



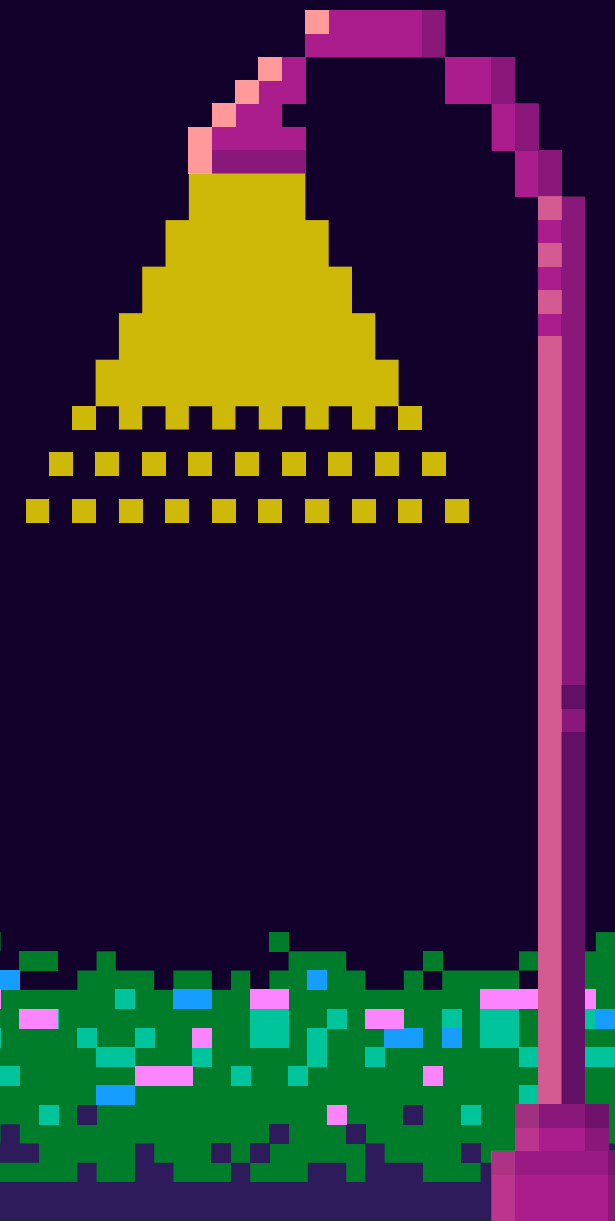
HELLO
WORLD



INTRODUÇÃO AO FRONTEND

ESTRUTURA DE UMA APLICAÇÃO WEB

A aplicação web diz respeito a uma solução que é executada diretamente no browser (ou navegador), não sendo preciso realizar uma instalação na máquina do usuário. Pode-se, também, utilizar como definição “tudo aquilo que é processado em um servidor terceiro”. As plataformas de e-commerce e as redes sociais são alguns dos exemplos que se enquadram nesse perfil.



Como funciona uma aplicação web?

Para que uma aplicação web funcione, ela depende de um servidor web, de solicitações realizadas pelos usuários, do uso de protocolos e métodos (normalmente o HTTPS) e da resposta do protocolo.

A aplicação deve permitir que as pessoas usuárias consigam fazer uma solicitação e receber algo em resposta. Ou seja, elas precisam mediar essa interação de forma natural, devolvendo o que a pessoa deseja como resultado — por exemplo, se a pessoa pede para abrir uma foto, é preciso que isso seja devolvido à ela, e não a abertura de uma página aleatória.

Nesse sentido, o servidor tem por função receber a solicitação do público e devolver uma resposta para a aplicação. A resposta pode ser a abertura de uma nova página, imagens, documentos, entre outros.

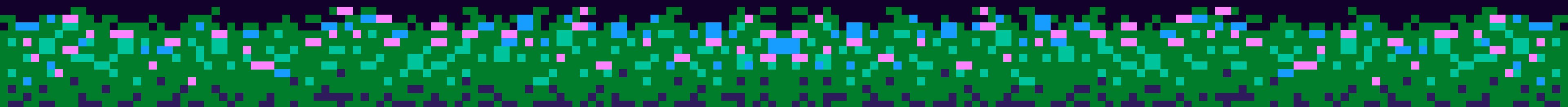
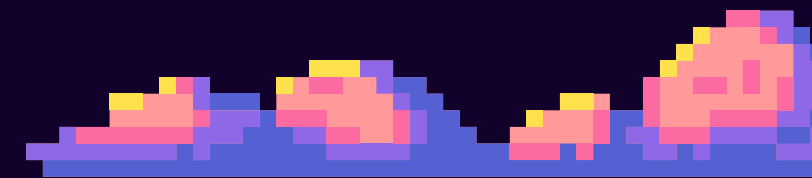
Outro componente da aplicação web é o protocolo HTTP. Ele é como uma espécie de “linguagem” que determina o padrão pelo qual a solicitação realizada se comunica com o servidor. Outros protocolos envolvidos em aplicações web podem ser:

FTP (transmissão de arquivos);

