**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-graduação *Lato Sensu* em Engenharia de *Software***

Projeto Integrado

Relatório Técnico

HTTP Cache com NodeJS e Redis

Alexsander da Silva Loula

Belo Horizonte

Junho de 2023.

# Projeto Integrado

**Sumário**

Projeto Integrado 3

1. Cronograma de Trabalho 4

2. Introdução 5

3. Definição Conceitual da Solução 6

3.1 Diagrama de Casos de Uso 6

3.2 Requisitos Funcionais 6

3.3 Requisitos Não-funcionais 7

4. Protótipo Navegável do Sistema 7

5. Diagrama de Classes de Domínio 8

6. Arquitetura da Solução 8

6.1 Padrão Arquitetural 8

6.2 C4 model - Diagrama de Contexto 9

7. Frameworks de Trabalho 10

8. Estrutura Base do Front End 10

9. Modelo Relacional ou Projeto de Banco de Dados NoSQL 10

10. Plano de Testes 11

11. Apropriação de Horas no Projeto 11

12. Código da Aplicação 12

13. Avaliação Retrospectiva 12

13.1 Objetivos Estimados 12

13.2 Objetivos Alcançados 12

13.2 Lições aprendidas 12

14. Referências 13

## Cronograma de Trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 19/06/2023 | 21/06/2023 | 1. Revisão do conteúdo sobre o Projeto Integrado. | Entendimento do escopo do Projeto Integrado |
| 22/06/2023 | 23/06/2023 | 2. Relatório Técnico: Objetivos do Trabalho, Apresentação do Problema, Descrição Geral do Sistema. | Preenchimento do Relatório Técnico – Descritivos. |
| 26/06/2023 | 26/06/2023 | 3. Relatório Técnico: Identificação dos atores, requisitos funcionais e não funcionais. | Preenchimento do Relatório Técnico – Definição Conceitual. |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 4. Relatório Técnico: Identificação da ferramenta e desenvolvimento do diagrama de Caso de Uso. | Preenchimento do Relatório Técnico – Diagrama de Caso de Uso. |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 5. Relatório Técnico: Diagrama de Solução e Sequência. | Preenchimento do Relatório Técnico – Diagrama de Solução e Sequência. |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 8. Relatório Técnico: Definição do Padrão Arquitetural do Projeto e Tecnologias. | Preenchimento do Relatório Técnico – Descrição da Arquitetura escolhida e tecnologias utilizadas. |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 9. Relatório Técnico: Criação do Diagrama de Contexto do Projeto e uma breve explicação. | Confecção do Relatório Técnico – Diagrama de Contexto (C4 Model) e explicação. |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 10. Relatório Técnico: Apresentar Frameworks e Estrutura Base do Front End. | Confecção do Relatório Técnico – Descrição dos Frameworks e visualização do layout do front end. |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 11. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 12. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 13. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 14. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 15. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 16. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 17. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 18. |  |
|  |  |  |  |
|  |  |  |  |

**Observação: acrescente ou retire linhas, caso seja necessário.**

## Introdução

***2.1. Objetivos do trabalho***

O objetivo deste trabalho é apresentar a descrição e desenvolvimento de uma aplicação (serviço) para Cache de requisições HTTP (*Hypertext Transfer Protocol)*. O serviço faz a interface entre o Cliente e o Servidor, interceptando as requisições e armazenando em um Cache em Memória.

O serviço utiliza design minimalista e com configurações feitas diretamente em um arquivo JSON, podendo rodar em infraestrutura *Cloud*, *On-premise* ou mesmo em hardware de baixo desempenho (SBC’s - *Single Board Computers*) como Raspberry Pi.

***2.2. Apresentação do problema***

O HTTP é um Protocolo de Transferência de Hipertexto e fica na camada de aplicação OSI. Ele é um dos principais protocolos usados em redes baseadas em IP para a transmissão de dados. Esse protocolo é a base de qualquer troca de dados na Web e é um protocolo cliente-servidor, o que significa que as solicitações são iniciadas pelo destinatário, geralmente o navegador da Web ou outro cliente. Clientes e servidores se comunicam trocando mensagens individuais.

