

# red{info}

6ª EDIÇÃO | 2019

## FERRAMENTAS DO MERCADO



Departamento  
de Informática



Programa de  
Educação Tutorial  
em Ciência da Computação

# APRESENTAÇÃO



## Bem-vindos

É com prazer que apresentamos a 6ª edição da REDinfo, revista criada para difundir sobre os principais temas e inovações que surgem na área de Ciência da Computação, e nesta edição trazemos o tema: Ferramentas do Mercado.

Este é um trabalho do Programa de Educação Tutorial Ciência da Computação (PET-CC), com a participação especial de egressos do programa. Esperamos que o conteúdo aqui presente agregue na vida de cada leitor e que possamos contar com a contribuição do mesmo em nossas próximas edições.

## Boa leitura!

- JOÃO PAULO SILVA DE MOURA.  
ALUNO DO 7º PERÍODO DE CIÊNCIA  
DA COMPUTAÇÃO NA UERN E  
MEMBRO DO PET-CC



## Expediente

### **Universidade do Estado do Rio Grande do Norte**

Faculdade de Ciências Exatas e Naturais

Departamento de Informática

Programa de Educação Tutorial Ciência da Computação

Tutor: Rommel Wladimir de Lima

Petianos: André Luiz de Lima Queiroz

João Paulo Silva de Moura (projeto gráfico),

Marcelo Rossi Oliveira Araújo

# SUMÁRIO

---

## JavaScript

03

Por Claudivan Barreto - Desenvolvedor Fullstack na ViBe e egresso do curso de Ciência da Computação UERN

## Entrevista

05

Com a participação do Backend Bruno Agenor Soares Santana

## Ionic X React

08

Por Giovanna Andrade - Application Development Analyst na Accenture e egressa do curso de Ciência da Computação UERN

## Meu primeiro Commit

11

Por Jefferson Ximenes - Graduando do curso de Ciência da Computação e membro do PET-Ciência da Computação

# JavaScript

## Claudivan Barreto



JavaScript é uma linguagem de programação interpretada e baseada em objetos, é bastante utilizada para manipulação de páginas Web, mas também é usada em outros ambientes como servidores e banco de dados. O JavaScript segue o padrão ECMAScript que é a especificação da linguagem. Portanto, sempre que nos referirmos a versões da linguagem as pessoas dizem ECMAScript. A versão atual do JavaScript é o ECMAScript 9 ou ECMAScript 2018.

O JavaScript é indiscutivelmente a linguagem de programação mais open-source que existe. A evolução da linguagem é tratada pela TC39, um comitê composto por várias empresas, incluindo os principais fabricantes dos navegadores. As engines JavaScript fizeram um tremendo progresso, evoluindo de interpretadores lentos para compiladores rápidos.

Para uma linguagem ser amplamente utilizada ele segue normalmente dois requisitos. Primeiro, a linguagem tem que ser melhor documentada e suporta-

