



MANUAL DO



TECH RECRUITER



- Linguagens de programação, bibliotecas e frameworks;
- Áreas de atuação na tecnologia;
- Organização de squads, metodologia ágil;
- Boas práticas de recrutamento e seleção;
- Dicionário técnico na programação.

SUMÁRIO

1. Tecnologia na atualidade;
2. Áreas de atuação na programação;
3. Principais linguagens de programação;
4. Frameworks;
5. Organização de equipes e metodologias;
6. Boas práticas de recrutamento e seleção;
7. Dicionário técnico na programação.





TECNOLOGIA NA ATUALIDADE

Todo profissional precisa entender a fundo qual é o foco da empresa que ele está inserido, seja no varejo, financeiro ou no caso da Escalate: tecnologia. Esse foco trás uma direção de qual será o público que o RH precisa voltar seu olhos, ações para o melhor desenvolvimento do negócio.

Nos últimos anos a tecnologia nunca esteve tão presente quanto agora, a criação e transformação de ambientes para o digital tende a aumentar.

Por isso, o mercado tecnologia se mantém aquecido, cheio de oportunidades. Todos os dias algum profissional de tecnologia é abordado por diversas empresas, muitas delas ainda conduzem um processo seletivo de forma padrão, resultando muita das vezes um declínio do candidato, falta interesse ou aceite de outras oportunidades.

O objetivo da Escalate é formar recrutadores que entendam a necessidade do cliente, na busca de profissionais com boas habilidades. Para isso precisamos entender como esses profissionais se organizam, como se classificam e principalmente qual ferramenta eles dominam. Assim é possível estabelecer diálogos fluidos e enriquecedores com a equipe interna e com futuros candidatos.



A busca por profissionais de tecnologia está cada vez mais intensa devido a demanda nas áreas de criação de sites, aplicativos, softwares e banco de dados.



Vamos mergulhar no vocabulário de profissionais de programação para compreender os termos mais cotidianos, entender como o time e tarefas são divididas, boas práticas de recrutamento e acompanhamento do profissional durante o processo seletivo, ao final vamos garantir ótimos talentos para a Escalate.

Bora desenrolar esses fios?



ÁREAS DE ATUAÇÃO NA PROGRAMAÇÃO_



FRONT-END



O front-end é o campo ligado à interface, estrutura e apresentação, isto é, aquilo que terá interação direta com o usuário. O talento que ocupará esta função terá um trabalho de excelência se tiver conhecimentos sobre UX (Experiência do usuário).

BACK-END



O back-end diz respeito ao que está por trás da aplicação. Estabelecem o que chamamos de regras de negócio, conectando dados e sistemas para determinar uma funcionalidade.

FULL-STACK



Atuação do back-end ao front-end, dominando todas as etapas do processo. Por ter conhecimento avançado, costuma obter as melhores oportunidades.

DATA-SCIENCE



Cientistas de dados são fundamentais para empresas que buscam qualificar resultados e entender o comportamento de clientes. Essa é uma posição estratégica e analítica, que exige conhecimento técnico avançado.

MOBILE



O desenvolvimento mobile é a atividade voltada à programação para dispositivos móveis. Com o smartphone cada vez mais presente na rotina das pessoas, a carreira de DEV Mobile ganha muita relevância.

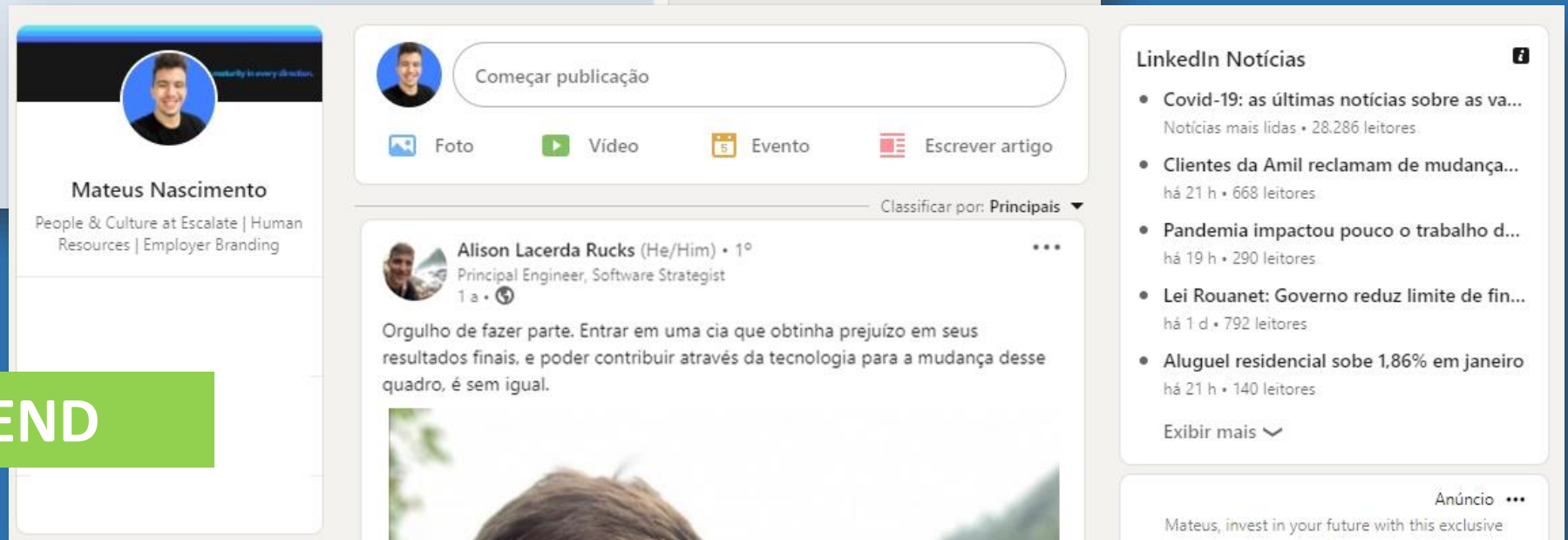


Para melhor entendimento...



FRONT-END

BACK-END



PRINCIPAIS LINGUAGENS DE PROGRAMAÇÃO



A **programação** é um **processo de escrita**. Linguagem são formas diferentes de redigir scripts que orientam computadores, smartphones, smart TVs, etc. Novidades saem todos os dias quando o assunto é tecnologia da informação. Isso se reflete na linguagem programação continuamente atualizada e testada em novos modelos com frequência. Por isso a necessidade de buscar e se informar sempre que possível. Independente das transformações que ocorrem, o mercado conta com linguagens bem consolidadas, que devem permanecer no radar de cada Tech Recruiter.





JavaScript

Permite adicionar interatividade às suas páginas. Exemplos comuns que você pode ter visto nos sites são controles deslizantes, interações de clique, pop-ups e assim por diante. Além de ser usado no **Front-end** em navegadores, existe o Node.js, que é um ambiente de tempo de execução JavaScript de **Back-end**, multiplataforma e de código aberto que é executado no mecanismo V8 e executa código JavaScript fora de um navegador da web. Além disso, esse fator pode representar ganhos de reutilização de código e criação de equipes multidisciplinares, com melhor aproveitamento de recursos. Após um ano de seu lançamento, a Microsoft utilizou a linguagem para seu navegador, a alçando a uma das tecnologias mais importantes e utilizadas.

Recursos: Desenvolvimento rápido, Menos sobrecarga de script, corte de custos, código aberto, controle de entrada e saída de dados produzidos.

Leia mais sobre JavaScript.