Por ser um dos protocolos mais usados para comunicação entre computadores é importante otimizarmos sua utilização para melhor aproveitamento de recursos computacionais e consequentemente redução de custos de operação.

O Cache é um recurso de armazenamento temporário para reuso de recursos utilizados com frequência, como por exemplo uma página Web. A utilização desse recurso tornou-se essencial com o aumento do uso da Internet e a maior exigência dos usuários em relação ao desempenho de sistemas.

***2.2. Descrição geral do serviço***

O desenvolvimento do Serviço de Cache HTTP tem por objetivo fornecer uma camada de persistência temporária em memória, aumentando a velocidade de troca de mensagens entre Cliente e Servidor.

Além da melhoria na experiência do usuário que está utilizando o serviço, temos também uma redução de tráfego e custo com infraestrutura.

## Definição Conceitual da Solução

O serviço foi desenvolvido pensando no minimalismo e desempenho, permitindo o uso em hardware de baixo-custo.

## Diagrama de Casos de Uso

Segue abaixo o Diagrama de Caso de Uso do Serviço de Cache HTTP:

|  |
| --- |
|  |

## Requisitos Funcionais

Abaixo seguem listados os Requisitos Funcionais (RF) do serviço:

* RF01: O usuário (Admin) deve ser capaz de configurar a porta de entrada do serviço para conexão com o cliente;
* RF02: O usuário (Admin) deve ser capaz de configurar a porta de saída do para conexão com o serviço externo (API);
* RF03: O usuário (Admin) deve ser capaz de configurar o host do servidor de cache em memória (Redis);
* RF04: O usuário (Admin) deve ser capaz de configurar o tempo de expiração dos dados armazenados no cache em memória (Redis);
* RF05: O usuário (Admin) deve ser capaz de habilitar ou desabilitar o serviço de cache;
* RF06: O usuário (Admin) deve ser capaz de habilitar ou desabilitar mensagens de log no console do serviço.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| RF01 | Configurar porta de entrada do serviço | M | A |
| RF02 | Configurar porta de saída do serviço | M | A |
| RF03 | Configurar host do servidor do Redis | B | M |
| RF04 | Configurar o tempo de expiração do cache | A | M |
| RF05 | Habilitar ou Desabilitar o serviço de cache | M | B |
| RF06 | Habilitar ou Desabilitar mensagens de log no console | B | M |

\* B = Baixa, M = Média, A = Alta.

## Requisitos Não-funcionais

1. Abaixo seguem listados os Requisitos Não-Funcionais (RNF) do serviço:

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| RNF01 | O serviço deve rodar em infraestrutura Cloud | A |
| RNF02 | O serviço deve rodar em infraestrutura On-Premise | B |
| RNF03 | O serviço deve rodar em hardware de baixo desempenho com Raspberry Pi | B |
| RNF04 | O Tempo de Resposta Médio com cache ligado deve ser menor que 20ms para 50 usuários simultâneos | A |
| RFN05 | O Tempo de Resposta para 90% das requisições com cache ligado deve ser menor que 50ms para 50 usuários simultâneos | M |
| RFN06 | O serviço de suportar mais de 1000 RPS (*Requests Per Second* – Requisições Por Segundo) com cache ligado para 50 usuários simultâneos | M |
| RFN07 | O serviço de suportar mais de 50 RPS (*Requests Per Second* – Requisições Por Segundo) com cache desligado para 50 usuários simultâneos | M |

## 

## Protótipo Navegável do Sistema

O Serviço de Cache HTTP é minimalista e roda diretamente no terminal. Segue abaixo uma amostra do serviço rodando num terminal Bash da distribuição Linux Ubuntu:

Texto

Descrição gerada automaticamente

Faça um **vídeo de apresentação do protótipo navegável** desenvolvido e disponibilize-o de forma que os professores envolvidos no processo de avaliação do trabalho possam visualizá-lo. Esse vídeo deve ter duração de, **no máximo**, **3 minutos**. Utilize, preferencialmente, o formato **MP4**.