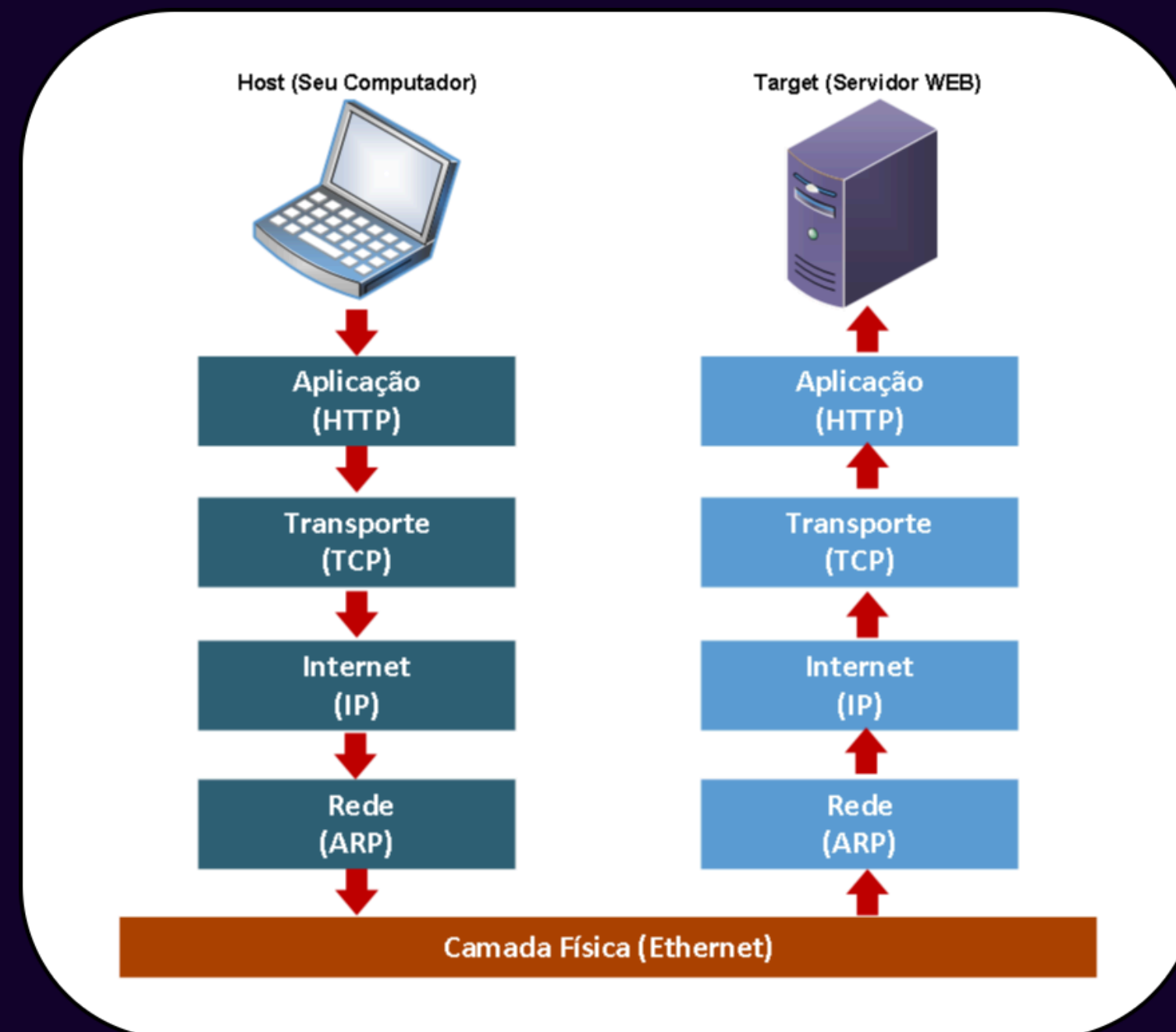
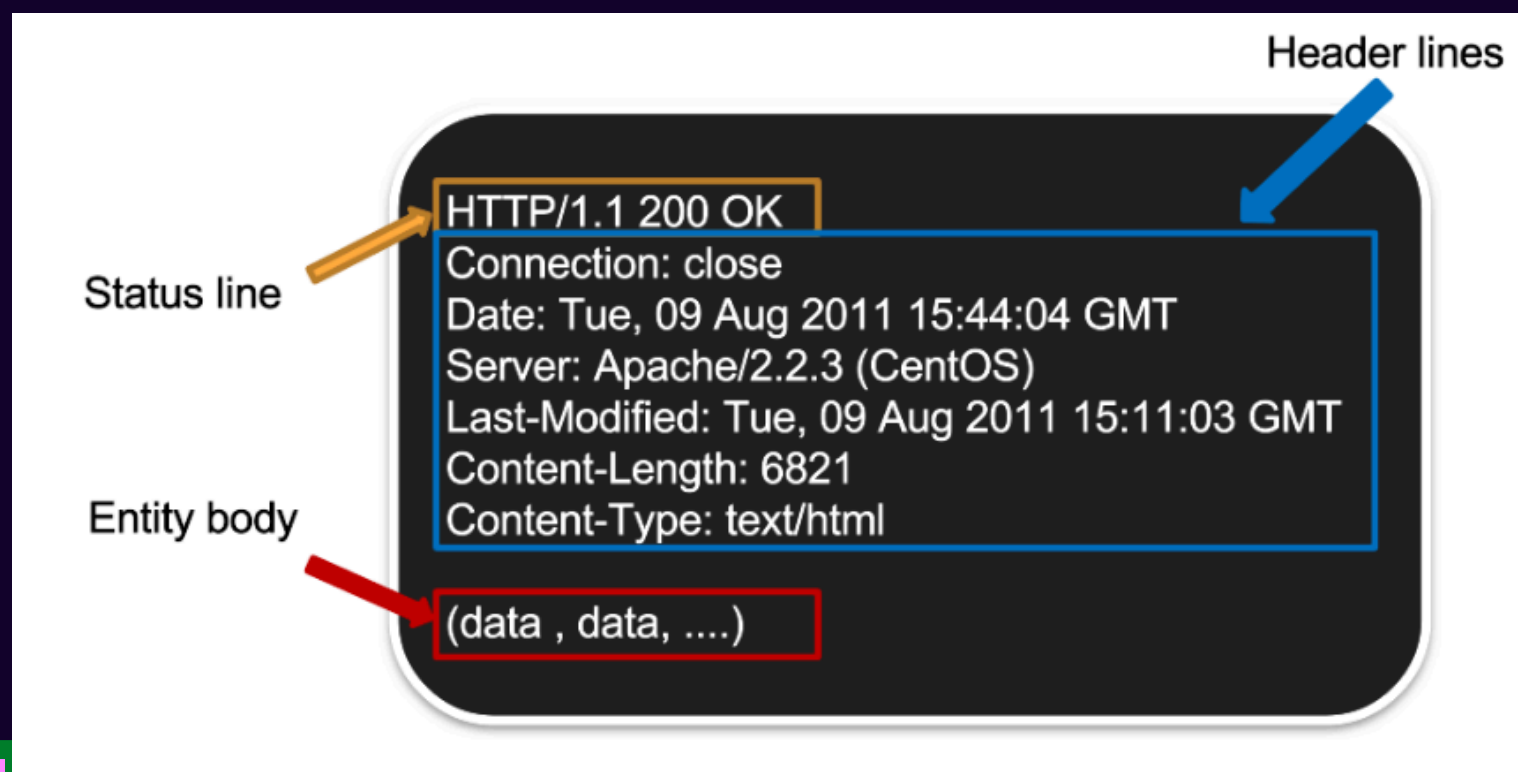
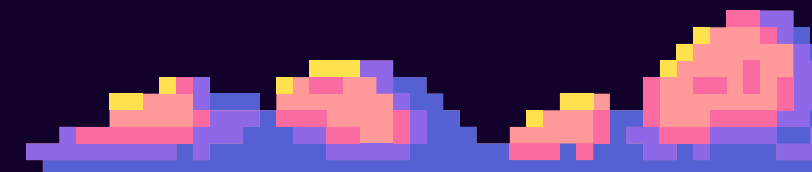
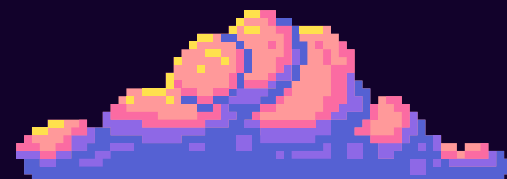
SMTP (envio de mensagens para um servidor de e-mail);