JAVA

Em segundo lugar entre as tecnologias de desenvolvimento de **Back-end** hoje. Surgiu como a plataforma de escolha para desenvolvedores que preferem criar aplicativos da web com recursos personalizados e inovadores. Java também é útil para desenvolvimento móvel e aplicativos em dispositivos incorporados.

Recursos: Escalável e direto, capacidade executar várias tarefas simultaneamente, Bibliotecas de código aberto massivas, Segurança aprimorada



NODE JS



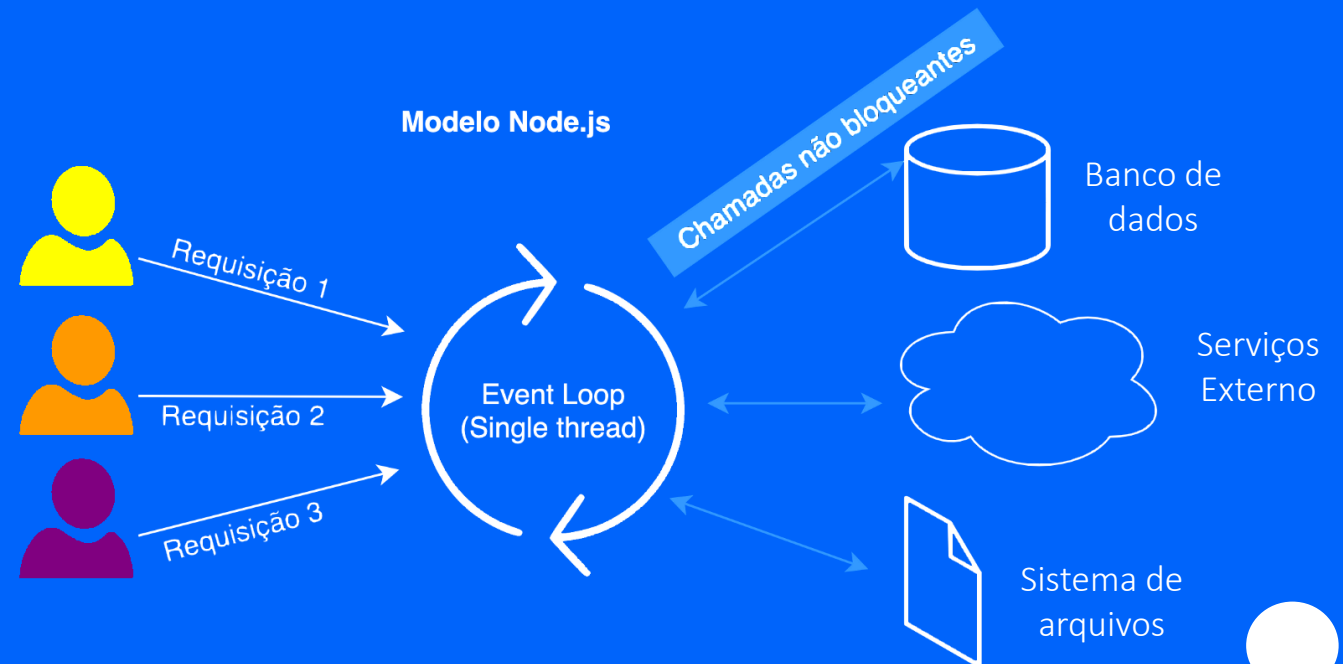
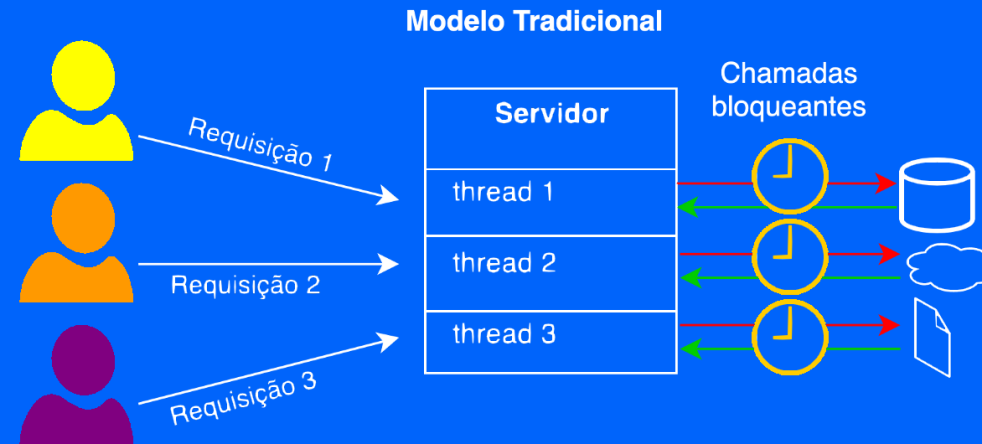
Biblioteca JavaScript, ambiente de execução que ganhou espaço por trazer a linguagem **JavaScript para o Back-end**. É ideal para quem busca criar serviços web e APIs no estilo REST.

Significa que com o Node.js é possível criar aplicações JavaScript para rodar como uma aplicação independente e autossuficiente em uma máquina, não dependendo de um browser para a execução.

Apenas uma thread é responsável por executar o código JavaScript da aplicação. Essa thread é chamada de Event Loop, e leva esse nome pois cada requisição é tratada como um evento. O Event Loop fica em execução esperando novos eventos para tratar, e para cada requisição, um novo evento é criado. Enquanto que nas outras linguagens, a execução é multi-thread, ou seja, uma nova thread é criada a cada requisição e quando o limite desse recurso é atingido, novas requisições terão que esperar para serem tratadas.

Recursos: Flexibilidade, leveza, produtividade da equipe, aplicações em tempo real.

Ent-enda como funciona uma requisição:





PHP

Linguagem Back-end, muito popular que vem ganhando força no mercado ao ser utilizada em sistemas e serviços web. Entre esses sites, se destacam Facebook, Twitter, Wikipédia e gigantes chineses, como Baidu e QQ.

A linguagem PHP conquistou o mercado e os seus desenvolvedores, tendo uma comunidade vibrante e uma evolução rápida nos últimos anos

Recursos: Código aberto e versátil, Econômico, fácil de usar, bom para iniciantes, funções de automação, segurança embutida.



PYTHON

Linguagem Back-end que tem ganhado popularidade na área de tecnologia, dado a sua característica de exigir poucas linhas de código e permitir uma leitura fácil, além de ter várias bibliotecas do mundo de **Data Science, Machine Learning, desenvolvimento web, automação e jogos**. Permite ao programador desenvolver aplicativos sofisticados e está presente em aplicativos como Pinterest e Instagram.

Recursos: Relativamente fácil de aprender, enormes bibliotecas, eficiente em termos de custos, recursos de iot, códigos incorporáveis.





C ou C++

Linguagem Back-end, uma das mais utilizadas até os dias de hoje, a C teve influência direta na criação do C++, Java, Python, JavaScript. São consideradas as melhores portas de entrada para o aprendizado de programação, pois oferecem insumos que facilitam a compreensão de diversos outros tipos.

Como são linguagens nativas, permitem estabelecer uma boa comunicação com os sistemas e seus componentes. Não à toa, têm sido muito aplicadas para o desenvolvimento de jogos, por exemplo.

Recursos: Funciona em qualquer plataforma, programação de objetos, bom gerenciamento de memória, usado para programar a maioria dos recursos de hardware.



C#

Uma linguagem de programação que funciona dentro da plataforma da .NET.

Ele faz um excelente trabalho na automação de codificação em servidores Windows e plataformas web (usando o framework ASP.Net.) C-Sharp, que é mais ou menos uma extensão do C ++, já existe há muito tempo.

