



# SOFTWARE

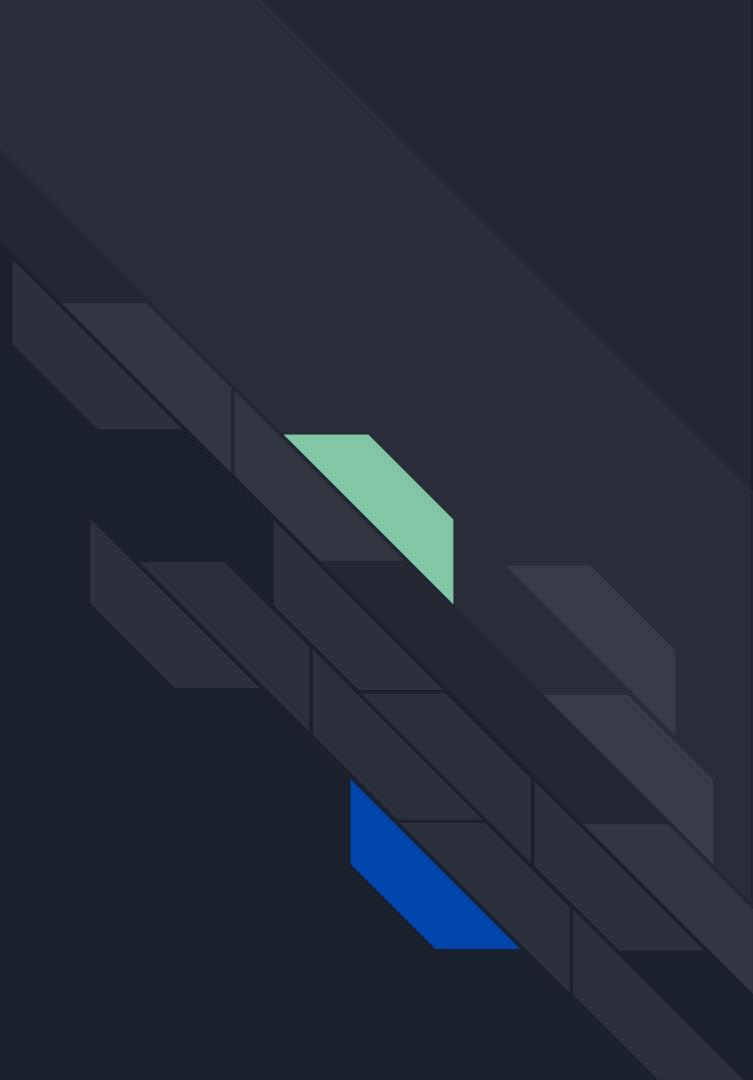
*APP*



# Aplicação Mobile (APP)

- Software que roda em um sistema operacional mobile
- Acelerar processos, diminuir custos e diminuir mão de obra
- Nativo e Híbrido
- Android Studio e Xcode
- Ionic e React Native
- Vantagem da utilização de IDE e Framework
- A utilização de API nos APPs

# *Framework*





# Conceito

- Um framework é definido como um software parcialmente completo projetado para ser instanciado. O framework define uma arquitetura para uma família de subsistemas e oferece construtores básicos para criá-los. Também são explicitados os lugares ou pontos de extensão (hot-spots) nos quais adaptações do código para um funcionamento específico de certos módulos devem ser feitas. Em síntese o framework é um conjunto de classes que colaboram entre si proporcionando melhores práticas de desenvolvimento e diminuição à repetição de tarefas. Além disso, evita variações de “soluções diferentes para um mesmo tipo de problema”. O que facilita a reutilização e customização dos códigos.



# Benefícios decorrentes da utilização de frameworks:

- Os principais benefícios decorrentes da utilização de frameworks, segundo Fayad et al. (1999b) e Pinto (2000) advêm da modularidade, reusabilidade, extensibilidade e inversão de controle que os frameworks proporcionam. Frameworks encapsulam detalhes de implementação voláteis através de seus pontos de extensão, interfaces estáveis e bem definidas, aumentando a modularidade da aplicação. Os locais de mudanças de projeto e implementação da aplicação construída usando o framework são localizados, diminuindo o esforço para entender e manter a aplicação.
- Além disso, frameworks incentivam o reuso, pois capturam o conhecimento de desenvolvedores de determinado domínio ou aspecto de infra-estrutura ou de integração de middleware. As interfaces do framework promovem pontos de extensão que as aplicações estendem gerando instâncias que compartilham as funcionalidades no framework. Desta forma, aumentam a produtividade dos desenvolvedores, pois o processo não começa do zero, a qualidade e a confiabilidade do produto final, pois idealmente as funcionalidades do framework já foram testadas em diversas instâncias, e a interoperabilidade de software, pois as instâncias de uma mesma família compartilham características em comum.



