Mais do que expor e exemplificar as técnicas e conceitos, os autores conduzem-nos pela implementação de um projeto, que traz o elusivo e abstrato mundo das ideias para o palpável e concreto resultado que os leitores certamente ambicionam ter nas suas mãos e delas outros originar futuramente. Serão certamente essas futuras produções, originadas pelas mãos de todos quantos aprendam com este livro, o seu mais eloquente testemunho.

**Leonel Morgado**

Professor Auxiliar com Agregação do Departamento de Ciências e Tecnologia da Universidade Aberta

Investigador Sénior do INESC TEC

Membro da Direção da Immersive Learning Research Network



---

# O

# INTRODUÇÃO

---

O mundo do desenvolvimento Web está em constante evolução. São inúmeras as tecnologias que gravitam na órbita da Web, o que torna complicado, hoje em dia, encontrar alguma *Web stack* (coleção de software para o desenvolvimento para a Web) predominante. No meio desta revolução, existem linguagens, *frameworks* e ferramentas que importa conhecer e dominar. O objetivo deste livro é apresentar algumas delas de uma forma progressiva e sustentada. Além dos conceitos teóricos obrigatórios, o livro dá ênfase à construção de uma aplicação Web de média dimensão, chamada **Animalec**, que permitirá ao leitor aplicar e consolidar todo o conhecimento apreendido, de uma forma lógica.

## 0.1 O QUE POSSO ENCONTRAR NESTE LIVRO?

Este livro faz parte de uma série que tem como principal temática a programação Web. No livro inicial – *Introdução ao Desenvolvimento Avançado para a Web – Do front-end ao back-end: uma visão global!* – os autores fizeram uma introdução ao desenvolvimento moderno para a Web, com a apresentação de diversos tópicos da área e a inclusão de um projeto exemplo. Este segundo livro tem como principal objetivo partilhar conhecimentos avançados do desenvolvimento para a Web. Neste livro, esses conceitos são divididos em *front-end* (Parte II) e *back-end* (Parte III).

Na parte do *front-end* (Capítulo 3), é apresentada a *framework* JavaScript chamada **Vue.js** (Capítulo 3), que tem como objetivo gerir de forma sofisticada todo o trabalho de *front-end*, dando ênfase à criação de interfaces para o utilizador.

Na parte do *back-end* (Capítulos 4, 5 e 6), destacam-se os tópicos da segurança e do armazenamento de dados através da base de dados MongoDB.

De forma a consolidar todos os conceitos teóricos, é feita a implementação completa do projeto prático **Animalec** (Capítulos A e B), que pretende dar a conhecer tudo sobre animais de uma forma progressiva e divertida.

## 0.2 PÚBLICO-ALVO

Este livro foi escrito tendo como público-alvo todos aqueles que, com bons conhecimentos de programação, pretendem complementar os seus conhecimentos com as tendências tecnológicas associadas ao desenvolvimento de aplicações para a Web, nomeadamente:

- Professores e alunos de cursos/disciplinas de Informática, Ciência de Computadores e Tecnologias Web;
- Profissionais de empresas tecnológicas (p. ex., programadores Web, gestores de projetos, Web designers).

É também uma obra fundamental para todos aqueles que pretendem colocar-se rapidamente a par de todas as novidades introduzidas nos últimos anos a nível do desenvolvimento Web, incluindo todos os que programaram para a Web há mais de 10 anos e que queiram “reciclar” os seus conhecimentos.

Apesar de se partir do pressuposto de que o leitor já tem conhecimentos na programação para a Web, tal não é obrigatório para a leitura e compreensão da obra. Na realidade, são introduzidos de uma forma sustentada todos os conceitos necessários para a criação avançada de aplicações para a Web.