POP e IMAP (acesso a mensagens de e-mail eletrônico).

Além disso, todas as atualizações necessárias são feitas por meio de um único servidor central, não sendo preciso baixar aplicações ou realizar reinstalações. Logo, temos tudo feito de forma centralizada, sem maiores problemas, bastando realizar a adaptação no servidor.

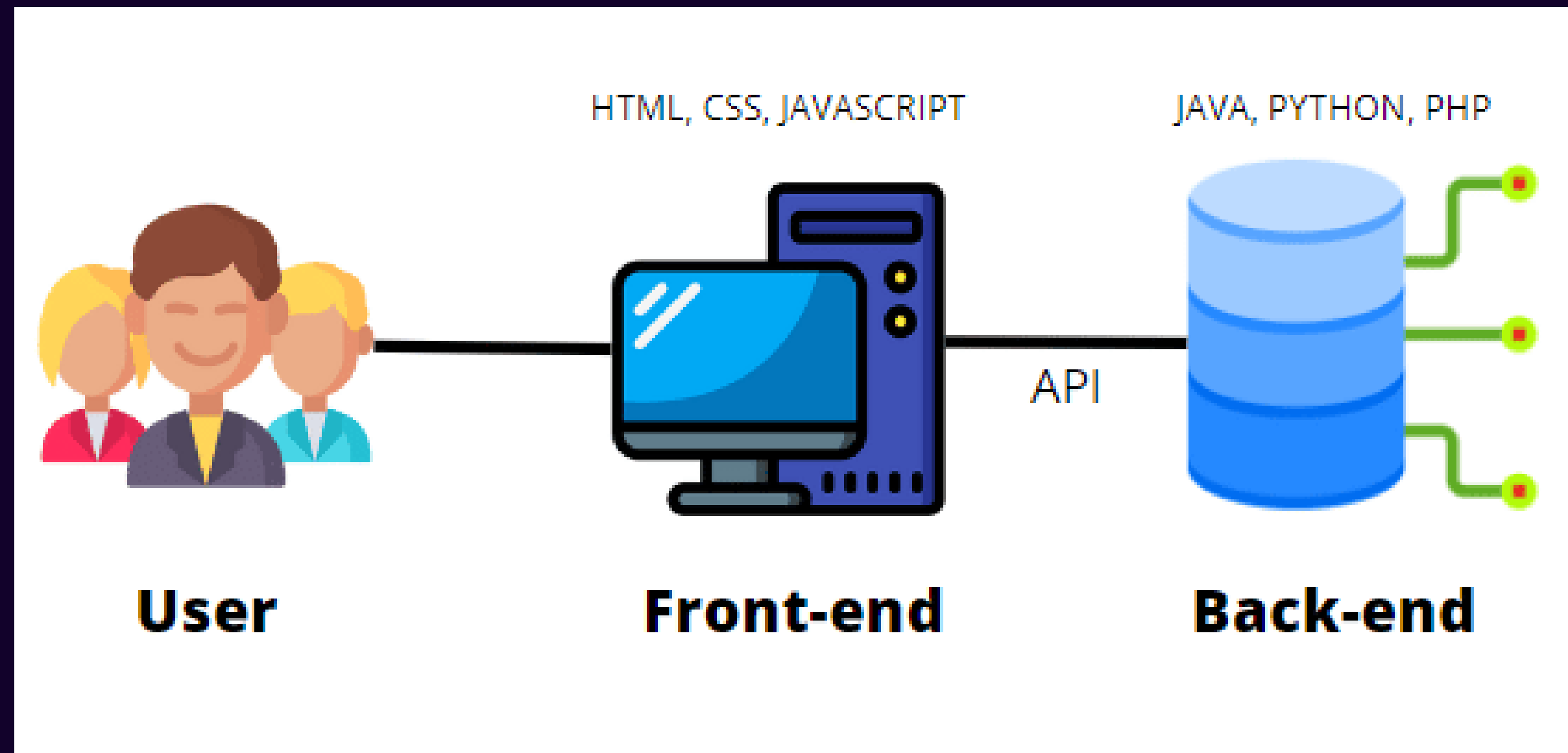


Protocolo HTTP / HTTPS:



Quais as principais partes da estrutura de uma aplicação Web?

Enxergar uma aplicação web pronta, testada e funcionando é muito bom, não é mesmo? Pois bem, mas temos que entender que boa parte das aplicações depende diretamente de um banco de dados, de um sistema de middleware na web, entre outros componentes e ferramentas. Assim, caso você enfrente algum problema em sua aplicação, terá mais facilidade para contorná-lo por conhecer a arquitetura. As principais partes que compõem uma aplicação web são:



Front-end

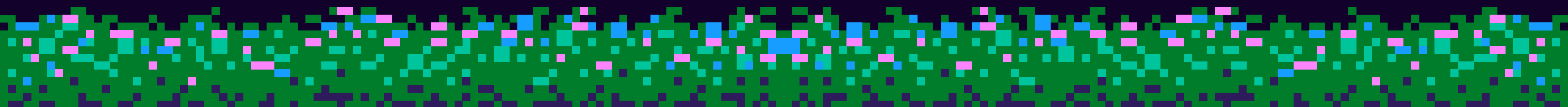
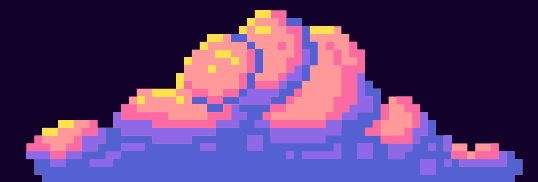
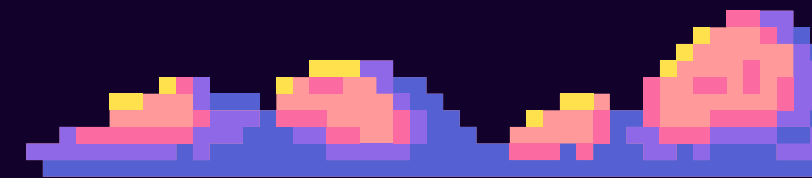
O front-end nada mais é que toda a parte visual de um site. Através dos códigos, uma interface é construída (de preferência, bastante amigável). As tecnologias mais utilizadas para a construção de um site são o HTML e CSS, bem como o Javascript para implementação de algumas funcionalidades.

