

Is Down ?

Documento de Arquitetura de Software

Versão 0.2

Is Down ?	Version: 0.3
Documento de Arquitetura de Software	Date: 24/01/2022
Is Down Doc V.0.3	

Histórico da Revisão

Data	Versão	Descrição	Autor
09/12/2021	0.1	Início do desenvolvimento da Introdução e Metas e Restrições da Arquitetura	Gabriel Inácio
21/01/2022	0.2	Modificação na descrição da arquitetura do sistema e desenvolvimento dos tópicos	Gabriel Inácio
24/01/2022	0.3	Revisão dos 6 primeiros tópicos e desenvolvimento dos próximos	Gabriel Inácio

Is Down ?	Version: 0.3
Documento de Arquitetura de Software	Date: 24/01/2022
Is Down Doc V.0.3	

Índice Analítico

1. Introdução	4
1.1 Finalidade	4
2. Metas e Restrições da Arquitetura	5
3. Suposições e Dependências	6
4. Requisitos Arquiteturalmente Significantes	7
5. Decisões, Restrições e justificativas	8
6. Mecanismos Arquiteturais	9
7. Camadas da Arquitetura	10
8. Visões da Arquitetura	11
9. Qualidade	12

Is Down ?	Version: 0.3
Documento de Arquitetura de Software	Date: 24/01/2022
Is Down Doc V.0.3	

Documento de Arquitetura de Software

1. Introdução

Esse documento tem o objetivo de definir a visão arquitetural de um sistema de monitoramento de serviços da internet de nome “Is Down ?”.

1.1 Finalidade

Este documento oferece uma visão geral arquitetural abrangente do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.

1.2 Escopo

As principais características do sistema e as decisões que levaram a definição da arquitetura da solução estão detalhadas nesse documento, de forma a definir uma base sólida para a equipe de desenvolvedores do projeto.

1.3 Termos

A tabela a seguir define os termos usados ao longo do documento e seu significado.

Termo	Significado
SGBD	Acrônimo de Sistema de Gerenciamento de Banco de Dados.
API	Acrônimo de <i>Application Programming Interface</i> .
<i>Query</i>	Palavra “Consulta” em inglês.
<i>Back End</i>	Parte da aplicação que não é diretamente acessada pelo usuário. Tipicamente responsável pelo acesso ao banco de dados.
<i>Front End</i>	Parte da aplicação que o usuário usa diretamente.
<i>Twitter</i>	Twitter é uma rede social e um serviço de microblog.

2. Metas e Restrições da Arquitetura

2.1 Motivação

Diversos serviços da internet, como redes sociais e correios eletrônicos, sofrem com períodos de instabilidade e inoperabilidade. Assim, é muito comum que usuários dessas aplicações não saibam se o problema ocorre exclusivamente com ele – podendo ser um problema com o dispositivo e/ou sua conexão com a internet – ou é um problema do serviço.

Além disso, existem também os usuários corporativos, que dependem parcialmente ou totalmente de um ou mais desses serviços. Logo, é fundamental que saibam quais serviços são mais estáveis.

Portanto, com o objetivo de sanar essa dor, o “Is Down ?” tem o objetivo de monitorar os principais serviços da internet e oferecer ferramentas capazes de mitigar o impacto que problemas nesses serviços podem causar.

Is Down ?	Version: 0.3
Documento de Arquitetura de Software	Date: 24/01/2022
Is Down Doc V.0.3	

2.2 Metas e Restrições

2.2.1 A solução deve ser robusta

Como o objetivo da aplicação é monitorar os serviços e ajudar nossos usuários quando um problema ocorrer, a solução deve sempre estar disponível. Assim, deverão ser adotadas medidas para que isso seja possível.

2.2.2 A coleta de dados deverá ser regular e assíncrona

Um período de instabilidade pode ocorrer a qualquer momento, logo os serviços deverão ser monitorados regularmente e ações automáticas disparadas sempre que um problema for detectado.

2.2.3 Deverão ser usadas mais de uma fonte de dados

A solução deverá utilizar diversas fontes de dados para coletar reclamações de usuários. A dependência de uma única fonte de dados poderá comprometer a disponibilidade do sistema e a confiabilidade dos dados.

2.2.4 A solução poderá ser integrada a outros sistemas via API

Um dos objetivos do "Is Down ?" é permitir que outros sistemas utilizem dos dados coletados por nossa solução.

3. Suposições e Dependências

3.1 Introdução

Em uma pesquisa realizada pela nossa equipe em busca de soluções que realizam o monitoramento de serviços da internet e o comportamento de usuários em redes sociais, foi definido que o sistema usará APIs para analisar comentários e publicações em redes sociais para identificar e registrar reclamações.

3.2 API do Twitter

A principal fonte de dados para monitoração dos serviços será a API do *Twitter*. Para usá-la, o sistema deverá ser capaz de realizar requisições HTTP e tratamento de objetos JSON.

4. Requisitos Arquiteturalmente Significantes

4.1 Requisitos Funcionais

RF1 - O usuário externo deverá obter um gráfico com o histórico do comportamento dos serviços monitorados.