## 0.3 CONVENÇÕES

Ao longo deste livro, optou-se por seguir um conjunto de convenções que facilitem a interpretação do texto e do código. Assim, todos os excertos de código são apresentados no seguinte formato:

```
const bookTitle = "Desenvolvimento Avançado para a Web"
console.log(`Título do livro: ${bookTitle}`)
```

Por sua vez, as notas ou observações importantes poderão ser encontradas no interior de uma secção semelhante à seguinte:



Esta é uma nota ou observação importante.

## 0.4 ORGANIZAÇÃO DO LIVRO

Este livro está organizado em três partes:

- **Parte I: Programação Avançada para a Web** – apresentação de um roteiro para a Web Moderna (Capítulo 1) e introdução ao projeto prático Animalec (Capítulo 2);



- **Parte II: *Front-end*** – introdução à *framework* Vue.js (Capítulo 3);
- **Parte III: *Back-end*** – revisão de conceitos base (Capítulo 4) e introdução a conceitos avançados do lado do servidor, tais como cibersegurança (Capítulo 5) e armazenamento de dados em bases de dados NoSQL (Capítulo 6).

As Partes II e III são complementadas por um capítulo de implementação, respectivamente, do *front-end* (Capítulo A) e do *back-end* (Capítulo B), no qual é aplicado, através do projeto prático *Animalec*, todo o conhecimento assimilado das *frameworks* e dos componentes explanados nos capítulos anteriores (Figura 0.1).

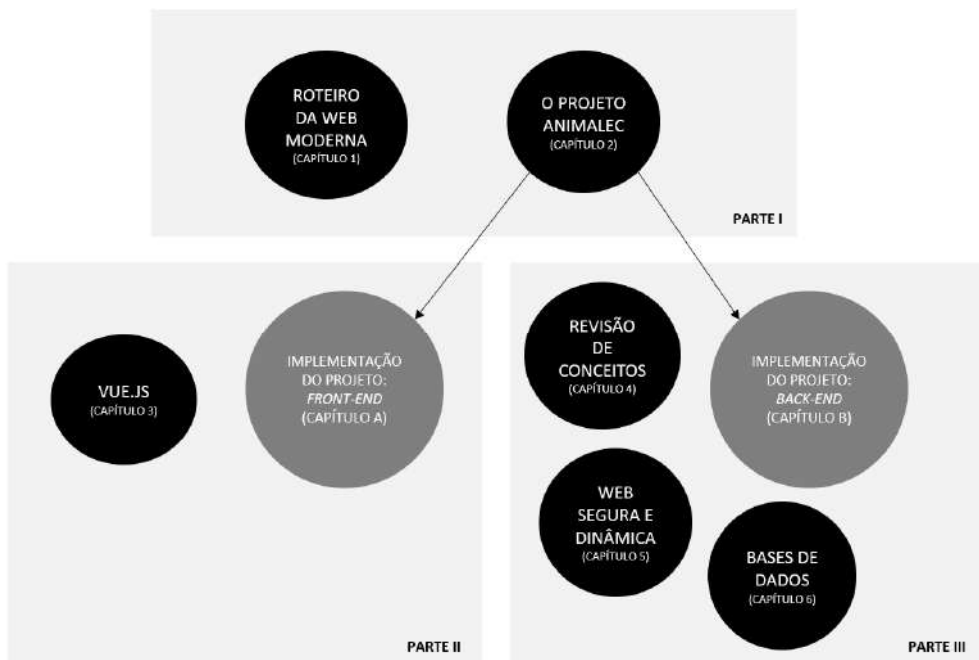


FIGURA 0.1 – Estrutura do livro

Os capítulos podem ser lidos sequencialmente ou, se o leitor assim o preferir, de forma alternada (isto é, sem respeitar a ordem de capítulos apresentada), dado que as poucas dependências entre os mesmos estão devidamente identificadas. O leitor poderá ler apenas uma parte da obra (*front-end* ou *back-end*) ou, até mesmo, um capítulo de uma determinada tecnologia, sem prejuízo da sua compreensão.