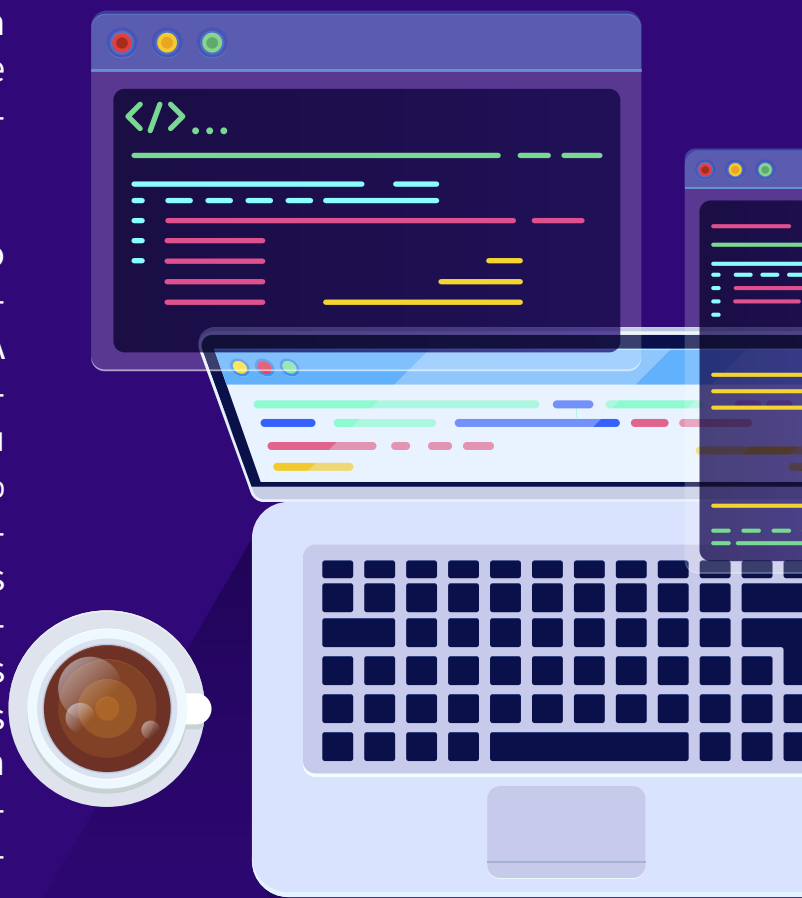
da. Segundo, muitos programadores devem a conhecer, o que é importante sempre que você precisa contratar alguém. Atualmente, a documentação e o suporte para JavaScript vêm em todos os formatos e tamanhos: livros, podcasts, postagens em blogs, boletins informativos por e-mail, fóruns e muito mais. A demanda por desenvolvedores em JavaScript vem crescendo exponencialmente nos últimos anos, mas suas fileiras também estão aumentando constantemente.

Grandes tecnologias vêm sendo desenvolvidas em JavaScript, alguns estão se destacando e tendo bastante aceitação pela comunidade, que é o caso do Angular, React, Node.js e o MongoDB. Os dois primeiros são frameworks para o front-end, (parte da aplicação que interage diretamente com o usuário) que automatizam o desenvolvimento e permitem a reutilização de código com base em componentes. O Node.js é uma plataforma para construir aplicações de rede rápidas e escaláveis, seu grande forte é ter um modelo de I/O não blo

queante que o torna ideal para aplicações em tempo real. E o MongoDB é um banco de dados NoSQL baseado em documentos com alta performance e disponibilidade e de fácil escalabilidade.

Segundo o GeekHunter, em 2018 o salário médio no Brasil para desenvolvedores JavaScript era de 4.800.00 reais. A pesquisa informa também que desenvolvedores que conhecem Angular ou Node.js podem ter um valor de 10% acrescido no salário em 2019. As empresas estão a procura de desenvolvedores que tenham um conhecimento profundo e experiências sólidas nas principais ferramentas JavaScript, além disso elas procuram profissionais que sejam proativos, tenham comunicação assertiva, trabalhem em equipe, falam fluentemente inglês, entre outros.

Se você está estudando JavaScript ou pretende estudar, continue, vá em frente, o mercado está a sua procura, você pode ser o próximo.



# Entrevista

## Com Bruno Santana

### 1- O que é backend? quais suas possibilidades?

É a subdivisão do desenvolvimento web responsável pelo processamento em servidor, ou seja, em sistemas web onde é usada a arquitetura cliente-servidor tudo o que não acontece no dispositivo do usuário é considerado como a parte de trás do sistema, como por exemplo: o controle de quem é o usuário que está acessando o sistema; o envio de um e-mail; consulta e armazenamento de informações em bancos de dados; e principalmente a preparação de conteúdo dinâmico que será usado no frontend.

Mas é claro que, com a tecnologia e ferramentas de desenvolvimento que temos hoje é possível fazer tudo o que listei acima diretamente no frontend, porém isso seria tão seguro e eficiente quanto transportar um sofá no teto de um carro enquanto dirige sentado nesse mesmo sofá.

### 2- O que te levou a se tornar um desenvolvedor Backend?

Na verdade, quando comecei como pro-

gramador eu preferia o desenvolvimento para desktop, que é voltado para os programas que precisamos instalar para usar, até que vi o desenvolvimento web como uma área em ascensão e com um mercado de trabalho carente, principalmente devido à possibilidade de atingir não só computadores, mas qualquer dispositivo que suporte acesso à rede e um navegador, e isso sem serem necessárias adaptações para plataformas específicas.