Recursos: Execução em vários sistemas operacionais, orientada a objetos, bom funcionamento enquanto os recursos para eliminação de resíduos do sistema está sendo executado.





RUBY

A linguagem de programação que compartilha sintaxe semelhante com Python, Java, também surgiu como uma das tecnologias de **Back-end** líderes hoje. É uma plataforma popular para o desenvolvimento de aplicativos da web e é preferida pela maioria dos desenvolvedores para prototipagem, o processo de transferência de qualquer objetivo físico ou virtual para a realidade.

Recursos: Alta legibilidade de código e muitas bibliotecas de terceiro, permite-se adquirir dados de outro programa e se modifique durante a execução, grandes bibliotecas e recursos de teste e automatização.



Bora falar sobre uma biblioteca?



Não estamos falando de livros em cima de uma prateleira, mas sim de um biblioteca de programação.

São pedaços de códigos (como funções) já programados e que podem ser usados em outros códigos. A biblioteca é uma extensão de fora do contexto, ou seja, ela pode ser usada e desplugada de um contexto para outro. As bibliotecas são consumidas pela aplicação principal. Sendo assim, elas já formam testadas e referendadas, sem que você precise criar algo do zero. A sua principal função é facilitar a programação, garantindo mais agilidade e menos erros.

Se você está usando a linguagem JavaScript e precisa saber o número de dias entre uma data e outra, você pode aplicar uma biblioteca. Isso porque a linguagem, sozinha, não tem essa funcionalidade.

Dessa forma, você precisa criar uma função com esse comando. Além disso, se precisar incrementar essa pesquisa, com horas, minutos ou uma data intermediária, será possível executá-la com o auxílio de uma biblioteca. Assim, usa-se uma coleção de funções direcionadas ao cálculo do tempo.

Em suma, a biblioteca funciona a partir de uma coleção de códigos. O objetivo é resolver um determinado tipo de problema.



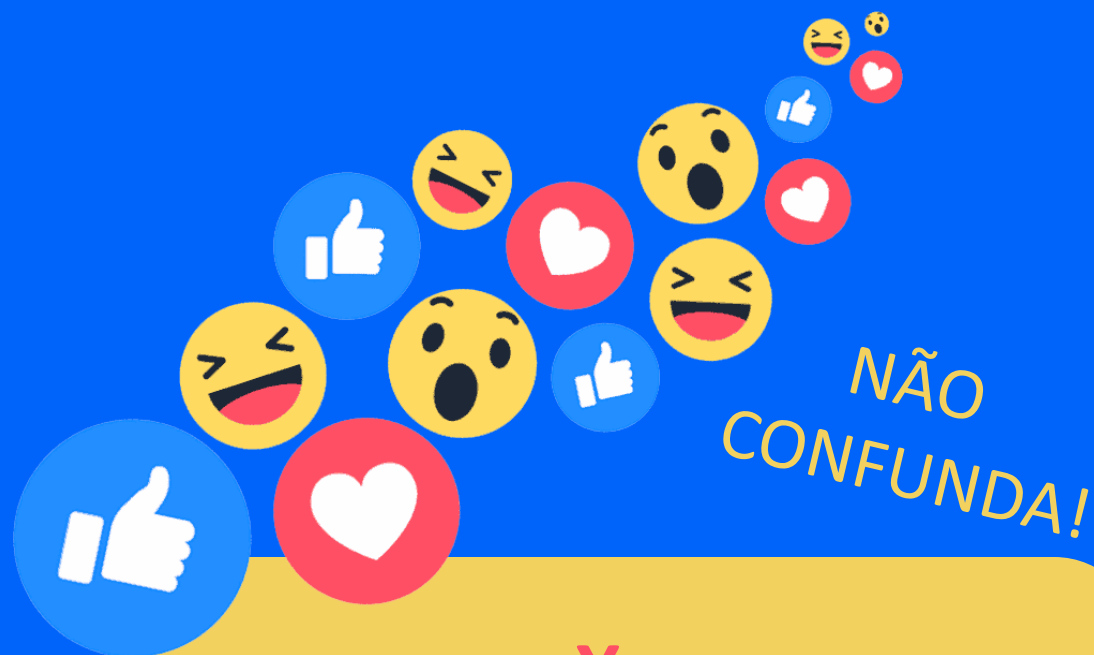


REACT

Uma biblioteca de JavaScript, desenvolvida para o **Front-end**. Criado pelo Facebook utilizada para construir interfaces reutilizáveis, nomeadas para aplicações de página única.

O diferencial do React é ser rápida, escalável e simples, podendo ser usada com outras bibliotecas ou frameworks de JavaScript, como o Angular JS. Essa amplitude de funcionalidades e sua forma descomplicada de realizar processos são os responsáveis pela ferramenta ser tão buscada pelos programadores.

Assim, possibilita aos desenvolvedores criarem aplicações web, tendo a facilidade de alterar elementos ou os dados exibidos, sem recarregar a página. Por exemplo, o número de likes de uma rede social pode aumentar ou diminuir sem a necessidade de realizar o refresh na página.



NÃO
CONFUNDA!

REACT.JS X REACT NATIVE

O React Native funciona como um **framework** híbrido, que permite que o desenvolvedor programe em um só código, que pode ser aproveitado para Android e iOS, facilitando recursos de JavaScript.

MAS O QUE É UM FRAMEWORK?



Bora falar sobre framework?



Para entender de forma simples, vamos pensar que é uma espécie de template com diversas funcionalidades que podem ser utilizados em seus projetos. Ele conta com **ferramentas, sistemas, componentes e guias** que agilizam o processo de criação de soluções, sendo, portanto, um artifício essencial na vida de um profissional de TI. Eles também ajudam no desenvolvimento rápido e seguro de aplicações. Suas funcionalidades permitem atender qualquer necessidade do profissional, desde que o framework selecionado seja direcionado a função desejada.

Dentro deles existem várias bibliotecas. Veja um exemplo a seguir:

- Você está utilizando a biblioteca do JQuery durante o desenvolvimento do seu site;
- Se, em algum momento, resolver que ela não está mais sendo útil, poderá retirá-la da aplicação;
- É necessário, apenas, refatorar o código para retirar de onde ela tinha sido utilizada anteriormente.
- Se você decidir desenvolver o seu site utilizando um framework, não poderá abrir mão do seu uso no meio do caminho, apenas se reiniciar o projeto totalmente do zero.

Qual a diferença entre uma **linguagem** e um **framework**?

Uma **linguagem de programação** possui uma lógica interna que usamos para expressar algo (no caso da programação, para construir softwares).

Já os **frameworks** são algumas ferramentas de código pré-prontas (em uma determinada linguagem) que nos ajudam a economizar tempo. É algo que você vai usar como base para a sua construção.

**NÃO
CONFUNDA!**





.NET

Um ambiente criado pela Microsoft, para o desenvolvimento, gerenciamento e execução de seus projetos. Em sua primeira versão .NET Framework, funcionada apenas em sistemas Windows, porém com sua evolução para .NET Core, 5 e 6, já funcionam em qualquer sistema operacional, tornando assim uma multiplataforma.

De forma geral, podemos dizer que ele está composto de dois elementos principais, um mecanismo que manipula aplicativos em execução e outro uma biblioteca de classes NET Framework, uma biblioteca de códigos que já foram testados e podem ser reutilizados pelos desenvolvedores.

Dentro dela encontramos as linguagens de programação como C# e VB,



VUE.JS

Um framework Javascript de Código aberto, ou seja, aberta para todos contribuírem, uma ferramenta **Front-end** muito utilizada para criar aplicações **SPA (Single Page Applications)** e também para desenvolver vários tipos de interfaces, que possuem necessidades de maior interação e experiência mais valorosa para o usuário.

