**Sobre o bootcamp da DIO \/**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**Sobre o desenvolvimento moderno de software Desktop vs sistemas para internet (Web).**

Desktop: sistemas autônomos instalados no computador, ex: excel, word, jogos. Aplicativos para desktop só podem ser utilizados localmente na máquina que forem instalados, esta é uma grande desvantagem, e podemos perceber isso pensando em um aplicativo de banco que só funciona no computador específico do indivíduo. É por isso que esse tipo de aplicação tem caído em desuso para muitos serviços.

Já um sistema desenvolvido para web pode ser acessado de qualquer lugar, e por essa enorme vantagem muitos serviços estão sendo transportados para web

Ainda temos os casos das aplicações moveis que precisam de um instalador de acordo com a plataforma (android ou ios).

Dessa forma antes de criar um serviço precisamos pensar em quais plataformas queremos disponibilizar ele.

Diferentes áreas do desenvolvimento de software.

**UX (user experiencie)**

È este profissional que pesquisa e monta o perfil do cliente baseando-se em dados como idade perfil socioeconômico gênero e etc. é ele também responsável por criar o que chamamos de wireframe ou seja a estrutura do site para que este esteja de acordo com o usuário e assim possa proporcionar melhor usabilidade, acessibilidade e satisfação na interação com o software.

Uma imagem contendo Tabela

Descrição gerada automaticamenteDiagrama

Descrição gerada automaticamente

Nesta parta o profissional de UX/UI pensa onde alocar cada botão, qual o caminho o usuário vai percorrer para chegar até certa parte do software, quais são os melhores locais para evidenciar informações, quais informações devem estar em evidência e muitas outras coisas que envolvem a experiência do usuário.

Outro profissional que cuida da experiência do usuário é o UI user interface), depois que o UX pensa na estrutura do site é ele que cuida das cores, da tipografia, do micro interação, dos estilos etc, este é mais voltado a aparência. Estes dois profissionais precisam trabalhar de forma integrada.

**Front end:**

No desenvolvimento moderno de software temos a estrutura de Cliente-Servidor. Por exemplo em uma aplicação de banco quando consulto meu saldo, a parte que o cliente interage faz uma requisição deste saldo que vai até os servidores para resgatar esse dado e exibir ao usuário.

Diagrama, Esquemático

Descrição gerada automaticamente

Esta parte visual que o usuário interage é o que chamamos de Frond End, e aparte que faz essa requisição e resgate de dados além do armazenamento e outras funções é o Back End. Temos profissionais especializados em cada parte em um projeto normalmente pois os dois possui linguagens especificas para cada um.

por exemplo o frond end utiliza linguagens como o html, css e Javascript em conjunto com uma IDE que é um software que ajuda o programador com várias ferramentas, como compiladores, revisores de código etc.

**Framework**

O framework é um facilitador basicamente. Ele traz várias soluções pré definidas que você pode utilizar no seu código e é utilizado para agilizar o processo de criação de aplicativos já que muitas vezes eles nos ajudam a eliminar tarefas repetitivas. É importante verificar se o framework que você utiliza é compatível com a linguagem que você utiliza no seu programa. Temos como exemplos: angular para front and (trabalha com html css e javascript), spring para o back and (trabalha com java). Porém antes de trabalhar com um framework precisamos aprender a linguagem que ele utiliza pois muitas vezes estas ferramentas são descontinuadas e o que continua é a linguagem que ela utiliza

**Back and**

Como o nome diz a ideia vem do que tem por traz da aplicação, a parte que o usuário não interage de forma direta, porém influencia na experiencia, por exemplo seu usuário do ticoteco esta armazenado em um banco de dados em um servidor, e é la que os seus dados são processados e consumidos pelo usuário de forma visual. Dessa forma quando você faz uma requisição de verificação do seu usuário existe, e uma vez verificado ele retorna positivo ou negativo, você utilizou uma aplicação que o backand criou. O backand também cria regras de negocio, por exemplo como o desconto vai ser aplicado a um produto do usuário, ou como um item vai ser cadastrado e armazenado no banco de dados ou como o usuário vai verificar estes produtos, além das validações destes dados.