A escolha pelo backend ocorreu quando me dei conta de que eu preferia dar prioridade em desenvolver algo funcional, seguro e rápido e por último me preocupar em como isso seria visto e usado. O que não significa que eu considere a apresentação de um sistema algo sem importância, mas ao inverter essas prioridades são grandes as chances de se fazer um sistema “oco”, dado que tempo é um recurso limitado e feito sempre vai ser melhor do que perfeito.

### 3- Quais as vantagens e desvantagens de ser um desenvolvedor Backend?

A maior vantagem é desenvolver para um ambiente totalmente controlado, o servidor, ou servidores onde irá ser executado o código backend é preparado para que tudo funcione como deve e de

uma forma constante.

Um desenvolvedor backend vive nas sombras, e com isso uma desvantagem é que tudo o que é feito por ele quase nunca é visto, a maior parte dos usuários nem imagina que existe algo além daquilo que ele pode ver.

#### **4- Quais eventos existem voltados para Backend na nossa região?**

Eventos a nível estadual focados em backend, eu sinceramente não vejo muitos, quando ocorrem são eventos nacionais que escolhem Natal como sede e provavelmente existem eventos menores e pouco divulgados mais voltados para as instituições que os organizam.

A nível de nordeste é mais comum termos eventos em Pernambuco e Fortaleza, como: Python Nordeste; PHPeste; Tropical Conf; JSday. E esse ano o Darkmira Tour PHP, que é um dos maiores eventos nacionais, acontecerá no dia 8 de junho em Fortaleza-CE.

#### **5- Quais suas expectativas futuras em relação ao desenvolvimento Backend?**

Minha expectativa para o desenvolvi-

mento no geral é que tenhamos um maior amadurecimento da área e dos desenvolvedores para que cada um possa escolher e seguir uma especialidade, seja ela infraestrutura, banco de dados, frontend ou backend. Ainda temos muito da cultura do “menino da informática” que precisa fazer de tudo, ter um pouco de conhecimento sobre como tudo é feito é importante, mas é preciso se dedicar em um desses nichos para ser um bom profissional e desenvolver sistemas robustos fazendo parte de uma equipe maior onde cada um vai fazer o que sabe melhor.