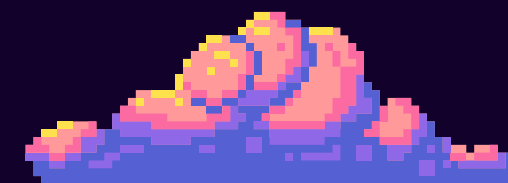
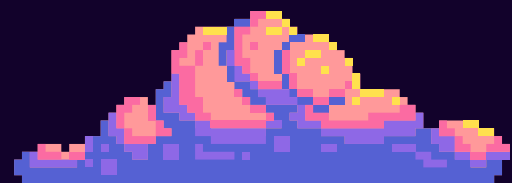
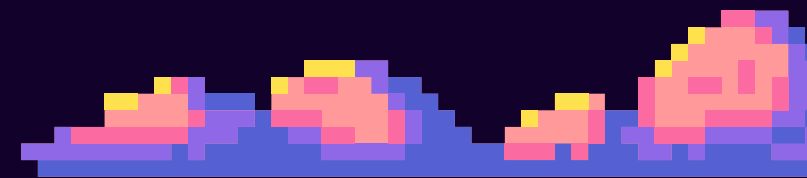
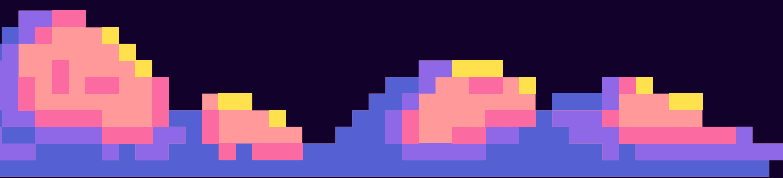
Back-end

O back-end, como o próprio nome já diz, é tudo aquilo que está por trás de uma aplicação. Toda essa construção da base da aplicação pode ser desenvolvida com o uso de linguagens de programação, como o Java, Ruby, C#, Python, PHP, entre outras. Além disso, é através do back-end que ocorre a conexão entre os bancos de dados e o carregamento de informações.

API

A API é um conjunto de normas que possibilita a comunicação entre diversas plataformas. Por sua vez, as pessoas desenvolvedoras costumam utilizar as APIs para comunicar funcionalidades diferentes com suas aplicações web.





TIPOS DE CODIFICAÇÃO

O que é Codificação de Texto?

A codificação de texto é um processo fundamental na área da tecnologia da informação, que consiste em transformar caracteres e símbolos em uma sequência de bits compreensível para os computadores. Essa codificação é necessária para que os dispositivos eletrônicos possam processar, armazenar e transmitir informações de maneira eficiente.

ASCII

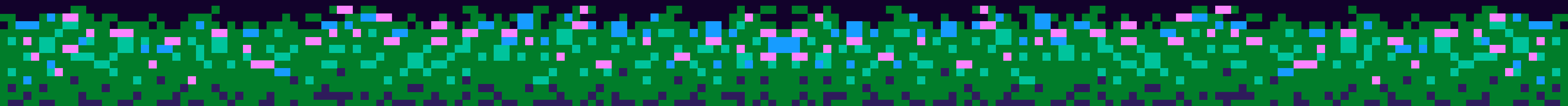
O ASCII é um dos padrões mais antigos e amplamente utilizados para codificação de texto. Ele utiliza 7 bits para representar os caracteres, o que permite a representação de 128 caracteres diferentes. No entanto, o ASCII é limitado a caracteres da língua inglesa e não é capaz de representar caracteres acentuados ou de outros idiomas.

Unicode

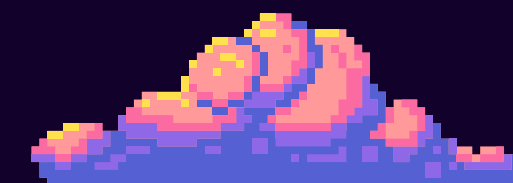
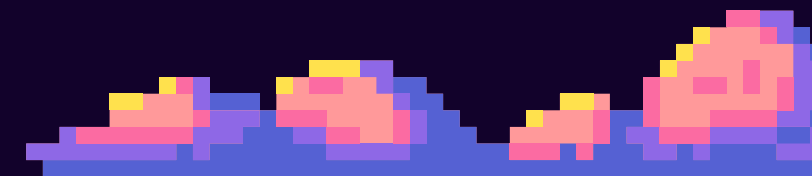
O Unicode é um padrão mais recente e abrangente, que permite a representação de uma ampla variedade de caracteres de diferentes idiomas e símbolos especiais. Ele utiliza 16 bits para representar os caracteres, o que permite a representação de até 65.536 caracteres diferentes.

UTF-8

O UTF-8 é uma codificação de texto baseada no Unicode, que utiliza uma sequência variável de bytes para representar os caracteres. Ele é capaz de representar todos os caracteres do Unicode, ao mesmo tempo em que mantém a compatibilidade com o ASCII. O UTF-8 é amplamente utilizado na web e é o padrão recomendado para páginas da web.



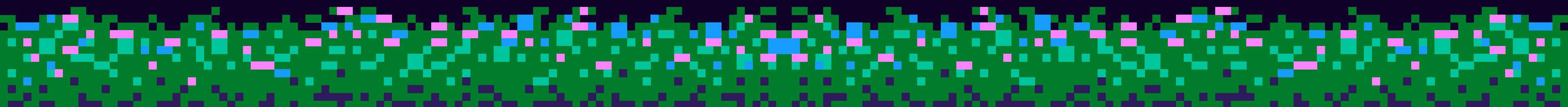
QUAIS OS 10 PRINCIPAIS TIPOS DE WEBAPPS?