# Benefícios decorrentes da utilização de frameworks:

- Frameworks oferecem pontos de extensão explícitos que possibilitam aos desenvolvedores estenderem suas funcionalidades para gerar uma aplicação. Os pontos de extensão desacoplam as interfaces estáveis do framework e o comportamento de um determinado domínio de aplicação das variações requeridas por uma determinado aplicação em um dado contexto.
- Por fim, alguns frameworks apresentam inversão de controle (Inversion of Control - IoC). Inversão de controle, também é conhecida como princípio de Hollywood : "Don't call us, we will call you" (Larman, 2004). Trata-se de transferir o controle da execução da aplicação para o framework, que chama a aplicação em determinados momentos como na ocorrência de um evento. Através da IoC o framework controla quais métodos da aplicação e em que contexto eles serão chamados em decorrência de eventos, como eventos de janela, um pacote, enviado para uma determinada porta, entre outros.



# Desafios decorrentes da utilização de frameworks

- Quando usado em conjunto com padrões de projetos, bibliotecas de classes, componentes, entre outros, frameworks de aplicação têm o potencial para aumentar a qualidade de software e reduzir o esforço de desenvolvimento. Contudo, instituições que tentam construir ou usar frameworks frequentemente falham a menos que resolvam os seguintes desafios: esforço de desenvolvimento, curva de aprendizagem, integração, eficiência, manutenção, validação e remoção de defeitos e falta de padrões.



# Exemplo de framework

- **Bootstrap**

- O Bootstrap é um framework mais direcionado para o aspecto visual das aplicações. Ele tem o poder de encapsular diversas funcionalidades de CSS (Cascading Style Sheets) que, juntas, vão contribuir para a criação de uma página bonita e com funcionalidades padronizadas. Como tem um apelo visual forte, torna-se intuitivo, deixando qualquer desenvolvedor a vontade e seguro no processo de criação.

- É mais indicado para trabalhos no HTML5 e que visam agregar responsividade às interfaces, deixando as páginas adaptáveis a qualquer tamanho de tela de dispositivo. Tudo isso sem comprometer as funcionalidades, a estrutura e o layout do aplicativo.

- Ou seja, a mesma coisa vista ou feita em um celular pode ser retransmitida em um tablet, computador ou monitor de TV. Você define as regras na sua interface e ela vai saber se adaptar automaticamente a esses tamanhos diferentes.



# **SO - Sistemas operacionais**

## **Funcionalidades**

Gestão do processador;

Gestão da memória RAM;

Gestão de IO;

Gestão da execução dos aplicativos;

Gestão dos direitos;

Gestão dos arquivos;

Gestão das informações;