RF2 - O usuário externo deverá ser informado do status dos serviços por meio de tags.

RF3 - O usuário externo deverá ter acesso a um catálogo dos serviços monitorados.

RF4 - O usuário externo poderá se cadastrar para ter acesso a privilégios de usuário interno.

RF5 - O usuário interno poderá solicitar que um novo serviço seja monitorado.

RF6 - O administrador poderá cadastrar, editar e excluir serviços.

RF7 - O administrador poderá cadastrar, editar e excluir administradores.

RF8 - O sistema deverá ter uma função de notificar usuários internos de problemas nos serviços monitorados.

4.2 Requisitos Não Funcionais

RNF1 - O sistema deverá usar o SGBD PostgreSQL.

RNF2 - O sistema deverá usar o framework Hibernate.

RNF3 - O sistema deverá estar disponível de 9 h às 18 h (Horário comercial).

RNF4 - A interface web deverá ser responsiva.

RNF5 - O monitoramento dos serviços deverá manter um mínimo de 48 horas de histórico.

Is Down ?	Version: 0.3
Documento de Arquitetura de Software	Date: 24/01/2022
Is Down Doc V.0.3	

5. Decisões, Restrições e Justificativas

5.1 O sistema deverá usar mais de uma representação do mesmo objeto

O objetivo da solução é monitorar serviços da internet e possibilitar o acesso dos dados coletados por duas formas: Interface Web e API. Assim, o objeto que será usado para persistência de dados será diferente do objeto retornado pela API, mesmo que representem o mesmo objeto abstrato.

5.2 Abstração do acesso a base de dados

As classes usadas pela Interface Web e API deverão usar classes que representem os registros da base de dados. Em nenhuma circunstância essas classes poderão fazer *queries* diretamente a base de dados.

O objetivo dessa restrição é diminuir o acoplamento de camadas usadas diretamente pelos usuários da camada de persistência de dados.

6. Mecanismos Arquiteturais

6.1 Correio

Uma das funcionalidades críticas da ferramenta é poder enviar mensagens aos usuários de maneira automática.

- **Latência:** O intervalo de tempo entre um evento e a ação automática de envio de mensagens deve ser otimizado.
- **Confiabilidade:** O número de mensagens que não chegam aos usuários deve ser o mínimo possível.
- **Volume:** O sistema deverá suportar um grande número de mensagens enviadas simultaneamente.

6.2 Gestão de recursos

A solução deve armazenar uma grande quantidade de dados sobre os serviços monitorados por longos períodos de tempo, assim, deverão ser utilizados recursos para mitigar o uso excessivo da base de dados.

- **Manutenção:** Dados antigos devem ser excluídos ou consolidados regularmente.

6.3 Gráficos

Deverão ser utilizados gráficos como recurso de apresentação dos dados.

- **Confiabilidade:** Os gráficos devem apresentar de maneira clara e não tendenciosa os dados coletados.

6.4 Gestão de eventos

O sistema deverá lidar com a coleta de dados assíncrona, envio de mensagens automáticas e acesso de usuários via interface web e API.

- **Paralelismo:** Atividades de coleta de dados e envio de mensagens deverão ser realizadas de maneira paralela.

Is Down ?	Version: 0.3
Documento de Arquitetura de Software	Date: 24/01/2022
Is Down Doc V.0.3	

7. Camadas da Arquitetura

[Descreva os padrões de arquitetura utilizados e como a arquitetura se manterá consistente e uniforme. Isto pode ser uma simples referência para um conhecido padrão arquitetural, como o padrão de divisão em camadas e uma descrição de como os componentes do sistema podem ser colocados juntos.]

Is Down ?	Version: 0.3
Documento de Arquitetura de Software	Date: 24/01/2022
Is Down Doc V.0.3	

8. Visões da Arquitetura

[Descreva as visões arquiteturais usadas para descrever a arquitetura. Isto ilustra as diferentes perspectivas disponíveis para rever e documentar as decisões arquiteturais.]

Visões Recomendadas:

- **Lógica:** Descreva a estrutura e comportamento de porções arquiteturalmente significantes do sistema. Isto deve incluir a estrutura de pacotes, interfaces críticas, importantes classes e subsistemas e as relações entre estes elementos. Isto também inclui visões físicas e lógicas dos dados persistentes.
- **Operacional:** Descreva os nós físicos do sistema e os processos, threads e componentes que rodam em cada um desses nós. Esta visão não é necessária se o sistema roda num único processo e num único thread.
- **Casos de Uso:** Uma lista ou diagrama dos casos de uso que contém requisitos arquiteturalmente relevantes.

Is Down ?	Version: 0.3
Documento de Arquitetura de Software	Date: 24/01/2022
Is Down Doc V.0.3	

9. Qualidade

[Uma descrição de como a arquitetura do software contribui para todos os recursos (exceto a funcionalidade) do sistema: extensibilidade, confiabilidade, portabilidade e assim por diante. Se essas características possuírem significado especial, como implicações de segurança, garantia ou privacidade, elas deverão ser delineadas claramente.]