1. Aplicativos web estáticos
2. Aplicativos web dinâmicos
3. Lojas online
4. Aplicativos de página única
5. Aplicativo com diversas páginas
6. Aplicativos web animados
7. Aplicações para gerenciamento de conteúdo
8. Aplicações de portais
9. Progressive Web App
10. Aplicativos de plataforma

Exemplos de webapps:

1. Twitter
2. Instagram
3. Pinterest
4. Uber
5. Ifood



FRONTEND VS BACKEND



Um pouco da história de cada uma

Tente imaginar como eram alguns anos atrás, em que as empresas que desejavam compartilhar documentos entre si, ou entre seus próprios funcionários não tinham a possibilidade de fazer isso de forma rápida e prática como hoje, em que com apenas com um clique conseguimos fazer todo esse transporte. Para eles, era necessário utilizar de outras ferramentas que demandam muito mais tempo. Sendo assim, em 1991, Tim Berners-Lee teve uma grande ideia que não apenas facilitaria o dia a dia de empresas, mas mudaria o mundo: ele projetou o HTML.

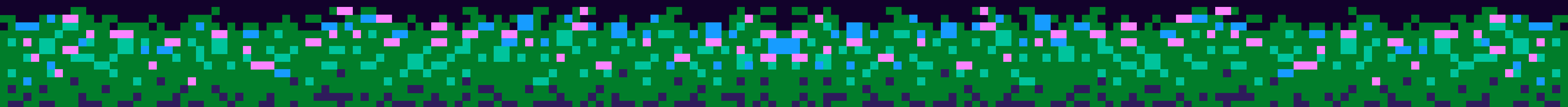
Essa linguagem de marcação, portanto, havia sido desenvolvida para possibilitar o compartilhamento de documentos de forma mais prática. Porém, a partir do ano seguinte, foi criada a World Wide Web (WWW), uma rede de alcance mundial, tornando o HTML uma ferramenta utilizada para tudo e por todos.

Já o CSS surgiu como consequência desse grande crescimento do HTML. Quanto mais o HTML era utilizado, mais os desenvolvedores usavam da criatividade para deixar as páginas bonitas e estilosas. Porém, toda essa parte estética era escrita no mesmo arquivo da parte estrutural, deixando-os gigantes e cada vez mais difíceis de entender. Sendo assim, em 1995, foi criado o CSS para tomar conta da parte estética da página, de forma separada da estrutura, papel do HTML.

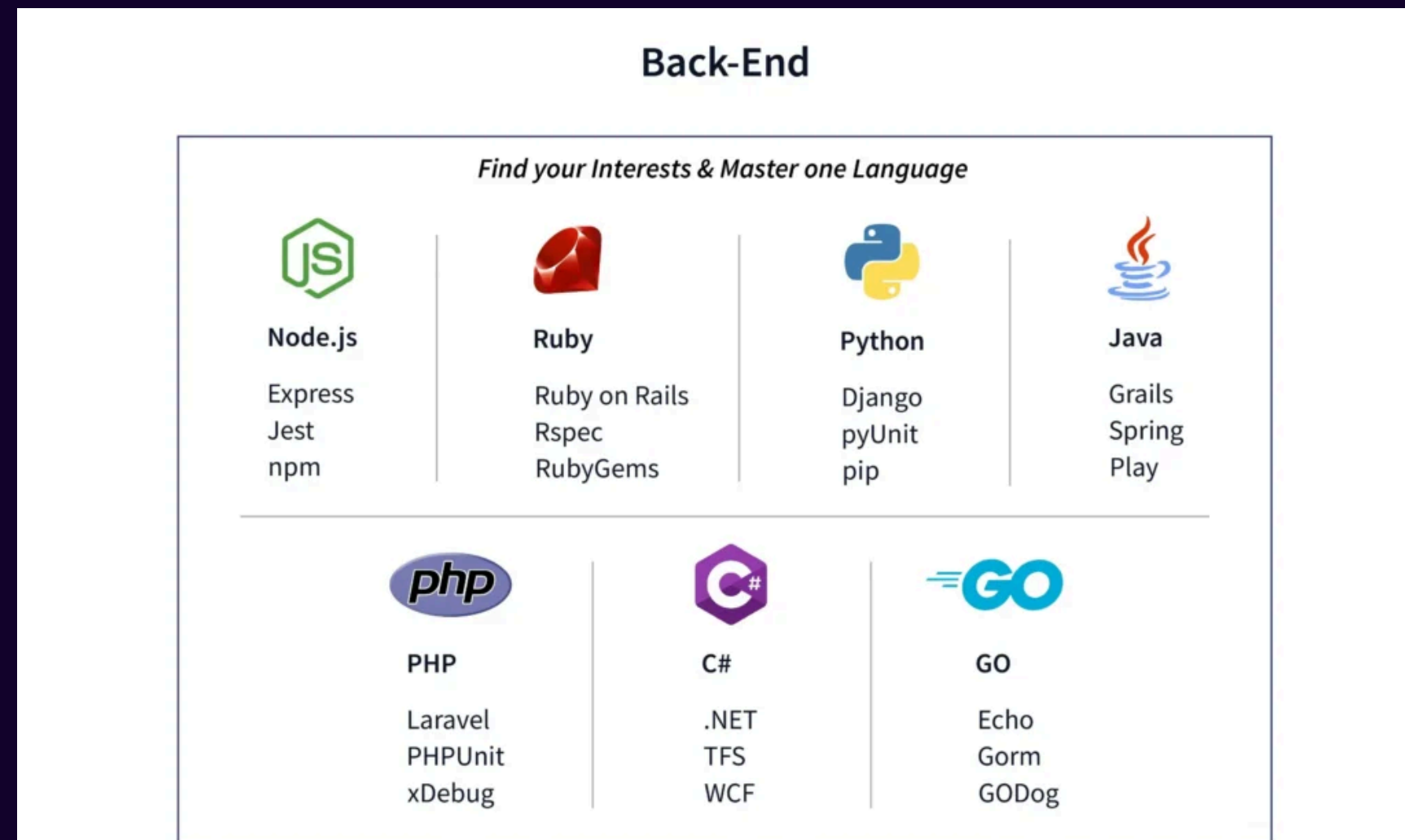
Por fim, o JavaScript foi criado a partir de uma grande competição entre gigantes do mercado na época: Microsoft e Netscape.