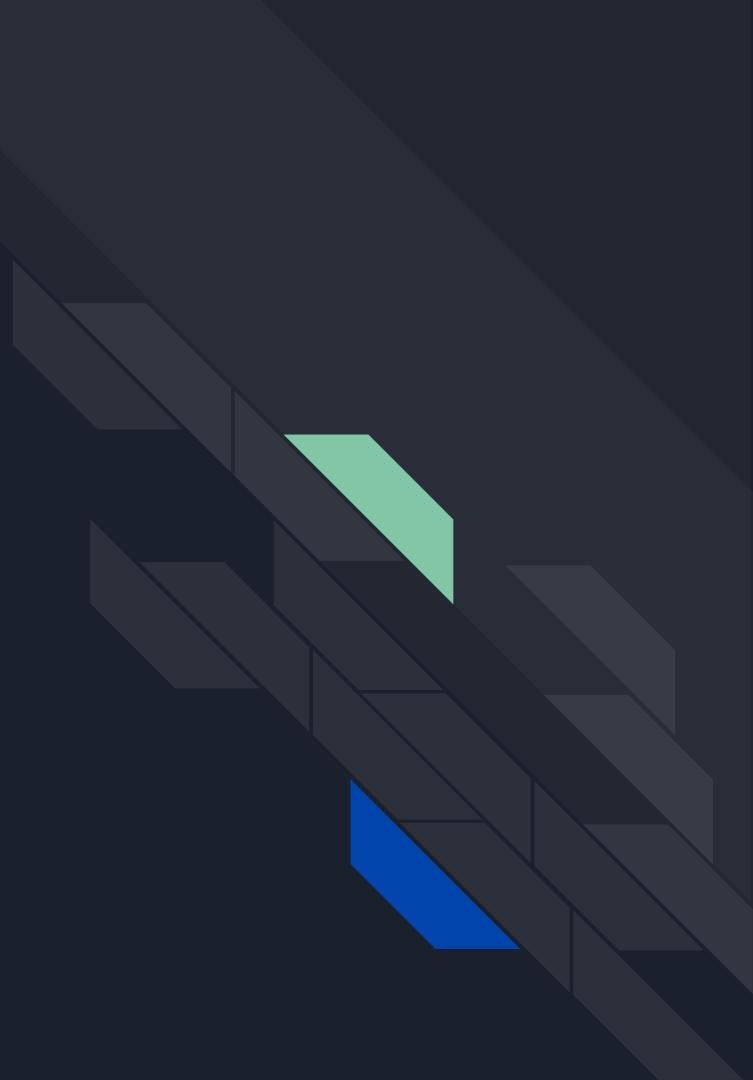
# Firmware

O Firmware é um software de computador que vem junto com Hardware. Ele serve para controlar as configurações do Hardware específico do dispositivo, funções de controle, monitoramento e manipulação de dados. O software do está presente dentro de um microchip instalado dentro Hardware do dispositivo. Sistemas que possuem firmware podem ser encontrados em:

- Semáforos
- Aparelhos Celulares
- Bios, UEFI e EFI
- Placas de vídeos

# SISTEMA

E M B A R C A D O





# CONCEITO

Um sistema embarcado é um sistema de computador que possui uma função dedicada, geralmente operando dentro de um outro sistema.

Sistemas embarcados são usados para controlar muitos tipos de sistemas e equipamentos.



# EXEMPLOS:

Relógios digitais, microondas, televisores, Eletrônica de Aeronaves, Eletrônica Automotiva, Automação residencial, celulares, impressora, MP3, Robótica (desde carrinho de controle remoto à robôs em Marte), vídeo games, controles de vôo, radares, sonares, roteadores, marca-passos.



# CARACTERÍSTICAS

Sistemas embarcados, geralmente, possuem baixo custo de produção, tamanho reduzido, pequeno tempo de resposta, baixo custo energético, utilizam memória ROM e Flash ao invés de disco rígido.

## INTERFACES:

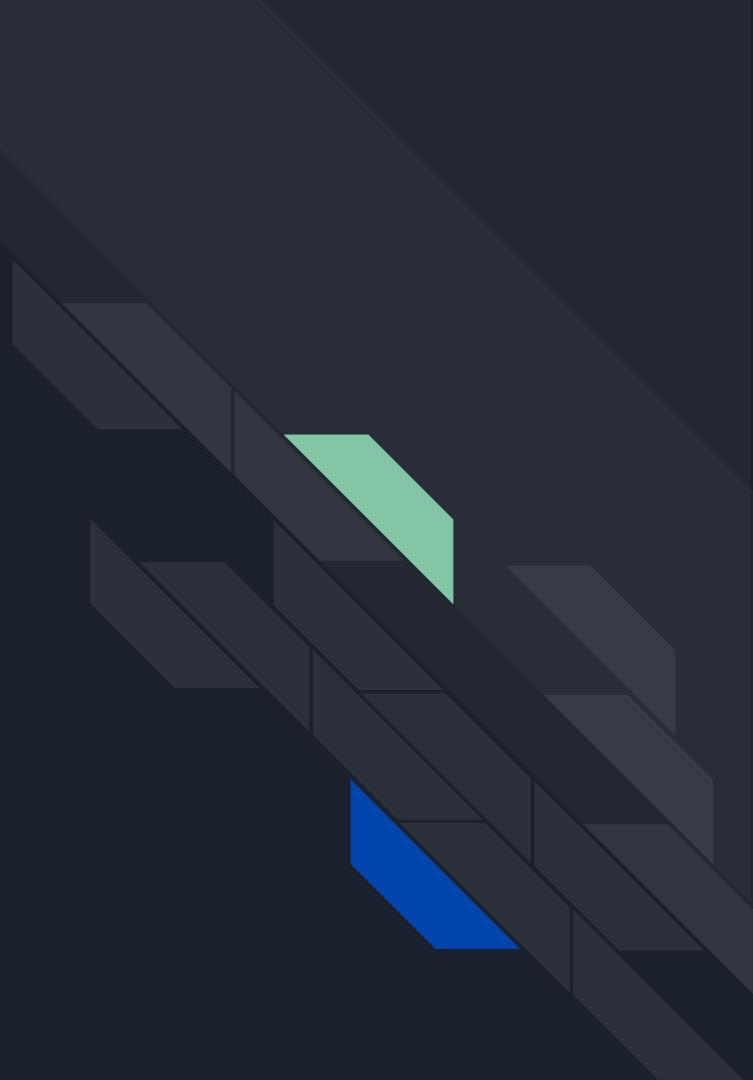
As interfaces normalmente dos sistemas embarcados são bem simples apenas com botões, sons e LCDs, mas alguns programas tem uma interface mais complexa enquanto outros não tem interface.