E para desenvolver tudo isso o programador backand utiliza o que chamamos de API ou aplicação de interface de programa, basicamente é o que faz o intermédio entre o backand e o front end, e dessa forma que guardamos, requirimos e tratamos dados de acordo com as regras de negocio e com as validações necessarias

O programador backand utiliza linguagens como o JAVA PHP e C#

Fluxograma

Gráfico

Descrição gerada automaticamente com confiança média

**Programador Full Stack**

Este profissional atua tanto no back quanto no front end, podendo atuar no desenvolvimento do software de ponta a ponta, é comum que esse profissional tenha uma especialização em uma área mas também tenha conhecimento suficiente para atuar em outras.

**Infraestrutura**

È a parte física do programa, afinal a exibição do front end acontece no hardware do cliente, porem o back, ou seja as aplicações os e o banco de dados precisam estar armazenados em hardwares também. Hoje em dia nós temos muitos serviços que o armazenamento de dados, aplicações e gerenciamento, a maior parte em nuvem que oferecem possibilidade de escalabilidade dinâmica.

Contudo tudo isso precisa estar em uma parte física, e esta parte física são servidores que ficam em datacenters (lugares próprios para servidores) Foto em preto e branco de computador

Descrição gerada automaticamente com confiança média

Temos alguns tipos de profissionais que cuidam dessa estrutura de cloud

1. Segurança de tecnologia da informação (logica e fisica)
2. Mão de obra especializada (software e hardware) no software não temos sistemas operacionais convencionais nestes servidores e quem cuida são os sis admins, são especializados nos servidores.
3. Infraestrutura local (pessoas que entendem de redes de computadores, cabeamento, backup, swit e roteadores e etc.)

**Cloud publica**

São sistemas de armazenamento de dados oferecido por empresas, via internet de acordo com o a utilização de cada um. ou seja as empresas não precisam ter servidores fisicos hoje em dia, elas só precisam alugar computadores de empresas como a amazon ou a google e pagar só pelo que for utilizado.

As vantagens de utilizar esse tipo de serviço são

1. Preço ( pague somente o que usar)
2. Facilidade de contratação, configuração, e infraestrutura.
3. Escalabilidade.
4. Performance.

Temos o profissional de cloud computing, que contrata o serviço conforme a demanda, configura e cuida da escalabilidade e da manutenção.

Texto

Descrição gerada automaticamente

**Mobile desenvolvimento**

No mobile nos temos o mesmo backand que as aplicações web, contudo o frontand muda de acordo com a plataforma Android ou Ios

Android: plataforma android studio que oferece acesso a tudo que um dispositivo móvel tem, tipo gps câmera etc. Como linguagem de programação utilizamos o JAVA ou o KOTLI ou os dois em conjunto

Ios: A plataforma de desenvolvimento é o excold e a linguagem de programação é o swift. (é necessário ter algum produto apple, como um macbook para desenvolver nesta plataforma e linguagem)

Desenvolvimento híbrido: este tipo de desenvolvimento atende as duas plataforma através do desenvolvimento web que através de certas frameworks e bibliotecas, conseguem encapsular o web em um aplicativo e atender a todos. Como opções nos temos o Ionic, flutter e react nativo. (nestas três opções nós temos o html css e Javascript, porém temos acesso aos recursos moveis citados acima como gps e etc.)

Este tipo de desenvolvimento é menos custoso, porém não é recomendado para todos os serviços por não ter a mesma performance a depender do produto.

Nos separamos os dois primeiros na classe de desenvolvimento nativo e este último tipo em desenvolvimento hibrido.

**QA (Quality assurance)**

Este profissional é responsável por cuidar da qualidade dos serviços que o software realiza, ou seja sua confiabilidade, se a usabilidade esta de acordo com o planejado (se o caminho que o usuário realiza esta sendo utilizado de maneira correta) e outras varias questões que envolve a entraga de um produto de qualidade verificado pelo QA.

Para testar a qualidade dos softwares os QAs utilizam ferramentas de testes automatizados para verificar se o código não tem bugs nem falhas, também analisam a qualidade dos códigos e outras questões relacionadas aos padrões de qualidade exigidos pela empresa que está disponibilizando o serviço assim como geram relatórios de erros para que estes possam ser discutidos e concertados.

Este profissional precisa conhecer a fundo as atividades do projeto assim como ter um perfil analítico. Porém não é exigido que o mesmo saiba programar, por mais que seja algo recomendável, este só precisa conhecer as ferramentas de testes e de criação de relatório além de entender das atividades do produto