Ao desenvolver softwares, pensar sobre:

-Quais plataformas usar;

-Problemas a resolver.

Sistema desktop: Sistemas autônomos para computador. Realizada por arquivo executável, como por exemplo Microsoft Word, Excel, anti-vírus, jogos e etc

Desvantagem: Uma vez que o sistema está instalado em um PC específico, não há acesso ao mesmo longe do computador.

Sistema web: Sistemas baseados em tecnologia web, que podem ser utilizados através de qualquer navegador de internet, sem necessidade de aparelho local específico.

Aplicação móvel: Feitos para instalação em smartphones e/ou tablets. Baixado pela Google Play ou App Store.

Partes do desenvolvimento de software:

-UX Design (User Experience): Processo que visa melhorar a satisfação do usuário com um produto ou serviço, melhorando a usabilidade. Leva em consideração através da pesquisa dados como faixa etária do usuário, interesses pessoais, etc (parte comportamental). Trabalha no front-end (visual), cria um protótipo para o programador seguir;

-UI Design (User Interface): Profissional responsável em criar o que o usuário verá no programa. Foca em cores, tipografia, microinterações, estilos etc (aparência). Trabalha em conjunto com o UX Designer;

-Modelo Cliente-Servidor: Estrutura de aplicação que distribui as tarefas e cargas de trabalho entre os servidores e os clientes. Tarefas feitas por clientes, também chamados de requerentes: logins, buscas por informações dentro de aplicações, etc. As respostas de tais tarefas são fornecidas por servidores.

-Front End: O desenvolvedor front end programa a aplicação utilizada pelo cliente durante o acesso e requerimentos. É responsável pela parte visual de um site ou aplicativo com o qual podemos interagir. Desenvolve por meio de códigos uma interface gráfica, geralmente através de tecnologias de base da Web (HTML, CSS e JavaScript).

-Back End: O desenvolvedor back end desenvolve a aplicação utilizada pelo fornecedor, que responde aos requerimentos dos clientes. Faz a ponte entre dados que vem do navegador rumo ao banco de dados e vice-versa, aplicando as devidas regras em um ambiente no qual o usuário final não tem acesso. Para isso, utiliza-se linguagens como JAVA, PHP, Python e C#.

Códigos são criados a partir de uma IDE (ambiente de desenvolvimento integrado, que são softwares para criar aplicações que combina ferramentas comuns de desenvolvimento em uma única interface gráfica do usuário (GUI).

API (Application Programming Interface) é um intermédio de software que permite que dois aplicativos conversem entre si. Ex: Ao usar um aplicativo como Facebook, enviar mensagens ou verificar a previsão do tempo no celular, você está usando um API.

-Framework: Facilitador que traz diversas soluções já pré-definidas e descomplicam o trabalho do desenvolvedor. Ex: Angular (HTML, Javascript e CSS), Laravel (PHP), Vue (Javascript), Spring (Java, especialmente back end) etc.

-Full stack: Desenvolvedores com conhecimentos técnicos capazes de desenvolver uma aplicação de ponta a ponta. Entende de todo o processo, do back end ao front end.

-QA (Quality Assurance): Profissional que cuida da qualidade do software. Analisa a qualidade dos códigos criados para evitar possíveis erros e identificar se o caminho do usuário é exatamente o previsto anteriormente.

-Infraestrutura: Ao pensar na infraestrutura de uma empresa quando falamos de criação e manutenção de aplicações, é importante considerar se a mesma utilizará uma nuvem pública ou privada. Caso haja a necessidade de uma nuvem privada, devemos pensar em toda a infraestrutura de hardware, e, assim sendo, também teremos mais profissionais envolvidos no processo além dos citados anteriormente, para que haja a manutenção correta da nuvem. São esses:

-Profissional de segurança da tecnologia da informação (lógica e física);

-Mão de obra especializada para software e hardware.

Nuvem pública: Sistemas de armazenamento de dados disponibilizados via internet, em vez de servidores físicos tradicionais. Ex: Google Cloud Platform, Alibaba Cloud, Amazon web services, Azure, Liquid Web, Vmware, Salesforce, etc. Nuvens públicas são mais acessíveis, e, por isso, muitas empresas estão dando preferência à esse tipo de armazenamento.

Vantagens:

-Preço;

-Facilidade de contratação, configuração e infraestrutura;

-Escalabilidade;

-Performance.

Profissional de cloud computing: É responsável pela infraestrutura de nuvem oferecida aos clientes. Escolhe as tecnologias que serão utilizadas pela empresa a qual presta serviço.

Desenvolvimento mobile (Android): Pode utilizar o mesmo back end que o desenvolvimento web, muitas vezes. Sua maior peculiaridade é seu front end, que além necessitar da linguagem de programação também muitas vezes requer uma plataforma de desenvolvimento. Para sistema Android temos o Android Studio, que disponibiliza recursos encontrados em celulares, tablets e outros dispositivos Android. Para a parte de programação, podemos usar Java e/ou KOTLIN.

Desenvolvimento mobile (IOS): Usado em dispositivos Apple, também só podem ser desenvolvidos através de dispositivos Apple, como Mac Pro, entre outros. Utiliza-se o XCode como plataforma de desenvolvimento, e o SWIFT para programação.

Desenvolvimento híbrido (Web): Atende tanto Android quanto IOS através de um código criado em web usando bibliotecas como IONIC, Flutter ou React Native. Seu código back end, por ser web, é desenvolvido em CSS, Javascript ou HTML. Como não se trata de uma aplicação nativa apenas para IOS ou Android, não consegue usar todas as ferramentas dos dispositivos mobile, comprometendo a performance.