

Grupo Projectx - Gerenciador de Desempenho Distribuído

Integrantes: Eduardo Santos Santana (202203503), Bruno Lopes Santos (201907448), Yatherson Lucas Teodoro Souza (201912485), Reynner Miranda Nunes (201907501), Gabriel Silva Miranda (201907464)

- Sistemas Distribuídos UFG 2024/2 - SERGIO TEIXEIRA DE CARVALHO

Vamos refletir no seguinte cenário...

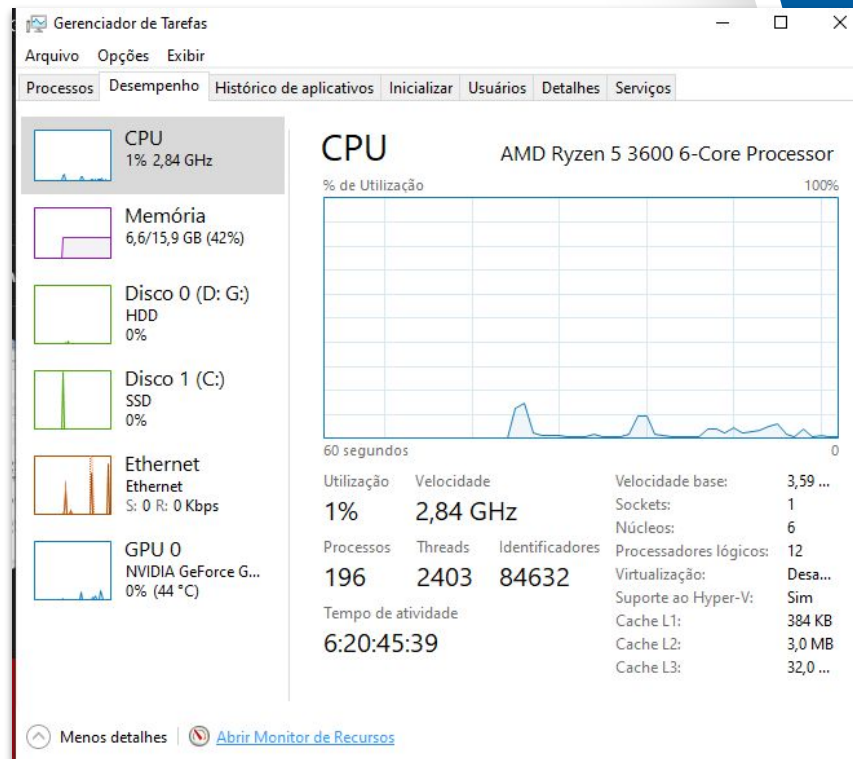


Empresas Nacionais enfrentando **dificuldades** em **monitorar o desempenho** de várias máquinas geograficamente **dispersas**.

- ▶ **História hipotética** : Joãozinho, gerente de TI na empresa Central em São Paulo, estava tendo um dia tranquilo até que, de repente, começou a receber várias ligações de emergência. Uma filial da empresa, localizada em outra cidade, havia sofrido um possível incêndio, supostamente causado por superaquecimento de um dos servidores.

Solução ?!

- Um **Gerenciador de Desempenho Distribuído** centraliza dados de desempenho (CPU, memória, rede) de múltiplos computadores, agregando-os em um único ponto para análise e monitoramento em tempo real.



Exemplo de ideia do protótipo de tela

Ideia de arquitetura inicial do projeto



Frontend: Interface acessada pelos clientes, consome dados de métricas da API Gateway e constrói gráficos dinâmicos

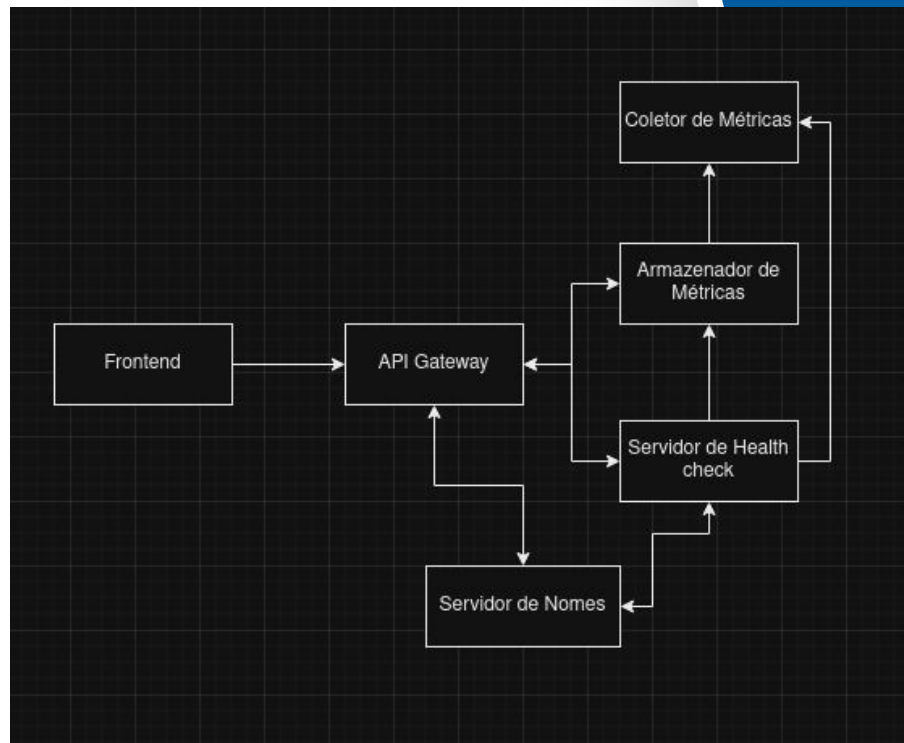
API Gateway: Faz o proxy das requisições do frontend para os serviços do backend, mascarando o funcionamento interno do backend.