## PROGRAMAÇÃO:

Para programar um sistema embarcado, normalmente são utilizadas linguagens de programação de uso geral como C, Assembly, C++, Python e em alguns casos até Java.

# ERP

(enterprise resource  
planning)





## Conceito:

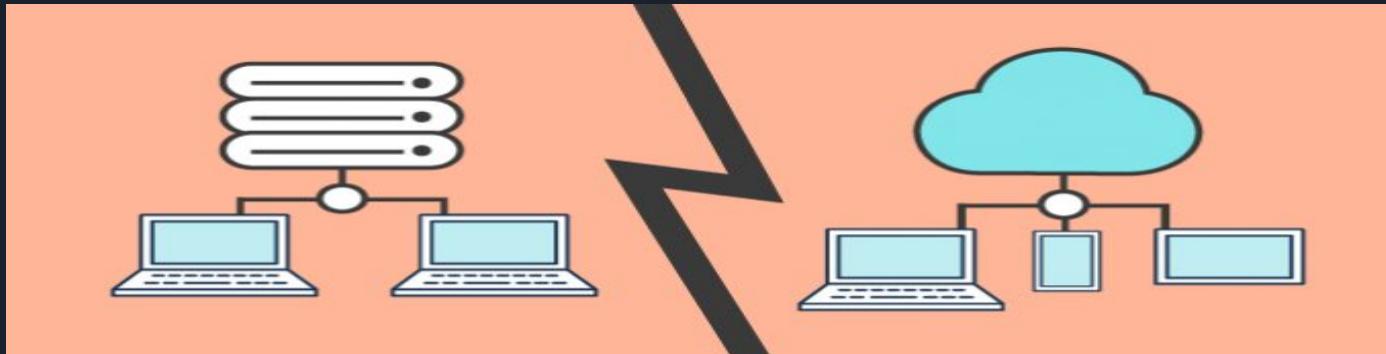
Planejamento de Recursos Empresariais ou planeamento de recurso corporativo é um sistema de informação que integra todos os dados e processos de uma organização em um único sistema. A integração pode ser vista sob a perspectiva funcional e sob a perspectiva sistêmica.



# ERP nuvem e ERP local

O ERP local é instalado no hardware e servidores da sua empresa e geridos por sua equipe de TI.

O ERP na nuvem é fornecido como um serviço de assinatura pela internet.



# *DESENVOLVIMENTO WEB*





# CONCEITO

Desenvolvimento Web é uma profissão responsável pela criação de sistemas ou páginas que podem ser executados direto no navegador através de uma rede, por exemplo web sites.



# ÁREAS DE ATUAÇÃO

O profissional que atua nessa área é chamado de desenvolvedor web suas áreas de atuação são:

Front-End

Back-End



# FRONT-END

É a parte do sistema a qual o usuário pode interagir, as funcionalidades presentes nessa parte são interpretadas pelo browser (no caso de páginas).

Para que esta parte seja desenvolvida é necessário o conhecimento em linguagens de marcação, linguagens de estilo e linguagem de programação.



Linguagens de marcação: definem a estrutura da página organizando onde a informação estará e como será apresentada. Uma linguagem muito usada é a HTML já que é usada para a criação de páginas web.

Linguagens de estilo: definem a parte visual da página, como: cores ou formato de caixas de texto. Uma das mais usadas é a CSS.

Linguagens de programação: definem funções no site que são processadas pelo browser. Uma comumente usada é a JavaScript.



# BACK-END

É a parte responsável pelas funcionalidades do sistema, interpretadas pelo servidor do mesmo.

Gerenciando APIs, Bancos de dados entre outros recurso disponíveis pelo sistema.



# FUNCIONALIDADES PRESENTES NO BACK-END

Servidor: é um software responsável por prover serviços específicos a uma rede de computadores. Exemplos: DNS, FTP e Proxy.

Banco de Dados: armazena dados de mesmo tipo presentes em um sistema, como usuários ou produtos. Para se criar e alterar um banco de dados é necessário o uso de uma ferramenta. Exemplo: MySQL.

API: é a comunicação entre dois ou mais sistemas podendo compartilhar várias coisas, como funções e bancos de



# PONTOS IMPORTANTES NO BACK-END

Escalabilidade: é um método utilizado para deixar o sistema mais flexível para se fazer manutenção ou atualização.

Exemplo de linguagens flexíveis: Python e C++

Segurança: garantir a segurança do sistema é necessário para que os dados do mesmo não sejam alterados por terceiros. Exemplo: HTTPS criptografa a página que está sendo acessada e cria uma segunda chave de acesso assim disponibiliza ao usuário a cópia e guarda a original.