## 0.5 SUPORTE

Caso o leitor encontre informação que lhe pareça incorreta ou tenha sugestões em relação ao conteúdo de alguma secção do livro, não hesite em enviar aos autores um

*e-mail* com as suas questões e/ou considerações: sobre a parte de *front-end* para [ricardo.queiros@gmail.com](mailto:ricardo.queiros@gmail.com); e sobre a parte de *back-end* para [filipeportela@iotech.pt](mailto:filipeportela@iotech.pt).

Eventuais alterações e uma errata serão publicadas na página do livro no *site* da editora, em [www.fca.pt](http://www.fca.pt). Na área Downloads, poderá também aceder a todo o código-fonte do projeto Animalec.

---

# PARTE I

---

## PROGRAMAÇÃO AVANÇADA PARA A WEB

A primeira parte deste livro apresenta aos leitores as principais tendências no desenvolvimento Web através de um roteiro simples e prático do estado atual da Web em termos de desenvolvimento *front-end* e *back-end*. É introduzido o projeto prático Animalec, uma aplicação educacional sobre animais, com implementação *front-end* e *back-end*, sendo abordada a gestão do projeto e destacadas as suas várias fases, como: análise de requisitos, prototipagem e edição e controlo de versões do código da aplicação.

**CAP.1** Roteiro da Web Moderna

**CAP.2** O Projeto Animalec



# 1

# ROTEIRO DA WEB MODERNA

O mundo do desenvolvimento Web está a mover-se de modo incrivelmente rápido. Neste capítulo, começa-se por fazer uma análise às principais tendências no desenvolvimento *front-end* e *back-end* de aplicações para a Web. Mais do que uma simples enumeração, a ideia é fornecer ao leitor um roteiro simples e prático do estado atual da Web em termos de desenvolvimento *front-end* e *back-end* e indicar qual será o seu rumo nos próximos anos.

## 1.1 INTRODUÇÃO

O desenvolvimento Web compreende todos os aspetos do desenvolvimento de um *site* ou de uma aplicação para a Web. Se só agora está a entrar neste mundo ou a querer expandir os seus horizontes para novos domínios da grandiosidade da Web, deixamos-lhe aqui o primeiro conselho: (re)visite as suas principais fundações (HTML, CSS e JavaScript). Sem um forte conhecimento destes três pilares, irá sentir muitas dificuldades em conseguir abraçar todos os desafios que surgirão no seu dia a dia enquanto programador Web.

Sendo assim, existe um ponto de partida bem definido quando se pretende desenvolver conteúdos para a Web. Esse ponto de partida assenta no conhecimento de três pilares de *software*:

- **HTML** – linguagem de marcação para estruturar conteúdo;
- **CSS** – linguagem de estilização que permite estilizar *layouts* e conteúdos;
- **JavaScript** – linguagem de programação que permite criar interatividade e dar comportamento a uma aplicação Web.



A editora FCA possui livros que se focam exclusivamente nestas três linguagens: *HTML5* (4.ª Ed. At. e Aum.), *CSS3* e *JavaScript 6* (mais informações em [www.fca.pt](http://www.fca.pt)). Pode também consultar a Parte I do livro *Introdução ao Desenvolvimento Moderno para a Web*, dos mesmos autores desta obra, também da editora FCA.

O desenvolvimento para a Web mudou radicalmente, ou tem vindo a mudar, nos últimos anos. Um bom exemplo é o JavaScript: há 10 anos, esta linguagem era usada so-

mente para validar formulários; hoje em dia, extravasou os limites do *browser* e pode ser usada em quase todos os cenários e plataformas para o desenvolvimento Web (*front-end* e *back-end*), *desktop* ou *mobile*. A própria linguagem (baseada na especificação ECMAScript) tem vindo a evoluir de uma forma meteórica, o que, por um lado, agradou aos seus fiéis seguidores, mas, por outro, tem aumentado a curva de aprendizagem para todos aqueles que querem aceder a este mundo maravilhoso.

