

Projeto Dashboard

Versão 0.3

Autor

Prof. Dr. Luiz Nacamura Junior

Adaptado por

Prof. Dr. Marco Aurélio Wehrmeister

INTRODUÇÃO

O projeto Dashboard visa desenvolver um aplicativo que permita visualizar as propriedades dos Sistemas Operacionais: Hardware do Sistemas, características dos processos em execução, sistemas de Diretórios, etc.

Na Wikipedia, encontra-se a definição de um dashboard como sendo “uma tela, composta de uma ou mais camadas, sob a forma de um painel, com instrumentos virtuais (knobs, mostradores, gráficos de barra) onde se associam variáveis a serem monitoradas além de gráficos que mostram a evolução de variáveis, por exemplo, no tempo. ... Os Scorecards fornecem uma representação visual dos indicadores chave de desempenho (KPIs) – indicadores cuidadosamente selecionados que ajudam as empresas a medir e gerenciar o desempenho.” A figura abaixo ilustra um exemplo de tela de um dashboard.



Exemplo de Dashboard [Fonte: <https://www.opservices.com.br/o-que-e-um-dashboard/>]

PROJETO A – Implementação da Funcionalidade Inicial do Dashboard

Objetivos

1. Monitorar e apresentar as características e propriedades de todos os processos existentes em execução no sistema operacional;
2. Monitorar e apresentar as informações do uso de memória dos processos;

Requisitos Gerais

Todos os requisitos abaixo devem ser obrigatoriamente obedecidos em ambos os projetos:

1. As informações atuais de cada processo devem ser apresentas ao usuário e **atualizadas em intervalos regulares de tempo**, p. ex. a cada 5 segundos.
2. Quantas e quais informações a serem mostradas ao usuário ficam a critério da equipe. Escolha as informações que considerar mais pertinente e relevante a um administrador de sistemas. A completude e complexidade das informações apresentadas serão objeto da avaliação; quanto mais informações coletadas e apresentadas, melhor será a avaliação do projeto.
3. O Dashboard deve permitir mostrar as **informações globais do sistema**, assim como **informações individualizadas para cada processo**. A ideia é trazer a mesma informação que o monitor de sistema do Windows ou Linux mostra para o usuário.
4. A informação apresentada deve ser “processada” pelo Dashboard. **Não deve ser apresentada a saída texto “crua”/raw** retornada pelo serviço, função, API ou comando do sistema operacional. Em muitos casos, mostrar somente a saída texto de uma função ou programa não é suficiente para demonstrar que o conhecimento necessário foi adquirido pelos alunos.
5. As informações a serem apresentadas no Dashboard não devem ser obtidas através de comandos do shell (como p.ex. os comandos ls, ps, du, etc.). Ao invés disso, **utilize as chamadas de sistema ou outro meio fornecido pelo sistema operacional** para obtenção de tais informações (p.ex. diretório /proc no Linux, tabela de partições do disk, etc.).
6. A linguagem a ser utilizada para implementação do Dashboard é livre, no entanto, a obtenção das informações dos processos deve ser feita via API do sistema operacional. **Não será aceita qualquer implementação que use apenas as funcionalidades das bibliotecas (de sistema) da linguagem**, p.ex. sys module do Python.

7. O Dashboard deve ser **implementado como um software multitarefa**. Sugestão: usar **Threads**; separar os processos de aquisição de dados, processamento desses dados, e apresentação dos resultados. Siga o padrão de projeto MVC (*Model-View-Controller*). **Não serão aceitas implementações que não sejam efetivamente implementadas como um software multitarefa.**

1. Monitorar e apresentar as características e propriedades de todos os processos existentes em execução no sistema operacional;

Este objetivo consiste em implementar o primeiro conjunto de funcionalidades do Dashboard visando apresentar as informações sobre os processos que existem e estão executando no sistema operacional. Os requisitos abaixo devem ser seguidos:

1. Mostrar dados globais do uso do processador, p.ex. percentual de uso do processador, percentual de tempo ocioso, quantidade total de processos e threads, etc.
2. Mostrar a lista de processos existentes juntamente com os respectivos usuários;
3. Mostrar informações sobre os threads de cada processo;
4. Mostrar informações detalhadas de cada processo. Nesse caso, as informações podem ser apresentadas em uma nova tela (ou aba) que, ao ser fechada, retorna a tela principal.