Tudo começou com a criação de uma linguagem de servidor pela Netscape para ser implantada nos servidores web da empresa. Quando a Microsoft descobriu que essa ideia poderia ter sucesso, não perdeu tempo e começou a desenvolver a linguagem JScript para implementar nos seus próprios servidores web ISS. Percebendo que sua ideia estava sendo copiada, a Netscape decidiu desenvolver uma tecnologia do lado do cliente, que rodasse no próprio navegador. Ela acabou tornando tudo mais rápido. Por isso, a Microsoft novamente percebeu uma grande oportunidade de mercado, e resolveu desenvolver um sistema semelhante ao da empresa concorrente.

Existiam então duas tecnologias “client-side”: JScript da Microsoft e Javascript da Netscape, cada uma funcionando apenas nos navegadores das respectivas empresas. Para padronizar o Javascript, a Netscape não perdeu tempo e o enviou para a empresa ECMA, permitindo sua incorporação à maioria dos navegadores existentes. A partir daí surgiu o ECMAScript, que nada mais era do que o novo nome para o Javascript, mesmo até hoje não sendo muito utilizado



O QUE É BACKEND?



O QUE É FRONTEND?



COMO SEPARAR AS FUNÇÕES?



HTML



CSS



Javascript

HTML (HyperText Markup Language)

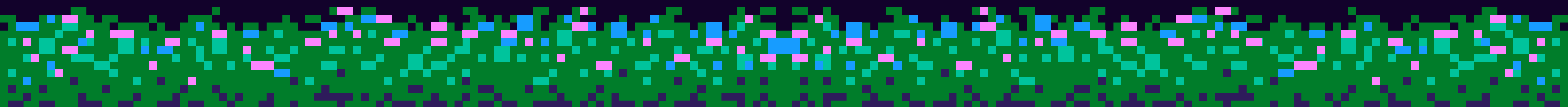
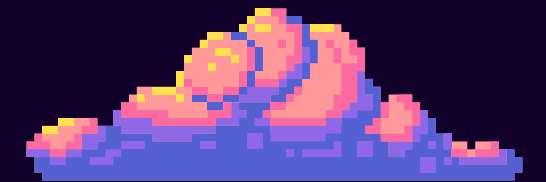
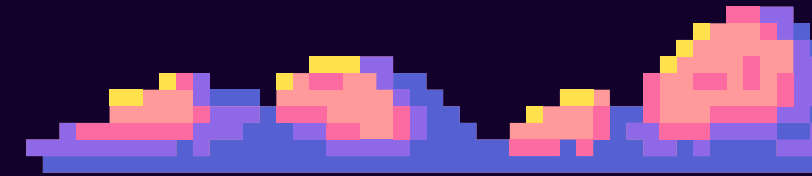
- O que é: O HTML é a espinha dorsal de qualquer página web. Ele é usado para estruturar e dar significado ao conteúdo web, como definir parágrafos, cabeçalhos, listas, links e elementos de mídia (imagens, vídeos, etc.).
- Função: Serve como o esqueleto da página, determinando a estrutura e o conteúdo. Sem HTML, não haveria uma forma de organizar o texto ou os elementos numa página web.

CSS (Cascading Style Sheets)