Recursos: Com um conhecimento básico em HTML, CSS e JavaScript, você já consegue construir uma aplicação básica em Vue.

SPA é basicamente uma aplicação web construída em uma só página, na qual a interação e a navegação entre as sessões de uma página ocorrem de maneira a qual não é necessário recarregar a página em cada uma dessas mudanças.





ANGULAR

É um framework **TypeScript** de código fonte aberto para mantido pela Google para a construção de SPA em desenvolvimento das aplicações **Front-end** de páginas da web. Oferece suporte para inúmeras plataformas, estendido para aplicações móveis e programas de desktop. Uns dos maiores atrativos do Angular é praticidade no desenvolvimento e organização dos arquivos dentro do projeto. Os códigos são estruturados de modo simples para entendimento e manutenção, já que as funcionalidades disponíveis permitem modularidade e quebram o código em pedaços.

Você não precisa utilizar editores de texto e criar códigos de configuração extensos dentro da aplicação, poupando tempo a equipe. Atualização da página em tempo real. Fazer alterações no código e observar as mudanças de modo instantâneo.



DJANGO

É um framework web Python de alto nível que permite o rápido desenvolvimento de sites seguros e de fácil manutenção. Construído por desenvolvedores, o Django cuida de grande parte do trabalho de desenvolvimento web, para que você possa se concentrar em escrever seu aplicativo sem precisar reinventar a roda.

Usado pelo Instagram, Mozilla e o Pinterest, ele oferece inúmeras funcionalidades, desde a conexão com banco de dados, até gerar automaticamente a interface que vai ser utilizada para gerenciar o site.

Algumas funções que podem ser desenvolvidas por esse framework Python são: URLs amigáveis, sistema de templates, formulários, sistema de cache, autenticação do usuário, upload de arquivos.





BOOTSTRAP

É um framework front-end que fornece estruturas da linguagem CSS e HTML para a criação de sites e aplicações responsivas de forma rápida e simples.

Uma das suas maiores funcionalidades é poder lidar com sites de desktop e páginas de dispositivos móveis da mesma forma, por conta disso, os desenvolvedores não precisam se preocupar em fazer muitas versões de um mesmo site para todos os tipos e tamanhos de telas disponíveis.

Utilizado pelo Twitter, a ferramenta apresenta diversos templates como por exemplo, navegação, sistema de grades, carrosséis de imagens e botões.



LARAVEL

É um dos frameworks PHP mais utilizados no mercado, permite criar uma estrutura de códigos muito mais bonita e legível. Para alcançar esse objetivo, o Laravel utiliza arquitetura de desenvolvimento MVC (Model, View, Controller) e do padrão de desenvolvimento PSR-2 que tem como foco a escrita de código limpo e com boa legibilidade.

Conta com uma biblioteca de funcionalidades pré-programadas (como autenticação, roteamento e modelos HTML) que estão no pacote “agilidade” dessa ferramenta.

Em resumo, o framework permite concentrar os esforços nos “problemas” da aplicação, sem perder tempo na arquitetura e configurações, te oferecendo funcionalidades ou ferramentas pré-prontas para agilizar e otimizar o trabalho.



Bora falar sobre bancos de dados?

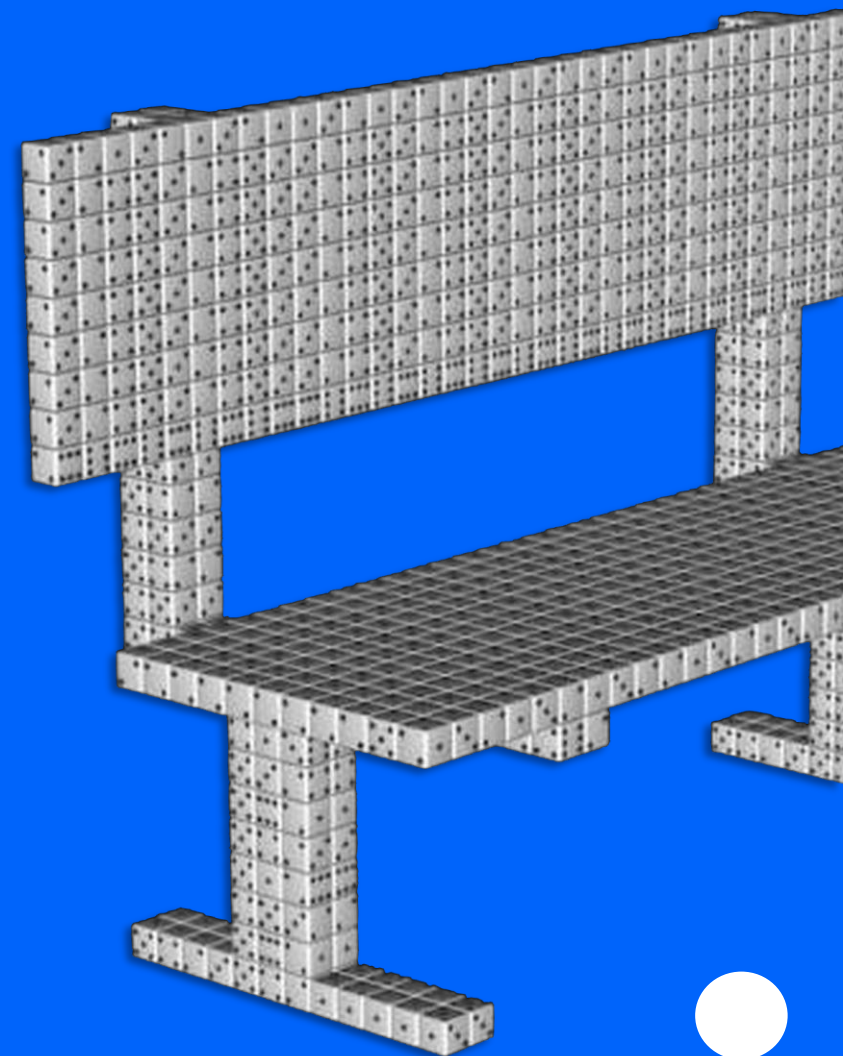
O banco de dados agrupa dados sobre um mesmo assunto, armazenando assim diversos dados como documentos, endereços, serviços, clientes, entre outros. Desta forma, um banco de dados serve para que estes dados sejam armazenados e consumidos posteriormente.

Como dito anteriormente, para seu gerenciamento, utilizamos os SGBDs (Sistema de Gerenciamento de Banco de Dados), são eles os softwares responsáveis por armazenar, listar, pesquisar, remover e qualquer outra manipulação feita em um banco de dados.

Existem dois tipos de SGBDs:

Relacionais (SQL) - Os SGBDs relacionais modelam os dados no formato de tabelas, que podem se relacionar entre si. Cada tabela pode possuir diversos atributos, com diversos tipos de dados (MySQL, PostgreSQL, OracleDB, SQL Server etc).

