

## Projeto Integrador

**27 de Abril de 2024.**

**Discentes:** João Luccas Marques – 2310209,  
Igor Alexander Ribeiro de Paulo -2313308,  
Matheus Marques Portela -2310823,  
Marcos Paulo Moreira Damascena-2310004,  
Pedro Henrique Matias Echebarria-2310055,  
Rebeca Barbosa Ferreira-2310294,  
Victor Duarte Madaleno-2310684  
Victor Manoel de Moraes-2311453

**Nome da disciplina:** Sistemas Operacionais

## Powp Sistemas

### Visão do Projeto

O Powp é um sistema de gerenciamento focado no ramo para pequenas empresas, microempresas e microempreendedores. Tem como objetivo inicial se concentrar no setor financeiro. No entanto, nossa visão de longo prazo envolve a expansão das funcionalidades do site para que ele se torne, no futuro, um Sistema de Planejamento de Recursos Empresariais (ERP). Essa evolução permitirá aos usuários realizar um gerenciamento abrangente de todos os aspectos de suas empresas. Esperamos produzir um sistema onde seja possível validar nossa ideia sobre o projeto com feedbacks constantes do usuário para assim podermos evoluirmos nosso sistema e obter uma direção correta a qual devemos seguir.

### Relatório sobre Gerenciamento de Memória e Ambiente de Desenvolvimento em Sistemas Operacionais

Este relatório aprofunda os tópicos de Gerenciamento de Memória e Ambiente de Desenvolvimento em Sistemas Operacionais.

#### Gerenciamento de Memória:

- **Como os sistemas operacionais gerenciam recursos de memória especificamente**

Os sistemas operacionais empregam diversas técnicas para gerenciar eficientemente os recursos de memória, garantindo que os processos em execução tenham acesso à memória necessária para operar corretamente. As principais técnicas incluem:

**Alocação de Memória:** O sistema operacional aloca memória para processos de acordo com diferentes algoritmos, como:

**Alocação Contígua:** Cada processo recebe um bloco contíguo de memória, facilitando o acesso endereçável.

**Alocação Não Contígua:** A memória é dividida em blocos de tamanhos variáveis, e os processos são alocados em blocos dispersos.

**Paginação:** A memória física e a memória virtual são divididas em páginas de tamanho fixo, permitindo alocação flexível e fragmentação reduzida.

**Gerenciamento de Espaço Livre:** O sistema operacional mantém um registro dos espaços livres na memória, permitindo a alocação eficiente de novas solicitações de memória.

**Substituição de Processos:** Quando a memória física fica saturada, o sistema operacional remove processos inativos da memória principal e os armazena na memória secundária (geralmente disco) para liberar espaço para outros processos.

**Memória Virtual:** A memória virtual cria a ilusão de um espaço de memória maior do que a memória física real, permitindo a execução de mais processos do que caberiam na memória física ao mesmo tempo.

- **Qual é o papel do escalonamento de processos no desenvolvimento de aplicações**

O escalonamento de processos é um componente crucial do gerenciamento de processos, determinando a ordem em que os processos recebem acesso à CPU. Sua importância no desenvolvimento de aplicações reside em:

**Otimização do Desempenho:** O escalonamento eficiente garante que os processos críticos recebam tempo de CPU adequado, otimizando o desempenho geral do sistema.

**Tempo de Resposta Rápido:** O escalonamento adequado garante que os processos interativos recebam tempo de CPU com prioridade, proporcionando um tempo de resposta rápido para os usuários.

**Imparcialidade:** O escalonamento justo garante que todos os processos recebam tempo de CPU de acordo com suas necessidades, evitando que um único processo monopolize a CPU.

#### **Ambiente de Desenvolvimento:**

- **Como os sistemas operacionais fornecem um ambiente de desenvolvimento para programadores**

Os sistemas operacionais fornecem um ambiente de desenvolvimento completo para programadores, incluindo:

**Interface de Linha de Comando (CLI):** Permite a interação com o sistema e a execução de comandos para tarefas como compilação, depuração e execução de código.

**Interpretadores de Linguagem:** Interpretam e executam código escrito em linguagens de programação de alto nível, como Python ou Java.

**Compiladores:** Traduzem código escrito em linguagens de alto nível para linguagem de máquina, permitindo a execução em hardware específico.

**Depuradores:** Permitem a análise e identificação de erros no código durante o desenvolvimento.

**Bibliotecas e Ferramentas:** Fornecem funcionalidades pré-construídas para auxiliar no desenvolvimento, como bibliotecas para manipulação de arquivos, redes e interfaces gráficas.

- **Quais ferramentas e utilitários estão disponíveis para depuração, compilação e execução de código**

Uma vasta gama de ferramentas e utilitários está disponível para auxiliar no desenvolvimento de software, incluindo:

**Depuradores:** GDB, LLDB, Valgrind.

**Compiladores:** GCC, Clang, Visual Studio C++.

**Interpretadores:** Python, Java Interpreter, C Shell.

**Bibliotecas:** Standard Library (C/C++), NumPy (Python), Boost (C++).

**IDEs (Integrated Development Environments):** Visual Studio, Eclipse, IntelliJ IDEA.

## **Conclusão**

O gerenciamento de memória e o ambiente de desenvolvimento são aspectos fundamentais dos sistemas operacionais, fornecendo recursos essenciais para a execução de programas e o desenvolvimento de software. A escolha das técnicas de gerenciamento de memória e das ferramentas de desenvolvimento varia de acordo com as necessidades específicas do projeto e do sistema operacional utilizado.

É importante ter conhecimento das diferentes técnicas e ferramentas disponíveis para otimizar o desempenho e a produtividade no desenvolvimento de software.