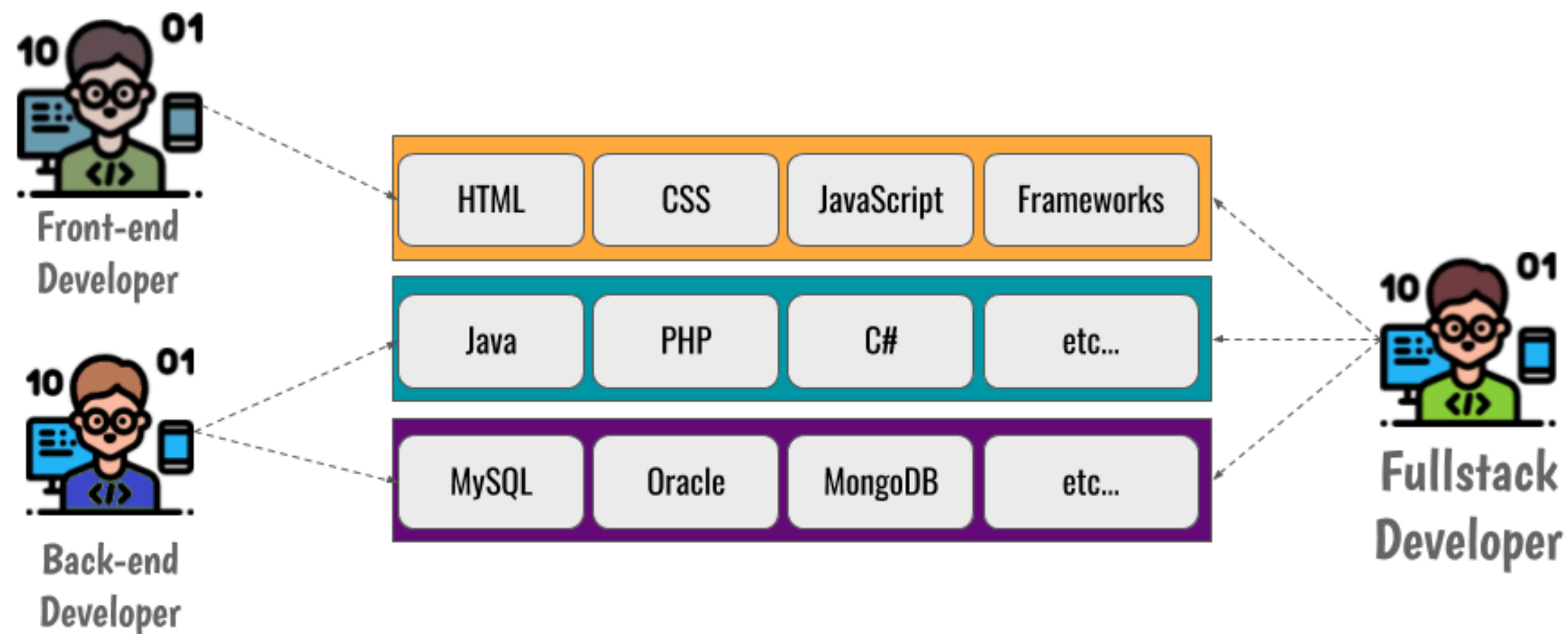
- O que é: CSS é usado para controlar a apresentação, o layout e o design de páginas web. Ele permite definir cores, fontes, espaçamentos, posicionamento de elementos e muito mais, sem alterar o conteúdo ou a estrutura fornecida pelo HTML.
- Função: Proporciona estilo ao HTML. Com CSS, você pode transformar um documento HTML bruto em uma página visualmente atraente, melhorando a experiência do usuário.

JavaScript

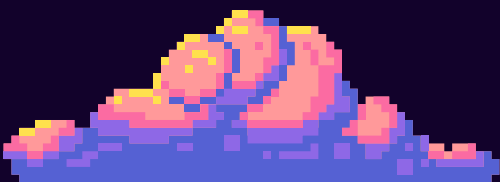
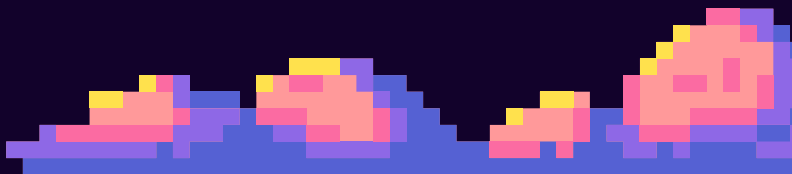
- O que é: JavaScript é uma linguagem de programação que permite adicionar interatividade e dinamismo às páginas web. Isso inclui coisas como responder a eventos de clique, enviar dados para um servidor sem recarregar a página, criar animações e muito mais.
- Função: Torna as páginas web interativas e reativas. Com JavaScript, você pode modificar tanto o HTML quanto o CSS para atualizar a interface do usuário em resposta às ações do usuário, dados recebidos de servidores externos, e outros eventos.



0 programador fullstack

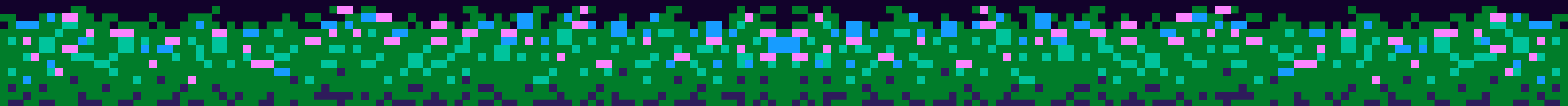


Pesquisa realizada em março de 2024



Salários de desenvolvedores no Brasil

Cargo	Salário médio
Desenvolvedor Full Stack	R\$5.000 - R\$14.000
Desenvolvedor Back-end	R\$3.700 - R\$13.000
Desenvolvedor Front-end	R\$3.500 - R\$12.000



NOSSOS CONTATOS:



27 99500-7495



<https://beacons.ai/prismatech>



producaoprismatech@gmail.com



Avenida Jerônimo Monteiro 145, Vitória