Além da linguagem JavaScript, existe atualmente uma panóplia de ferramentas, *frameworks* e paradigmas que a comunidade de programadores e entusiastas deve levar em conta, de forma a lidar com a complexidade das aplicações a desenvolver. Na realidade, quase todos os dias os programadores são confrontados com novos lançamentos, o que torna ainda mais complicado e, ao mesmo tempo, fascinante o mundo da Web.

## 1.2 ROTEIRO WEB: *FRONT-END*

O *front-end* contempla tudo aquilo que é processado e exibido no *browser*. A última década tem trazido muitos conceitos, paradigmas e tecnologias novas, pelo que achamos necessário fazer um breve roteiro para ajudar não só aqueles que dão agora os primeiros passos no desenvolvimento *front-end*, mas também aqueles que querem consolidar os seus conhecimentos. A Figura 1.1 apresenta um roteiro da Web da parte do *front-end*.

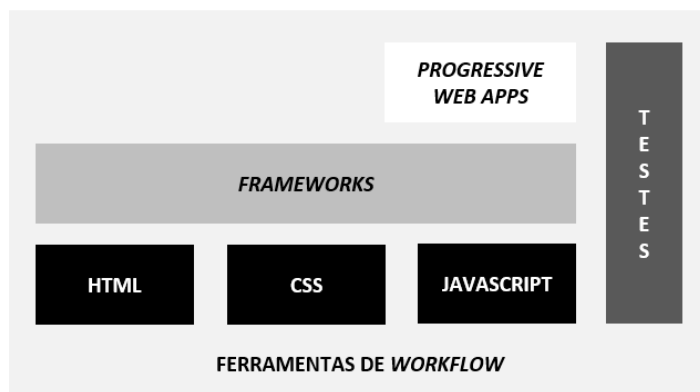


FIGURA 1.1 – Roteiro da Web moderna: parte do *front-end*



O roteiro da Web Moderna foi inspirado em três excelentes projetos: State of JavaScript<sup>1</sup>, State of CSS<sup>2</sup> e Web Developer Roadmap 2019<sup>3</sup>.

<sup>1</sup> Link: <https://2019.stateofjs.com/>.

<sup>2</sup> Link: <https://2019.stateofcss.com/>.

<sup>3</sup> Link: <https://github.com/kamranahmedse/developer-roadmap>.

---

# PARTE II

---

## *FRONT-END*

A segunda parte deste livro apresenta o Vue.js que foi a *framework* JavaScript escolhida para o *front-end* da aplicação Animalec, sendo apresentados passo a passo todos os detalhes da sua implementação.

CAP.3 Vue.js

CAP.A Implementação do Projeto: *Front-end*





Neste capítulo, são apresentados os principais conceitos associados a uma das *frameworks* JavaScript mais populares a nível mundial: o Vue.js. Numa primeira parte, apresentam-se os conceitos mais básicos, como a criação de instâncias, os *templates*, as diretivas, as propriedades calculadas, a renderização condicional e cíclica e os componentes. Depois, introduzem-se algumas ferramentas e bibliotecas. Neste âmbito, apresenta-se a ferramenta Vue CLI (acrónimo de *Command Line Interface*), que visa simplificar o processo de criação de aplicações Vue.js, e as bibliotecas Vue Router e Vuex, que têm como principal missão facilitar a criação e a gestão das *Single Page Applications* (SPA), através, respetivamente, do roteamento baseado em componentes e da gestão de estado centralizada.

### 3.1 INTRODUÇÃO

O Vue.js<sup>1</sup> (doravante denominado apenas **Vue**) é uma *framework* JavaScript progressiva de código aberto, criada em 2014, por Evan You. Tem como principal foco a criação de interfaces gráficas de utilizador e é comumente usada pela comunidade de programadores para:

- Organizar e simplificar o desenvolvimento *front-end*;
- Desenvolver interfaces mais interativas e reativas para a Web.