2. Monitorar e apresentar as informações do uso de memória dos processos;

Este objetivo consiste em fornecer as informações de uso de memória global do sistema e de cada processo individualmente. Os requisitos abaixo devem ser obrigatoriamente seguidos:

1. Mostrar dados globais do uso de memória do sistema, p.ex. percentual de uso da memória, percentual de memória livre, quantidade de memória física (RAM) e virtual, etc.
2. Mostrar informações detalhadas sobre o uso de memória para cada processo, p.ex. quantidade total de memória alocada, quantidade de páginas de memória (total, de código, heap, stack), etc.
3. Ao mostrar informações detalhadas de cada processo, as informações podem ser apresentadas em uma nova tela (ou aba) que, ao ser fechada, retorna a tela principal.

PROJETO B – Mostrar dados do uso dos dispositivos de E/S pelos processos

Objetivos e Requisitos

1. Monitorar e apresentar o sistema de arquivos.
2. Monitorar e apresentar as informações do uso dos arquivos e dispositivos de entrada/saída pelos processos que estão em execução.

Requisitos Gerais

Todos os requisitos abaixo devem ser obrigatoriamente obedecidos em ambos os projetos:

1. As informações atuais de cada processo devem ser apresentadas ao usuário e atualizadas em intervalos regulares de tempo, p. ex. a cada 5 segundos.
2. Quantas e quais informações a serem mostradas ao usuário ficam a critério da equipe. Escolha as informações que considerar mais pertinente e relevante a um administrador de sistemas. A completude e complexidade das informações apresentadas serão objeto da avaliação; quanto mais informações coletadas e apresentadas, melhor será a avaliação do projeto.
3. O Dashboard deve permitir mostrar as informações globais do sistema, assim como informações individualizadas para cada processo. A ideia é trazer a mesma informação que o monitor de sistema do Windows ou Linux mostra para o usuário.
4. A informação apresentada deve ser “processada” pelo Dashboard. Não deve ser apresentada a saída texto “crua”/raw retornada pelo serviço, função, API ou comando do sistema operacional.
5. As informações a serem apresentadas no Dashboard não devem ser obtidas através de comandos do shell (como p.ex. os comandos ls, ps, du, etc.). Ao invés disso, utilize as chamadas de sistema ou outro meio fornecido pelo sistema operacional para obtenção de tais informações (p.ex. diretório /proc no Linux, tabela de partições do disk, etc.).
6. A linguagem a ser utilizada para implementação do Dashboard é livre, no entanto, a obtenção das informações dos processos deve ser feita via API do sistema operacional. **Não será aceita qualquer implementação que use apenas as bibliotecas de sistema da linguagem**, p.ex. sys module do Python.

7. O Dashboard deve ser implementado como um software multitarefa. Sugestão: usar Threads; separar os processos de aquisição de dados do processamento e apresentação dos resultados. **Não serão aceitas implementações que não sejam efetivamente implementadas como um software multitarefa.**

1. Monitorar e apresentar o sistema de arquivos.

Este objetivo consiste em fornecer as informações gerais do sistema de arquivos. Os requisitos abaixo devem ser obrigatoriamente seguidos:

1. Mostrar as informações atuais do sistema de arquivos devem ser apresentadas, p.ex. partições, a quantidade de memória usada em cada partição, o tamanho da partição, percentual de uso, etc.
2. Permitir a navegação na árvore de diretórios, a partir da raiz.
3. Listar os arquivos contidos em um diretório, juntamente com os atributos de cada arquivo (p.ex. nome, tamanho, permissões, etc.)

2. Monitorar e apresentar as informações do uso dos arquivos e dispositivos de entrada/saída pelos processos.

Este objetivo consiste em fornecer as informações individuais dos recursos alocados para cada processo. Os requisitos abaixo devem ser obrigatoriamente seguidos:

4. Mostrar as informações atuais dos recursos abertos/alocados pelo processo, p.ex. arquivos abertos, semáforos/mutexes, sockets, etc.
5. Ao mostrar informações detalhadas de cada processo, as informações podem ser apresentadas em uma nova tela (ou aba) que, ao ser fechada, retorna a tela principal.