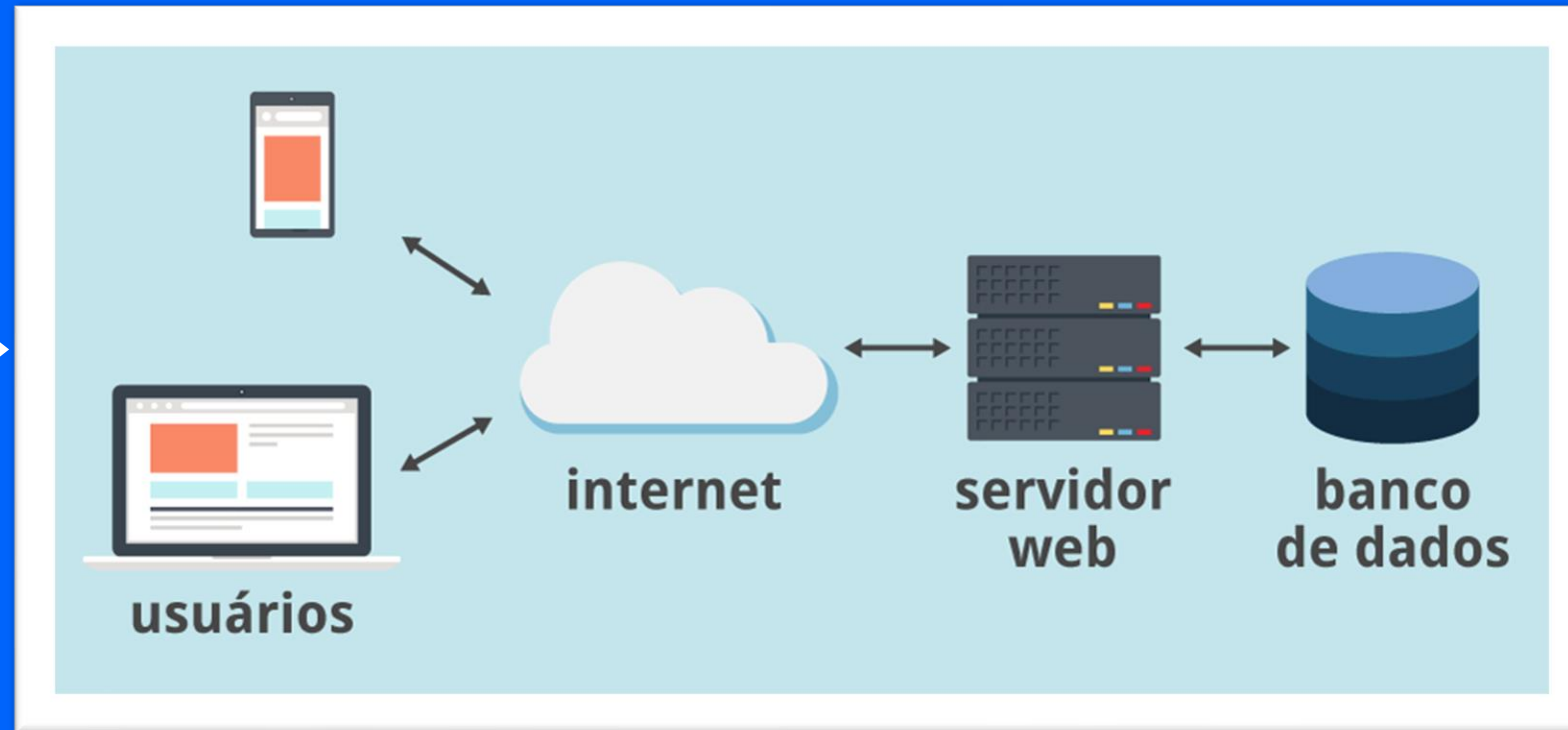
Não-relacionais (NoSQL) - NoSQL (Not Only SQL) é o termo utilizado para banco de dados não relacionais de alto desempenho, onde geralmente não é utilizado o SQL como linguagem de consulta. Estes bancos utilizam diversos modelos de dados incluindo documentos, gráficos, chave-valor e colunares. São amplamente reconhecidos pela facilidade em seu desenvolvimento, desempenho escalável, alta disponibilidade e resiliência (MongoDB, CassandraDB etc).



Como funciona um banco de dados na prática?

Essa requisição é enviado ao servidor que é o responsável por interagir e buscar as informações no banco de dados.

O usuário faz a requisição (busca) de algo através da internet em seus dispositivos.



Organização de equipes.

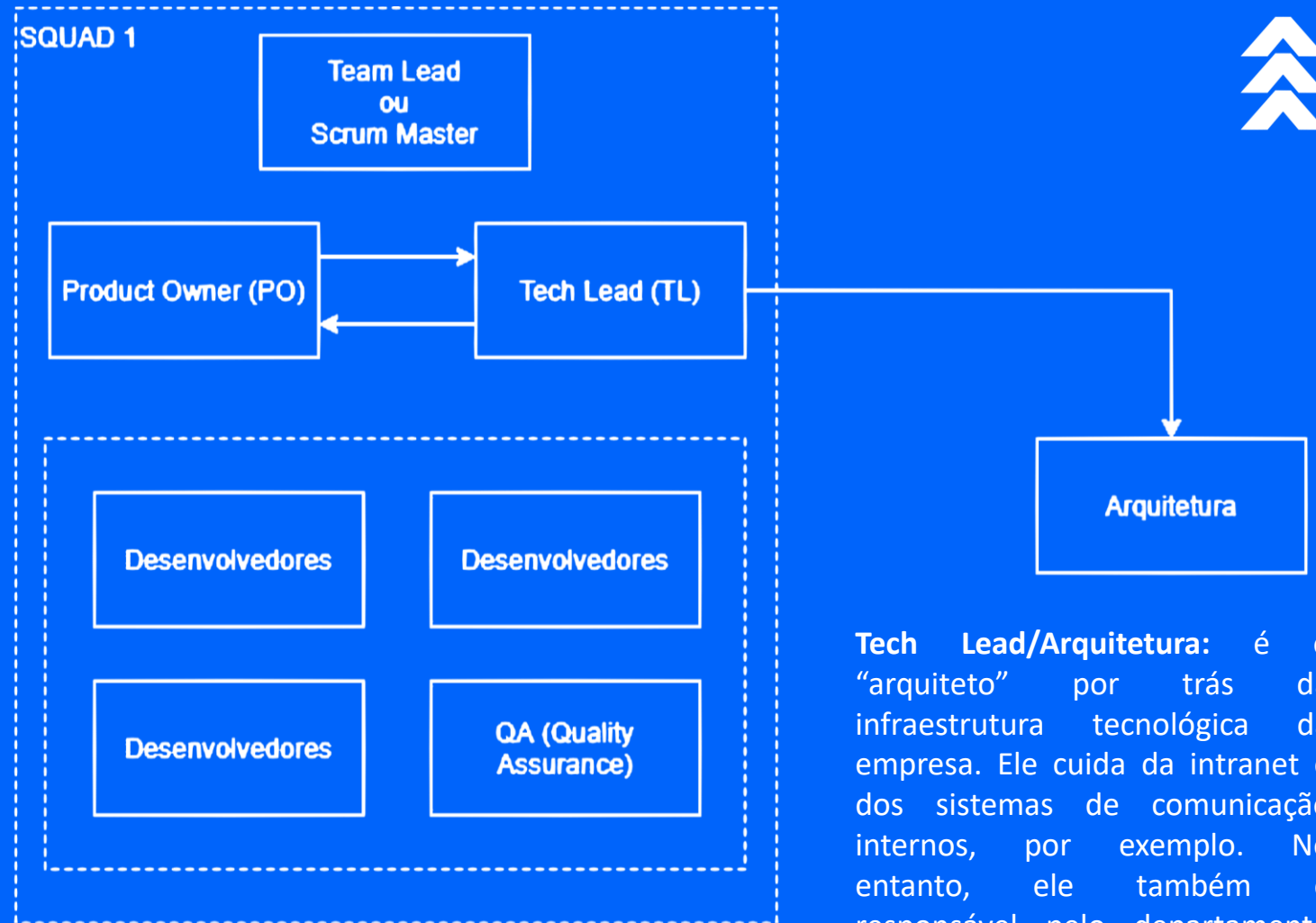
Como um time de T.I se organiza?

