



Resumo - Sistema Operacional e GIT/GITHUB

▼ 1. Sistemas Operacionais

Tipos

- Windows
- Linux
- Mac OS/X
- Android
- IOS

Parte física

- Hardware
- Componentes
- Periféricos

Parte lógica

- Software
- Sistemas Operacionais
- Permite que o usuário controle o hardware através de uma interface

Usuário → Sistema Operacional → Aplicação

Máquina Multinível

- Aplicativos:
 - Editor de texto
 - Teams
 - Pacote Office
- Sistema
 - Sistema Operacional

- Hardware
 - Linguagem de Máquina
 - Micro arquitetura
 - Dispositivos Físicos

Sistemas Operacionais

- Interação textual → Comandos
- Interação gráfica → Interface "Embalagem do comando"
- Computação Ubíqua → que funciona o tempo inteiro

Histórico

- 1ª geração - 1945/55
 - Computadores à válvulas
 - Linguagem de máquina
 - Ausência de Sistema Operacional
- 2ª geração - 1955/65
 - Transistores
 - Assembly
 - S.O do tipo lote
 - Fila de execuções
- 3ª geração - 1965/80
 - Circuitos integrados
 - Sistema OS/360(IBM)
- 4ª geração - 1980/1990
 - Circuito Integrado de Larga Escala
 - Surgimento do DOS e WINDOWS
- 5ª geração - 1990/Atual
 - Difusão da internet
 - Suporte a TCP



2. Tipos de Sistemas Operacionais

- Monotarefa ou monoprogramado
 - Só permite execução de um programa por vez que permanece na memória RAM até o final
 - Tudo fica dedicado à tarefa
 - Ex: MS/DOS
- Multitarefa ou multiprogramado
 - Mantém mais de um programa em execução na memória ao mesmo tempo
 - Sistema em Batch (lote)
 - Não há interação entre o homem e a máquina durante a execução
 - Simula execução em tempo real
 - Tarefas executadas simultaneamente
 - Ex: Linux/Microsoft

▼ 3. Sistemas Operacionais Mobile

- Um smartphone basicamente é um mini computador, pois possui os mesmos componentes, espaço de armazenamento, memória, processador...
- Para utilizar é necessário sistema operacional
- Isso se aplica a dispositivos IoT

Android

- Facilidade de acesso dos usuários
- Licença flexível
 - Cada fabricante pode realizar alterações no código-fonte
- Adaptação aos mais diferentes dispositivos
- Suporta uma grande variedade de tecnologias
- É formado por 5 camadas básicas:

Linux Kernel → Núcleo do Sistema / Drivers

HAL - Hardware Abstract Layer → Ligação da parte física com a próxima camada – câmera, sensores, alto-falante e etc

C/C++

Runtime → Bibliotecas que auxiliam o funcionamento

- Envia as solicitações para a camada anterior

Java API

→ Compartilhador de conteúdo
Gerenciadores: notificações, recursos

System Apps

→ Aplicativos de sistema. Permitem execuções de funções nativas do sistema

IOS

- Evolução para touchscreen
- Aplicativos nativo
- Design minimalista
- Facilidade de uso
- Alterações gradativas na interface
- Não permite alterações
- Não flexível

▼ 4. Filas?

Conceito de filas

Sistemas operacionais utilizam no dia a dia esse conceito

- Velocidade
- Desempenho
- Organização
- Economia de energia

▼ 5. Git e GitHub

Git:

- Versionador

- Facilita no controle de versões
- Evita confusões e multiplicações de arquivos
- Necessário em todo projeto de um desenvolvedor

GitHub:

- Rede Social de desenvolvedores
- Armazena projetos que estão versionados do Git
- É possível criar repositórios (armazenamentos) de projetos públicos e privados

Configuração Técnica:

- Usamos o Git Bash
- Configuração inicial:
 - `git config —global user.name "Cristine"`
 - E-mail: `git config —global user.email "cristineaguiarm@gmail.com"`

• Criando um repositório:

- Criar uma pasta em algum local
- Com o comando `CD` podemos encontrar a pasta que criamos
- Com o comando `git init` podemos iniciar um repositório na página que estamos navegando
- Quando adicionamos um novo arquivo ele não funciona logo de cara, é necessário dar um comando para o git entender:
 - Verificamos primeiramente esse status com: `git status`
 - Commits: são pontos de versionamento
 - Untracked: não traqueado/versionado
 - Para adicionar "Git add"
 - Quando queremos adicionar todos os arquivos da pasta basta adicionar no final `."`
 - Stage: local onde o arquivo é armazenado para ser versionado
 - Para registrarmos uma nova versão usamos o `"git commit -m 'algum comentário'"`

- Se não der o commit novas versões não serão realizadas, mas o Git mostrará que houve uma modificação
- Para adicionar ao Git somente os arquivos novos e modificados podemos usar o comando "Git add *"
- **Configurações GitHub:**
 - É simples criar um novo repositório, basta ir em "Create a new repository"
- **Junção Git e GitHub:**
 - Para unirmos os dois repositórios, primeiramente copiamos a URL do nosso repositório online no GitHub
 - No terminal, na pasta criada para o repositório, usamos o comando: "git remote add origin <https://seuurlgithub.com.br>"
 - Para visualizar: "git remote -v"
 - Para enviar os arquivos, basta usarmos o comando: "git push origin master" → Será solicitado autenticação
 - Assim seu projeto estará linkado ao repositório online
 - Com o comando git clone conseguimos clonar um projeto público
 - Com o comando FORK (dentro do GitHub) também conseguimos ter acesso a um projeto público e utilizarmos ele