Nesta seção, indique o ***link* desse vídeo e do repositório** (como o [GitHub](http://github.com), [Bitbucket](https://bitbucket.org/product/), etc) onde seu protótipo navegável está disponível.>

## Diagrama de Classes de Domínio

< Cole aqui uma imagem legível do diagrama de classes de domínio de todo o sistema.>

## Arquitetura da Solução

## Padrão Arquitetural

< Nesta seção, você deve indicar o **padrão arquitetural** escolhido para o desenvolvimento da aplicação (por exemplo, MVC, MVVM, etc). >

## C4 model - Diagrama de Contexto

< Para esta modelagem arquitetural, optou-se por utilizar o modelo C4 para a documentação da arquitetura do software. Mais informações a respeito podem ser encontradas aqui: <https://c4model.com/> e aqui: <https://www.infoq.com/br/articles/C4-architecture-model/>.

Apresente, nesta seção, um **Diagrama de Contexto** que mostre a visão geral da solução proposta e, em seguida, explique-o brevemente, de forma textual. Esse diagrama não precisa seguir os padrões da UML, deve ser completo e tão simples quanto possível, apresentando a **macro arquitetura** da solução, como no exemplo abaixo:



**Figura 1 - Visão Geral da Solução. Fonte:** <https://www.infoq.com/br/articles/C4-architecture-model/>

***Observação: Essa figura deve ser substituída por outra elaborada por você, que seja adequada ao seu projeto. Lembre-se que cada arquitetura é única.***

A Figura 1 mostra o diagrama de contexto da solução proposta, com todos os seus principais módulos e interfaces...

Apresente imagens legíveis do C4 *model* – Diagrama de Contexto da aplicação. >

## Frameworks de Trabalho

< Nesta seção, você deve apresentar os ***frameworks* empregados** no projeto para ***front end*, *back end* e persistência**.

Liste também todas as **tecnologias** que serão utilizadas em sua implementação da solução proposta. >

## Estrutura Base do Front End

< Nesta seção, você deve apresentar imagens legíveis do ***layout* mestre** e do ***menu* de opções do sistema**.>

## Modelo Relacional ou Projeto de Banco de Dados NoSQL

< Cole aqui uma imagem legível do modelo relacional ou do projeto de banco de dados NoSQL de todo o sistema.>

## Plano de Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Apropriação de Horas no Projeto

|  |  |  |
| --- | --- | --- |
| **Histórico de apropriação de horas** | | |
| **Data do registro** | **Atividade** | **Quantidade de horas** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Código da Aplicação

< Informe aqui o ***link*** para seu **repositório público de código**.

Informe também o **endereço do *site* de sua aplicação**. É importante observar que, no ambiente fornecido para a avaliação, a base de dados deverá apresentar exemplos de teste previamente cadastrados que permitam visualizar o correto funcionamento do sistema. Indique também as **credenciais de acesso**, para **todos os perfis de usuários** da aplicação, que devem ser **empregadas pelos avaliadores**.

Indique o ***link*** para acesso ao **vídeo de apresentação de seu projeto**. Espera-se a produção de um vídeo sintético de, **no máximo**, **5 minutos**, no formato **MP4**, apresentando o projeto e a solução desenvolvida.>

## Avaliação Retrospectiva

< Nesta seção, você deve apresentar uma avaliação do processo de desenvolvimento do trabalho.>

## Objetivos Estimados

< Descreva, de forma sucinta, quais eram os objetivos estimados para a execução do projeto.>

## 13.2 Objetivos Alcançados

< Descreva, de forma sucinta, quais foram os objetivos que realmente foram alcançados no projeto.>

## Lições aprendidas

< Descreva, de forma sucinta, quais foram as lições aprendidas na execução do projeto. A coluna “Classificação” deve ser preenchida com “Positiva” ou “Negativa”.>

|  |  |  |
| --- | --- | --- |
|  | **Retrospectiva (Lições Aprendidas)** | |
|  | **Descrição da Lição** | **Classificação** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |

## Referências

< Esse trabalho não requer revisão bibliográfica e, por isso, a inclusão das referências não é obrigatória, embora seja recomendada. Caso você deseje incluir referências empregadas em seu trabalho, relacione-as de acordo com as normas ABNT, disponíveis em [www.pucminas.br](http://www.pucminas.br), no *link*: <http://portal.pucminas.br/imagedb/documento/DOC_DSC_NOME_ARQUI20160217102425-n.pdf>.

Exemplo:

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.>