Team Lead/Scrum Master: é a pessoa responsável por garantir que todos estejam aderentes ao processo de desenvolvimento, que sigam suas cerimônias e que garantam a organização e realização dos seus artefatos.

Product Owner (PO): Dono do produto, pessoa responsável por definir quais as prioridades do product backlog e registrar de todos os requisitos, regras de negócio e prioridades das entregas de um produto.

Desenvolvedores: Time de desenvolvimento (pessoas que colaboram com o projeto).

Quality Assurance: Profissional de testes de desenvolvimento que visa garantir que um produto ou serviço a ser entregue aos clientes de uma empresa esteja de acordo com o padrão de qualidade pretendido. Evitar que o consumidor receba o que foi solicitado com qualquer tipo de falha na execução, erros, defeitos etc.



Tech Lead/Arquitetura: é o “arquiteto” por trás da infraestrutura tecnológica da empresa. Ele cuida da intranet e dos sistemas de comunicação internos, por exemplo. No entanto, ele também é responsável pelo departamento de Engenharia da companhia. Procura formas de melhorar os produtos oferecidos por meio da tecnologia.

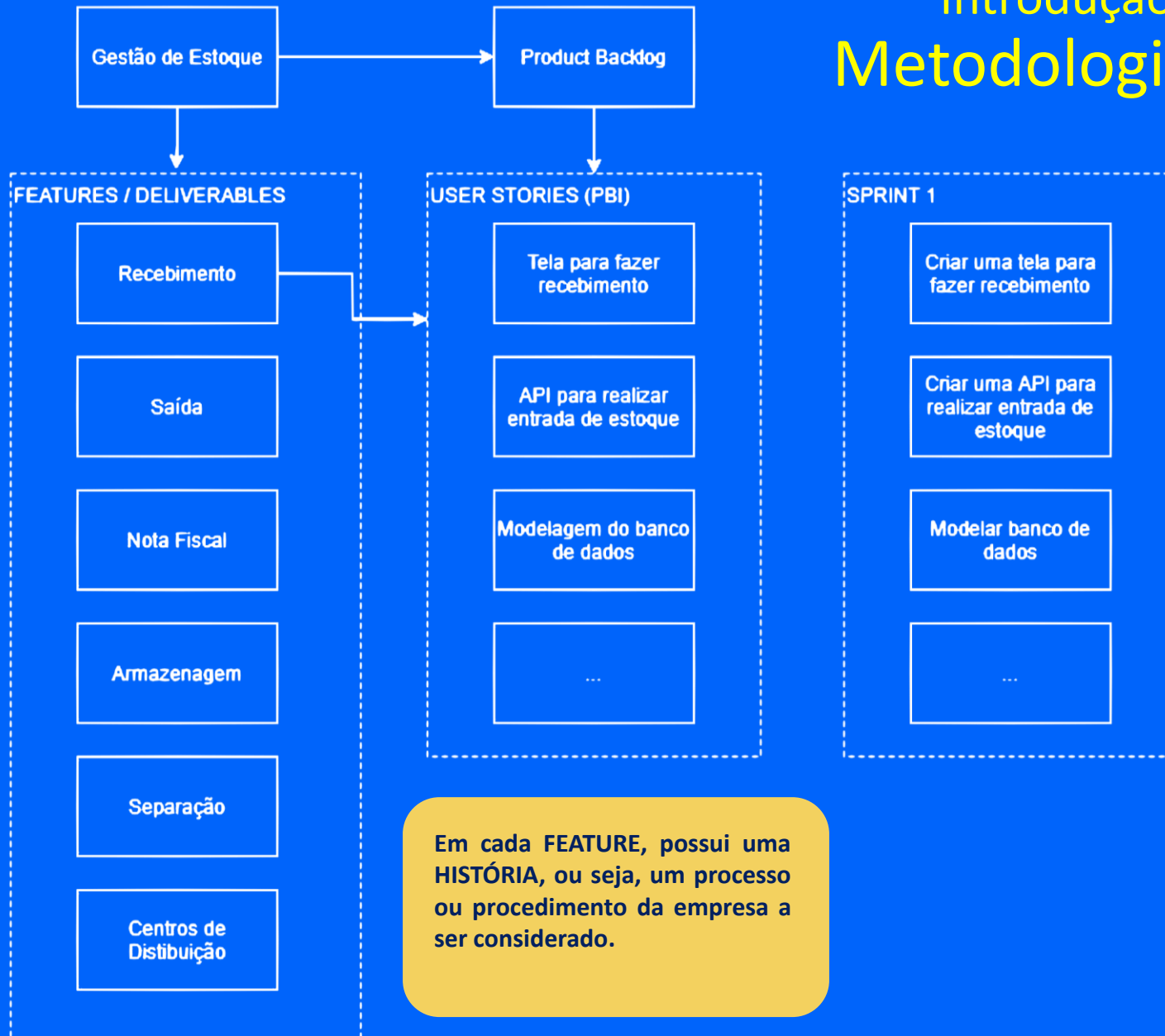


Visto anteriormente, os profissionais se dividem em SQUADS, para atuação em determinado tipo de produto.

Em uma empresa, existem diversos sistemas e setores que a tecnologia chama de PRODUTO; que é o setor de estoque, vendas, atendimento, financeiro, departamento pessoal etc. Em algumas empresa para cada produto, possui uma SQUAD.

Utilizando o exemplo para desenvolvimento de um sistema de gestão de estoque.

Dentro desse produto, temos os recursos, chamados de FEATURES ou DELIVERABLES.



Em cada FEATURE, possui uma HISTÓRIA, ou seja, um processo ou procedimento da empresa a ser considerado.

Introdução à Metodologia Ágil.



As HISTÓRIAS, geram as SPRINTS, que é o que fazer com aquela necessidade, aquele processo que foi identificado no modelo do negócio.

Daí as SPRINTS são distribuídas para os desenvolvedores e são acompanhados através da metodologia ágil.