À data de escrita deste livro (julho de 2020), o Vue encontra-se na versão 2.6.11 e é o repositório baseado em JavaScript mais popular do GitHub<sup>2</sup>.

A sua popularidade deve-se ao facto de ser fácil de instalar, fácil de aprender (curva de aprendizagem pequena) e fácil de integrar com outros projetos e bibliotecas. Esta integração suave e progressiva faz com que o seu uso seja escalável, seja a simples inclusão numa página HTML, como uma biblioteca, ou como uma *framework* num contexto com mais requisitos. Neste último caso, o Vue suporta outros recursos avançados, como o roteamento e a gestão de estado.

---

<sup>1</sup> Link: <https://vuejs.org>.

<sup>2</sup> Link: <https://github.com/vuejs/vue>.

### 3.1.1 INSTALAÇÃO DO VUE

O Vue pode ser instalado de várias formas:

- Através da *tag* `<script>`;
- Através do npm;
- Através do Vue CLI.

A forma mais simples de iniciar o desenvolvimento de aplicações com o Vue é usá-lo como uma biblioteca. Existem duas versões da biblioteca no *site* oficial do Vue<sup>3</sup>:

- **Versão de desenvolvimento** – é uma versão não minimizada. É útil na fase de codificação e *debugging*, pois inclui mensagens de aviso e erro;
- **Versão de produção** – é uma versão minimizada e compactada (33 KB), ideal para a distribuição da versão final da aplicação para o servidor.

Depois de descarregar a biblioteca, basta incluir uma *tag* `<script>` no ficheiro HTML com uma referência para a localização da biblioteca Vue descarregada, conforme demonstra o próximo excerto de código:

```
<html>
  <head>
    <script src="pasta_vue/vue.js"></script>
  </head>
  <body> ... </body>
</html>
```

Como alternativa, pode usar uma referência *online* através de um *Content Delivery Network* (CDN). Para fins de prototipagem ou aprendizagem, use a versão mais recente com a seguinte referência:

```
<script src="https://cdn.jsdelivr.net/npm/vue"></script>
```

Para produção, recomenda-se a vinculação a um número de versão específico, de modo a evitar a quebra das funções das versões mais recentes:

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.6.10"></ script>
```



*Content Delivery Network* (CDN), ou Rede de Distribuição de Conteúdo, em português, é uma rede de servidores que armazenam réplicas do conteúdo de outros *sites* na memória (*cache*) e depois os entregam aos visitantes. Esta ação baseia-se na localização geográfica de forma a conectá-los ao servidor mais próximo e rápido, reduzindo assim o tempo de transferência dos dados (latência).

---

<sup>3</sup> Link: <https://vuejs.org/v2/guide/installation.html#Direct-lt-script-gt-Include>.

---

# PARTE III

---

## *BACK-END*

A terceira parte deste livro apresenta o desenvolvimento da API de servidor através da *framework* Node.js Express e da base de dados MongoDB. Esta parte apresenta todos os detalhes da implementação da camada de servidor e base de dados.

**CAP.4** Revisão de Conceitos

**CAP.5** Web Segura e Dinâmica

**CAP.6** Bases de Dados

**CAP.B** Implementação do Projeto: *Back-end*



# 4

## REVISÃO DE CONCEITOS

Neste capítulo, é efetuada uma revisão de conceitos essencial à compreensão da camada de programação do lado do servidor (*back-end*). Inicialmente, é feita uma breve descrição do padrão MVC (sigla de *Model-View-Controller*), considerando a sua estrutura e iteração; depois, são apresentados o Node.js e a sua *framework* Express, tendo por base o conceito de *middleware* e a utilização de módulos. De seguida, são explicados o paradigma CRUD (acrónimo de *Create, Read, Update, Delete*) e o conceito de aplicações RESTful. No final, são abordados os tipos e códigos de erros mais comuns e feito um resumo das boas-práticas a seguir na implementação de um servidor.

### 4.1 INTRODUÇÃO

A programação Web do lado do servidor é responsável por implementar toda a camada de lógica de negócio de uma aplicação. Esta deve seguir uma estrutura bem definida, uma linguagem de programação robusta e que seja escalável e ser capaz de comunicar com bases de dados. Este tipo de programação é normalmente materializado na construção de uma API que será chamada pela interface (*front-end*).

### 4.2 PADRÃO MODEL-VIEW-CONTROLLER (MVC)

O uso de um padrão estrutural para o projeto tem como principal intuito seguir um modelo predeterminado de desenvolvimento, que procure uma adaptação adequada às necessidades dos programadores. O JavaScript, por si só, não é capaz de fornecer uma base estável para a escrita de código coerente com a qualidade exigida. Como resultado, surgiram várias *frameworks* MVC capazes de fornecer a estrutura e a orientação ao desenvolvimento deste tipo de aplicações.

O MVC é utilizado em muitos projetos devido à arquitetura que possui, o que possibilita a divisão de um projeto em camadas muito bem delimitadas. Cada uma delas – **Model** (modelo), **View** (vista) e **Controller** (controlador) – executa o que lhe é definido e nada mais do que isso. A utilização do padrão MVC apresenta como principal finalidade o isolamento das regras de negócio, da lógica de apresentação, da interface e do utilizador. Tal permite a existência de várias interfaces com o utilizador, que podem ser modifi-

cadascuma sem que haja necessidade de alterar regras de negócio, proporcionando maior flexibilidade e mais oportunidades de reutilização das classes.

O padrão MVC pode ser aplicado a vários tipos de projetos (*desktop*, Web e *mobile*), sendo que a comunicação entre a interface e as regras de negócio é efetuada através de um controlador. A existência deste último torna possível a separação de conteúdos entre a interface e as regras de negócio.



Quando um evento é executado na interface gráfica (p. ex., clicar num botão), a interface comunica com o controlador, o qual, por sua vez, comunica com o modelo (regras de negócio).

#### 4.2.1 ESTRUTURA DO MVC

Os objetivos principais do MVC são a reutilização de código e a separação de conceitos. O MVC constitui-se, então, como um padrão de arquitetura de software que separa a representação da informação, da interação do utilizador com o próprio software. Compõe-se das seguintes partes:

- **Model (modelo – M)** – corresponde aos dados da aplicação, às regras de negócios, à lógica e às funções. O *model* gere o processo de negócio, responde a pedidos do controlador e apresenta os resultados na *view*;
- **View (vista – V)** – pode ser qualquer saída de representação dos dados, como uma tabela ou um diagrama. É possível haver diversas *views* do mesmo dado, como um gráfico de barras ou uma tabela;
- **Controller (controlador – C)** – faz a mediação da entrada, convertendo-a em comandos para o *model* ou a *view*. É baseado em comportamentos, podendo ser partilhado por diversas *views*, sendo responsável por determinar qual é o tipo de exibição na *view*.

O MVC permite criar objetos que “escondem” as suas informações e a forma como são manipulados, de modo a apresentar ao utilizador uma interface simples. Importa referir que o recurso ao padrão MVC poderá complementar a utilização de rotas (*routes*), que são mecanismos que a *view* utiliza para chamar um *controller*, ou seja, a *view* acede a uma rota, que evoca um determinado *controller*.

#### 4.2.2 ITERAÇÃO DO MVC

Abordando a iteração de cada um dos objetos do padrão MVC, temos em primeiro lugar a *view*, que, através do cliente (*browser*), mostra a sua interface visual e recebe os *inputs* do utilizador para serem posteriormente interpretados pelo *controller*.