#### **6- Quais dicas você pode dar para uma pessoa que quer iniciar seus estudos em desenvolvimento Backend?**

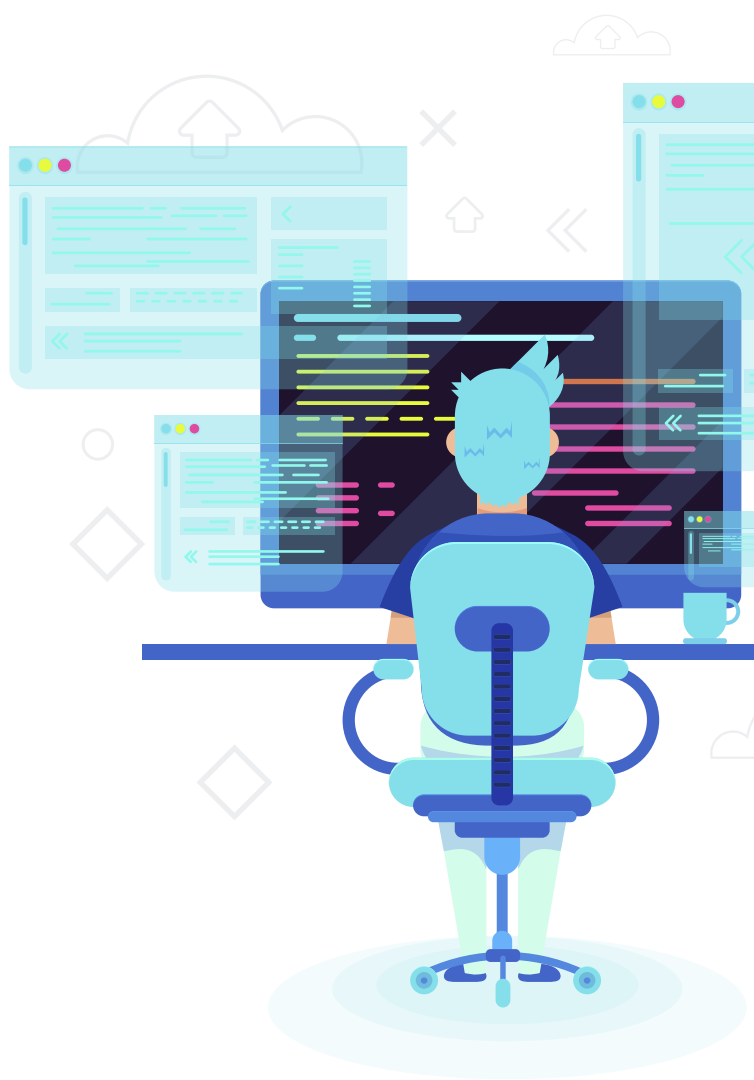
Alguém experiente para lhe orientar é bastante importante, entretanto todo bom desenvolvedor precisa ser autodidata e nunca deixar de se atualizar. Cada pessoa tem seu estilo de aprendizagem, uns preferem videoaulas, outros gostam de apostilas e livros digitais, e como no meu caso tem quem prefira livros físicos. Meu conselho é que cada um descubra quais métodos funcionam melhor. Para aprender não se prenda nem discrimine nenhuma linguagem ou tecnologia específica, não existe melhor ou pior, todas elas tem características em



comum, que é o que se deve focar em conhecer, e todas elas tem especialidades que vão ser entendidas e dominadas com a experiência do uso.

Tenha paciência ao aprender, procure entender o motivo de estar fazendo cada coisa, e não simplesmente faça porque é assim que funciona, isso vai permitir que os sistemas feitos por você sejam mais confiáveis, é importante saber como a linguagem realiza acesso ao banco de dados mesmo que depois um framework permita que você não se preocupe com isso. E já tocando no ponto, quando já estiver confiante sobre o que aprendeu em uma tecnologia procure por frameworks, eles são essenciais para a produtividade.

E meu último conselho é que um desenvolvedor backend nunca deve confiar nos usuários do seu sistema, é preciso partir da premissa de que ou o usuário ou é muito inexperiente e irá comprometer o sistema por não saber o que faz, ou o usuário tão esperto que irá comprometer o sistema por más intenções. Por isso sempre envie apenas o necessário para o frontend e sempre verifique se as informações vindas dos usuários são realmente o que o sistema espera que sejam.





# Ionic X React

## Giovana Andrade



O Ionic é um framework front-end para o desenvolvimento de aplicações mobile híbridas fornecendo componentes HTML, CSS e JavaScript otimizados para dispositivos móveis, como também ferramentas para a criação de aplicações interativas. Ele utiliza o Apache Cordova para construção da aplicação de um Webview, e por meio desta, é possível apresentar conteúdos da web através de aplicativos. Toda a criação e desenvolvimento de aplicações que utilizam o Ionic acontece, principalmente, por meio do utilitário de linha de comando o CLI.


Diferente do Ionic, o React não é um framework e sim uma biblioteca JavaScript, para o desenvolvimento front-end de aplicações, construindo interfaces visuais. Mesmo sendo uma “simples” biblioteca, o React conta com um incrível ecossistema que, a partir desse conjunto, torna-se possível a construção de aplicações completas. Dentre os itens do ecossistema estão: JSX, ES2015, Redux, axios/fetch.

Essas duas tecnologias estão entre as

mais utilizadas considerando o seu meio. Principalmente quando são colocadas frente a necessidade de escolha para o desenvolvimento. O Ionic por exemplo supre às necessidades para a criação de interfaces mobile, possibilitando a criação de aplicações híbridas fornecendo suporte multiplataforma: Android e IOS, com a construção de apenas um código fonte.