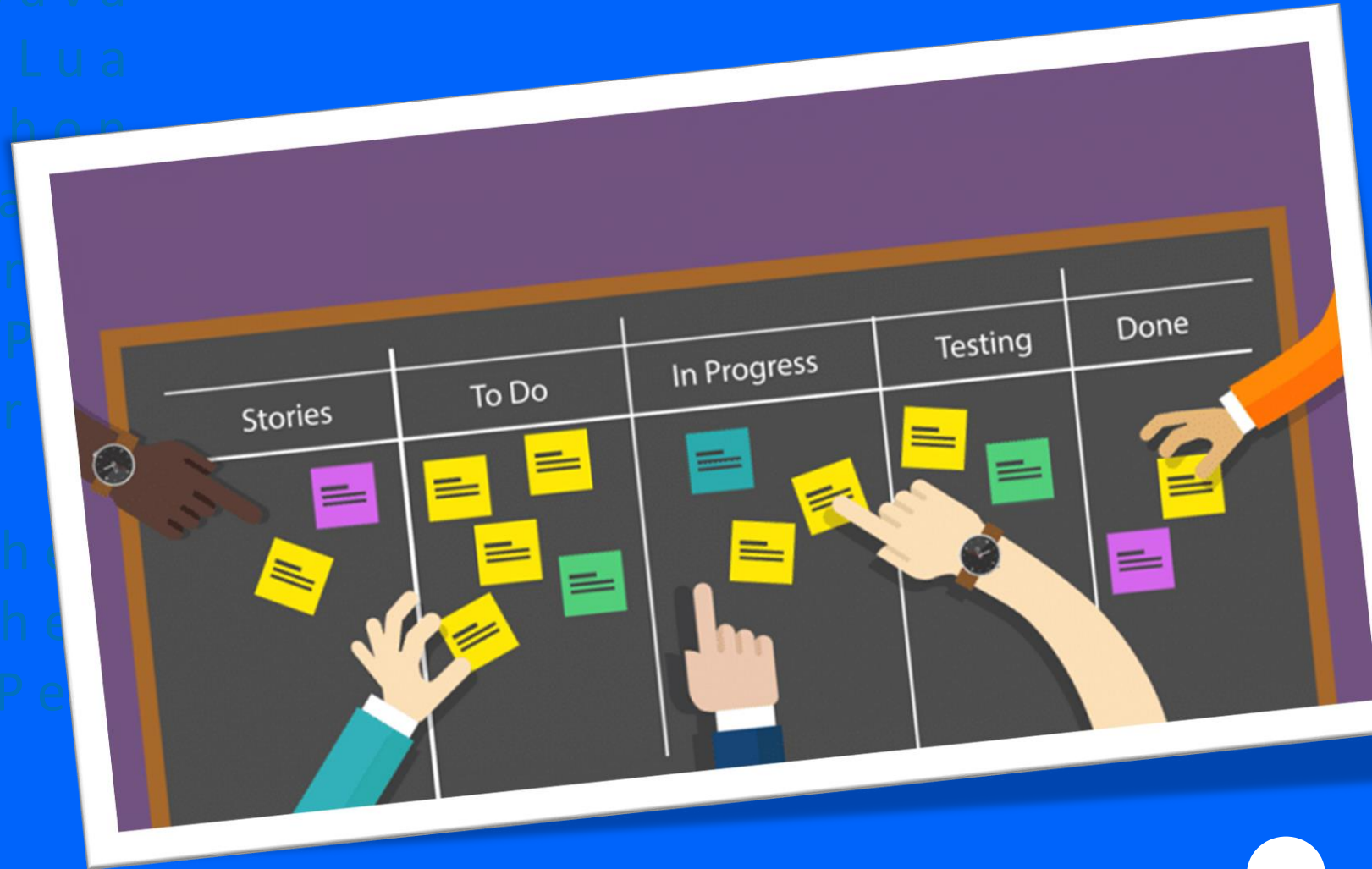


Metodologia Ágil.

A metodologia consiste em identificar todos os recursos disponíveis, dividir seus segmentos e suas histórias e organizar cada SPRINT, para identificar e acompanhar:

- O QUE FAZER?
- O QUE ESTÁ SENDO FEITO?
- O QUE ESTÃO SENDO VALIDADO?
- O QUE JÁ FOI CONCLUÍDO?

Assim se torna mais fácil identificar os componentes de um projeto, seja voltado a engenharia de software ou quaisquer projetos organizacionais.



Dicionário técnico na programação.

- **Algoritmo:** é basicamente um conjunto de passos para certa tarefa.
- **Linguagem:** conjunto de definições (sintáticas e semânticas) usadas para fazer um texto que o computador é capaz de entender, ou seja, um programa de computador.
- **Código:** conjunto de palavras ou símbolos contendo instruções para o computador.
- **Código Fonte:** código de um programa. O computador o usa para gerar uma versão do programa que é capaz de processar.
- **Código de Máquina:** código que a máquina consegue entender e executar.
- **Código Binário:** Código feito por apenas 0 e 1 que tem como função codificar algo. É a base de toda a lógica do computador. Tudo vira, no fim das contas, código binário.
- **Bug:** problema no código que faz com que ele não execute sua função corretamente.
- **Refatoração:** processo de alterar um programa para melhorar sua estrutura interna sem alterar o seu funcionamento. Essas melhorias podem ser de performance ou para facilitar trabalhos futuros neste código, por exemplo.
- **Frameworks:** são como coleções de ferramentas, componentes e soluções que você pode encontrar em um mesmo pacote (o framework) que facilitam a criação de aplicações complexas.
- **Queries:** pedido de informação e/ou dados para banco de dados.
- **Front-End:** é a parte de uma aplicação web que a pessoa que utiliza, vê e interage. É executada pelo browser.

- **Back-end:** o que o usuário não vê num software, como servidores e banco de dados. É o processamento “por trás dos panos”.
- **Full-stack:** junção de Front-End e Back-End.
- **SSL (Secure Sockets Layer):** é um tipo de segurança padrão para garantir que todos os dados sejam criptografados no trânsito entre um site e um navegador, por exemplo.
- **SOLID (Single-responsibility, Open-closed, Liskov substitution, Interface segregation e Dependency Inversion principles):** esses princípios, quando combinados, são tidos como boas práticas para facilitar a quem programa desenvolver softwares que são mais fáceis de se manter e expandir.
- **Backdoor:** porta criada por algum aplicativo malicioso com a finalidade de estabelecer um canal de comunicação do nosso dispositivo para um ponto no qual os nossos dados ou ações possam ser manipulados.
- **Backup:** cópia de arquivos em um outro dispositivo com objetivo de guardar informações.
- **Bandwidth** – largura de banda, em tradução livre. Diz-se do quanto é possível trafegar em uma rede.
- **Benchmarking:** ferramenta utilizada para medir e comparar o desempenho de uma máquina.
- **Big Data:** armazenamento de grande quantidade de dados.
- **BIOS (Basic Input Output System):** sistema utilizado para listar os dispositivos padrão de entrada (input) e saída (output) de um computador, isto é, o sistema consegue identificar o que está ligado a uma máquina como, por exemplo, teclado, mouse, monitor, HD, etc.



Dicionário técnico na programação.

- Bit:** abreviação de binary digit, ou seja, representa somente um valor binário (ou zero ou um).
- Bitcoin:** criptomoeda criada para estabelecer uma forma de pagamento virtual em diversos serviços digitais.
- Blockchain:** protocolo de comunicação descentralizado utilizado, principalmente, para controlar as transações com criptomoedas. O controle do protocolo é feito por máquinas distribuídas em uma rede, sendo que estas criam um elo (daí o nome chain – corrente) entre os pacotes e seus destinatários.
- Boot:** processo de inicialização de um sistema onde são verificadas se as peças e configurações mínimas necessárias estão todas funcionais.
- Botnet:** rede de computadores interconectados que executam uma ou mais tarefas para as quais foram programados. Geralmente essa rede fica associada ao uso de programas maliciosos para ataques em massa ou também para envio de spam.
- Byte:** união de 8 bits
- Mixin:** o termo mixin refere-se a uma classe que se estende de 2 ou mais classes, ou seja, sendo B e C classes, A herda ao mesmo tempo de B e C. Porém, para ser considerado um mixin e não uma herança múltipla, B e C não podem ter função inicializadora, do contrário não temos um mixin e sim a simples utilização da herança múltipla.
- Qualificação de nomes:** qualificação de Nome é utilizado nas linguagens de programação para se referir a nomes que poderão ser acessados através de outro nome.

- BLOB (binary large object):** o seu objetivo quando relacionado a Banco de Dados é sempre o armazenamento de grandes quantidade de dados no formato de bytes.
- Tipado:** significa uma variável ou algum valor que possui um tipo pré-determinado.
- Recursividade:** recursividade ou Função Recursiva significa a invocação de si mesmo. Ou seja, é uma função que após executar o seu bloco de instrução invoca a si mesma novamente.
- Olá, mundo:** o termo “Olá, mundo” se refere ao trecho de código mínimo para a impressão na tela da frase “Olá, mundo” com determinada tecnologia. O principal objetivo desses códigos é demonstrar o mínimo necessário para o desenvolvimento de uma aplicação com determinada tecnologia.
- Shell:** janela, normalmente preta, para a manipulação do sistema operacional através da linha de comando. Sistemas Operacionais Unix sempre disponibilizam uma janela para a interação com o sistema operacional através da linha de comando.
- Console:** janela, normalmente preta, em que é possível trabalhar na linha de comando.
- Paradigma de programação:** modelo de estrutura utilizado pela linguagem de programação. Há vários paradigmas existentes e cada linguagem utiliza um ou vários paradigmas.
- Iteração:** iteração é o nome das estruturas que repetem um mesmo bloco de código por uma quantidade de vezes finita, ou pelo tempo em que uma condição for verdadeira. Pode-se dizer que todas as linguagens de programação para computadores possuem, no mínimo, uma estrutura de iteração.