Servidor de Nomes: Mantém listas de nomes e IPs dos serviços, consultado por outros serviços como fonte de verdade para IPs, e relacionado ao serviço de health check para manter os IPs funcionais atualizados.

Servidor de Health Check: Verifica a conectividade dos serviços do backend e informa o servidor de nomes quando um IP registrado não está funcionando para manter a lista atualizada.

Armazenador de Métricas: Colhe, agrupa e armazena métricas, realizando a coleta de forma ativa através de scraping, acessando coletores de métricas periodicamente.

Coletor de Métricas: Instalado nas máquinas monitoradas, coleta e expõe métricas do sistema (uso de memória, CPU, etc.), atualizando-as periodicamente.





Conteúdo que será Explorado

Programa

Unidade I

1. Introdução a Sistemas Distribuídos: conceitos básicos; objetivos de projeto; tipos de sistemas distribuídos; infraestrutura de sistemas distribuídos.
2. Arquitetura de sistemas distribuídos: cliente-servidor; peer-to-peer; objetos distribuídos; arquiteturas baseadas em serviços; arquiteturas baseadas em eventos; middleware.

Unidade II

3. Comunicação em sistemas distribuídos: invocação remota (RPC, RMI), interface de serviços Web (RESTful, gRPC); comunicação assíncrona, filas de mensagens e sistemas publish-subscribe.
4. Sistemas distribuídos robustos: coordenação; replicação e consistência; tolerância a falhas.

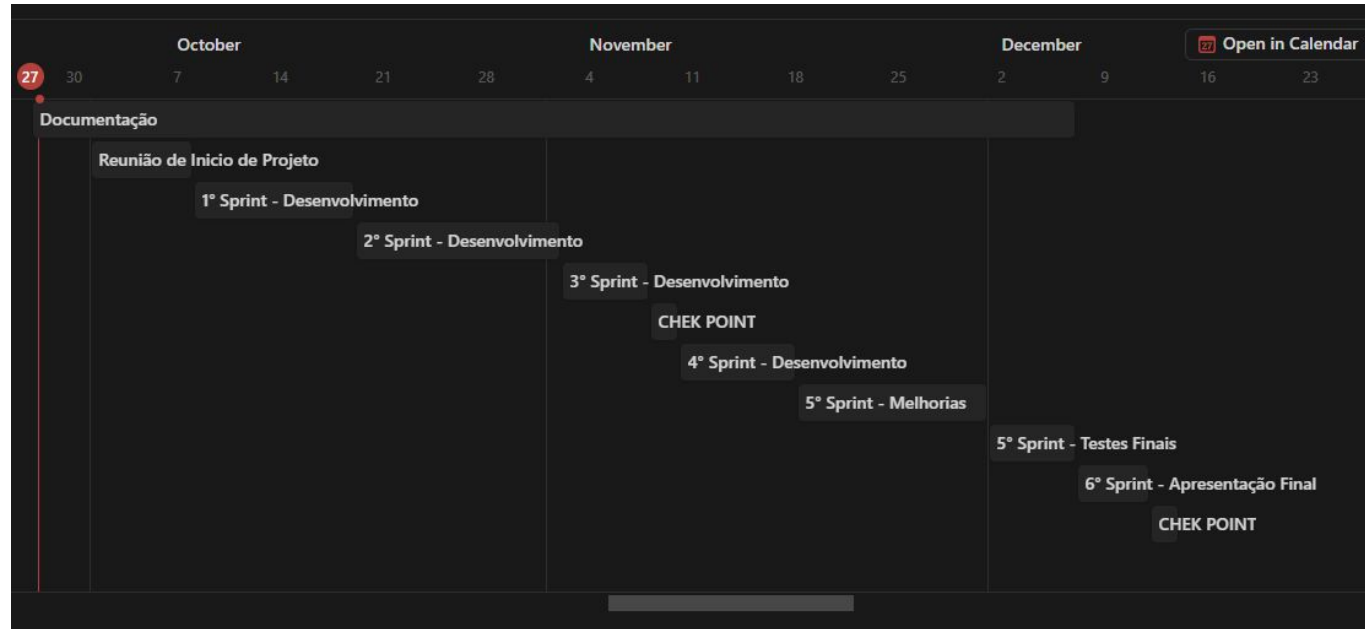
Unidade III

5. Computação móvel e ubíqua: computação móvel e na nuvem; offloading de tarefas; ambiente IoT; sistemas ubíquos cientes de contexto.
6. Tópicos emergentes: segurança em sistemas distribuídos; streaming de mídia; blockchain e consenso distribuído.

API GATEWAY - interface de serviços web que gerencia as comunicações utilizando protocolo RESTful

Cronograma de Execução

- ▶ Linguagem: c#, python, javascript, node, lua
- ▶ Plataforma será GitHub
- ▶ Documentação: Bruno
- ▶ Reunião: Todos
- ▶ Apresentação: Bruno
- ▶ Testes: Bruno
- ▶ 1° Sprint: Yatherson
- ▶ 2° Sprint: Eduardo
- ▶ 3° Sprint: Reydner
- ▶ 4° Sprint: Gabriel
- ▶ 5° Sprint: Todos



Obrigado!

Dúvidas ou sugestões?