O React é tido como uma tecnologia simples e leve para o desenvolvimento, principalmente levando em consideração seu concorrente direto o Angular. Como exemplo temos a importação de vários componentes e módulos providos pelo próprio Angular na criação de uma aplicação Angular, sejam utilizados ou não. Já no React ocorre às importações básicas para uma aplicação inicial, às demais criações e importações estão designadas ao desenvolvedor.

O Ionic permite integração com outros frameworks e, inicialmente, essa possibilidade era restrita apenas ao Angular.



Atualmente o Ionic está na versão 4.4.0 e foi totalmente reconstruído trazendo melhorias e também novas funcionalidades como sua compatibilidade multi-framework, permitindo integração com Angular, React e Vue. Além disso, também possui uma massa de componentes já criados para facilitar e agilizar o desenvolvimento.

Falando em componentes este é um dos pontos fortes do React. Mesmo não havendo componentes já estruturados como no Ionic, pode-se simplesmente criar seu próprio componente para ser utilizado em qualquer lugar da aplicação. Deixando o desenvolvedor aberto às possibilidades de utilização do mesmo, e não a restrição de um componente já existente. Além disso, também possui como vantagem sua velocidade, pois o React introduz o conceito de virtual DOM, o qual guarda o estado de cada componente garantindo que haja o mínimo possível de escrita e leitura no próprio DOM, já que o mesmo só é propriamente executado após o Virtual DOM analisar as mudanças e encontrar o melhor caminho para realizar a atualização.

Mesmo divergindo em algumas características como ser framework ou biblioteca, e reutilização de outro frameworks front-end, ainda é possível fazer um comparativo de similaridade entre essas tecnologias. Como principais pontos, podemos destacar:

- Ambas são PWA (Progressive Web Apps) que, nada mais é do que uma metodologia de desenvolvimento em que torna a experiência de uso de uma página web semelhante a de um aplicativo mobile;
- Multiplataforma;
- Utilizam o JavaScript como linguagem de programação;
- São SPA (Single-Page Application), aplicações que rodam em uma única página, com seu conteúdo podendo ser carregado de uma única vez ou dinamicamente.

Já com relação a escolha de determinada tecnologia para o desenvolvimento, quando não é um requisito do cliente, cabe ao desenvolvedor analisar a que melhor se adequa ao projeto. Mesmo que tenha uma tecnologia ou framework em que possua um conhecimento seguro sobre, pode não ser suficiente

quando o desempenho e outros demais fatores são levados em consideração. Então não é uma boa prática sentir-se “acomodado” quando falamos sobre desenvolvimento, é sempre útil e uma ótima prática buscar novos conhecimentos e manter-se atualizado, além de ser adepto a mudanças.



# Meu primeiro commit

## Jefferson Ximenes



### Visão geral

Trabalhar em equipe nem sempre é fácil, algumas atividades tais como desenvolvimento de projetos e edição de arquivos podem se tornar bastante complicadas, pois alguém pode sobrescrever algum arquivo, podendo comprometer o projeto. Além da dificuldade de gerenciar grandes projetos, sem o controle do que foi feito anteriormente. O Git é gratuito e de código aberto, com o intuito de ser um sistema de controle de versões distribuídas.

### Instruções

#### 1. Como configurar o Git?

Após a instalação do Git pode-se realizar alguns procedimentos para customizar seu ambiente, como definir seu nome de usuário e e-mail, isso é importante porque todo comando commit irá usar essas informações. O Git vem com uma ferramenta chamada git config que permite ao usuário obter e configurar informações, essas informações são salvas no arquivo .gitconfig localizado dentro do diretório do usuário do sistema operacional (SO), e você pode alterá-las

a qualquer momento.

Para definir seu nome de usuário e email, você utilizará os seguintes comandos no terminal do seu SO ou no bash do próprio Git:

```
$ git config --global user.name  
"seu nome"  
$ git config --global user.email  
"seu endereço de email"
```

Além do nome de usuário e e-mail, você pode definir qual é o editor de texto padrão do Git para caso seja necessário editar alguma mensagem ou códigos, executando o seguinte comando:

```
$ git config core.editor  
comando_do_seu_editor
```

Por exemplo, quem utiliza o editor Vim, o comando será:

```
$ git config core.editor vim
```

Você utilizará o comando do editor para o parâmetro.