Dicionário técnico na programação.

- Classe:** projeto de todo objeto.
- Instância:** a execução de uma classe.
- Objeto:** a execução de uma classe.
- Herança:** capacidade de herdar as características de outra classe
- Polimorfismo:** capacidade de ter funções com mesmo nome e assinatura, porém, com comportamentos diferentes.
- Membro de Classe:** pode ser uma função, ou então, uma propriedade (variável).
- Propriedade:** característica do objeto e na programação, é representado por uma variável.
- SuperClasse:** a classe pai; classe que está sendo estendida.
- SubClasse:** a classe filha; a classe que inicia a partir da implementação de outra.
- Classe-Pai:** sinônimo de SuperClasse; classe utilizada como base para a criação de outras.

Termos de ferramentas de programação

- Sistema operacional:** sistema que faz comunicação entre o hardware, gerenciando e distribuindo seus recursos, e os demais softwares e cria uma plataforma comum a todos os programas utilizados. Exemplo: Windows 10, Ubuntu.
- Terminal:** é uma Interface de Linha de Comando, isto é, uma interface (meio) para a pessoa enviar comandos ao computador.
- Editor de Texto:** um editor de texto é um aplicativo para edição de arquivos de texto. Isso inclui códigos de todos os tipos, que também são textos escritos num formato que o computador é capaz de entender.

- Compilador:** transforma o código de uma linguagem de programação para uma linguagem de máquina (ou alguma de nível mais baixo, ou seja, mais próxima do código binário que o hardware processa) que é lida pelo processador.
- Linguagem de máquina:** linguagem de mais baixo nível de entendimento para o ser humano, e a única entendida pelo processador (CPU).
- Interpretador:** programas de computador que leem um código-fonte de uma linguagem de programação interpretada, ou seja, convertida em código binário uma linha por vez (ao invés de grandes blocos de linhas de cada vez) e transformada em código executável.
- Browser ou Navegador:** programa desenvolvido para permitir a navegação pela web e visualização de suas páginas.
- Plugin:** uma extensão de código que pode ser adicionada a um código em que você está trabalhando. O plugin é uma extensão intrínseca ao contexto, por exemplo, um plugin de um navegador que não funcionará em outro navegador. Além disso, ele funciona consumindo código ou APIs da aplicação principal.



Dicionário técnico na programação.

Termos de desenvolvimento front-end

- **HTML (Linguagem de Marcação de Hipertexto):** linguagem de marcação que é utilizada como estrutura do Front-End de um site.
- **CSS (Folhas de Estilo em Cascata):** linguagem de estilo que define a aparência de documento web HTML. Exemplo: manipulação da posição dos elementos, cores, fontes, fundos, etc.
- **JavaScript:** linguagem que viabiliza a criação de páginas interativas, ou seja, que são modificáveis de acordo com a interação de quem usa e que conectam o Front-End ao Back-End.
- **AJAX (Asynchronous JavaScript And XML):** técnica para acessar servidores web a partir, por exemplo, do Front-End ou de outros Back-Ends.
- **jQuery:** biblioteca de funções em JavaScript que interagem com o HTML de uma maneira muito mais rápida e simples do que apenas pelo JavaScript.
- **Bootstrap:** framework CSS.
- **UI Design (Design da Interface do Usuário):** é o desenho e execução de uma interface para o usuário. Exemplo: como garantir que um botão vai ser visto e apertado.
- **UX Design (Design da Experiência do Usuário):** é o desenho e execução de uma experiência para o usuário. Exemplo: desenhar as ações que farão com ele se sinta bem ao entrar numa página, ou engajado, etc.
- **Responsive design:** o design que assegura que o site será visto de uma forma diferente para variados tamanhos de tela.

- **Adaptive design:** uma maneira de se construir um mesmo site que se adapta (muda) para cada dispositivo.
- **Fluid Design:** design que acompanha o tamanho da tela do dispositivo que você está usando sem mudar a estrutura da página.
- **Progressive Web App:** metodologia que busca fazer com que sites web e app tenham características de aplicativos nativos de dispositivos mobile, ou seja, feitos primariamente para eles.

Termos de desenvolvimento front-end

- **Servidor:** máquina onde roda o Back-End de um site, por exemplo, ou um Back-End acessível a outras aplicações diversas.
- **Node.js (node):** interpretador para JavaScript.
- **NPM (Node Package Manager):** Gerenciador de Pacotes (módulos) do Node.js.
- **Middleware:** software que se encontra entre o sistema operacional e os aplicativos nele executados. Ou seja, é um software que permite a interação entre sistemas isolados e funções. Exemplo: site com uma opção de pagamento. O que conecta os dois é um middleware.
- **Express.js:** framework para Node.js que ajuda na criação de rotas, middlewares e muitas outras funções para o desenvolvimento tanto de APIs quanto SPAs (Aplicações de Uma Página).



Dicionário técnico na programação.

•**API (Applied Programming Interface)**: interface que conecta dois programas, realizando tanto a comunicação entre eles e especificando quanto definindo como devem interagir.

•**Serviço Web**: solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes (é um programa web de Back-End, um tipo de API que precisa de uma rede para se comunicar).

•**Webhook**: é uma forma de recebimento de informações quando um evento (enviar um email, por exemplo) acontece. Por isso o webhook é, na prática, uma forma passiva de transmissão de informações entre dois sistemas, pulando o passo de request (requerimento) que uma API faz. Basicamente: um sistema disponibiliza um canal para receber informações (o webhook) e outros enviam para esse sistema informações através dele.

•**HTTP (HyperText Transfer Protocol)**: protocolo que especifica como clientes e servidores se comunicam entre si.

•**REST (REpresentational State Transfer)**: estilo de arquitetura de serviços web que padroniza algumas características tidas como boas práticas.

•**Restful**: é uma aplicação web que realiza a implementação do padrão REST.

•**CRUD (Create, Read, Update and Delete)**: quatro operações básicas de interação com uma base de dados (criação, consulta, atualização e destruição de dados).

Termos de metodologias

•**Pair programming**: técnica do desenvolvimento ágil de software onde duas pessoas utilizam um mesmo computador para programar alguma tarefa.

•**Sprint**: ciclo de tempo dentro do qual se realiza um conjunto de atividades.

•**Kanban**: sistema simples e visual para organizar as tarefas e o fluxo de trabalho.

•**Ágil**: conjunto de métodos e princípios que consiste na ideia de desenvolvimento iterativo (isto é, onde requisitos e soluções são identificados e cumpridos de maneira cíclica) a partir da colaboração de times auto-gerenciáveis e multifuncionais.

•**Scrum**: framework para gestão e planejamento de produtos e projetos.

•**MVP (Produto Mínimo Viável)**: maneira de se testar modelos de negócios e conceitos diretamente com o cliente, a partir do lançamento de uma versão mínima viável do produto. Exemplo: em vez de fazer um site como o que a Airbnb possui, os fundadores fizeram um site simples com poucas funcionalidades.