#### 2. Checando valores

Eventualmente pode ocorrer de o usuário querer consultar os dados que

foram registrados. Para obter tais informações, iremos usar novamente a ferramenta git config, de forma bem simples, após o git config você usará uma chave, que é o comando que você utilizou para registrar a informação que você quer obter, por exemplo, para obter o nome de usuário a chave utilizada irá ser 'user.name', o comando ficará da seguinte forma:

```
$ git config <chave>
$ git config user.name
```

Da mesma forma podendo ser utilizado com 'user.email':

```
$ git config user.email
```

E se desejar obter todos os dados registrados no seu git, basta utilizar o seguinte parâmetro '- -list', executando da seguinte forma:

```
$ git config - -list
```

ele irá listar todos os dados existentes referentes ao seu git.

### 3. Repositório local

Se você deseja utilizar o Git para um projeto, o primeiro comando que deve ser utilizado é o git init, da seguinte forma:

```
$ git init <nome do projeto>
```

Esse comando irá criar uma pasta oculta chamada .git dentro do diretório principal do seu projeto que irá referenciar aquele repositório como sendo um repositório Git.

Após fazer alterações no seu arquivo, você tem que utilizar o comando git add para salvar essa alteração no histórico de versões do seu arquivo, esse comando irá colocar esse arquivo na área de preparação, há duas formas de fazer isso com os seguintes comandos:

```
$ git add <nome do projeto>
```

Esse comando irá preparar somente o arquivo indicado, se você deseja preparar todos os arquivos alterados daquele repositório basta utilizar o ponto após o 'add', assim:

```
$ git add .
```

Para realizar a transição do seu arquivo, ou seja, para finalmente adicioná-lo ao histórico de versões você deve realizar um commit descrevendo na mensagem todas as alterações feitas no seu arquivo, dessa maneira:

```
$ git commit -m "mensagem"
```

#### 4. Repositório remoto

Repositórios remotos são versões do seu arquivo que estão hospedados em algum local na Internet. Repositórios remotos são utilizados quando desenvolvedores precisam compartilhar o projeto com outros membros da equipe. Para que esse compartilhamento seja possível, o repositório tem que ser hospedado em uma hospedagem de repositórios Git. Por exemplo, o GitHub.

O GitHub é um serviço web que oferece diversas funcionalidades, dentre elas, hospedar projetos pessoais gratuitamente. Além de hospedar, você poderá contribuir com projetos de outros usuários. A maioria dos projetos open source estão hospedados no GitHub, permitindo aos usuários informar bugs, ver novas versões, e até enviar códigos e correções.

Para adicionar repositórios remotos no Git, basta utilizar esse comando:

```
$ git remote add <nome do repositório> <url>
```

É uma boa prática o nome do repositório ser pequeno, pois irá facilitar quando for se referir ao repositório.

Se houver uma necessidade ou desejo de trabalhar em projetos remotos,

colaborativos ou simplesmente ajudar outros usuários nos seus projetos, há um jeito de clonar o repositório com seus histórico de versões para o seu computador local, com o comando:

```
$ git clone <url do projeto>
```

Se o projeto está sendo desenvolvido remotamente por várias pessoas, sempre haverá modificações nos arquivos, para não baixar o projeto todo, há uma forma de baixar somente as alterações feitas, assim:

```
$ git pull
```

Usar o Git se torna essencial para qualquer pessoa que deseja desenvolver software em alto nível, pois não terá preocupação em relação ao controle das versões, e o histórico de versionamento. Há vários outros comandos que podem ser utilizados no Git, para outras finalidades ou até mesmo para as repassadas, os comando no presente artigo irão auxiliar qualquer um que deseja começar a utilizar essa ferramenta, de forma simples e objetiva.

# red{info}

---

 fb.com/petccuern

 @petccuern

 petcc.uern.br

 di.uern@gmail.com

 (84) 3315-2225

 di.uern.br



**Departamento**  
de Informática



Programa de  
Educação Tutorial  
Ciência da Computação