**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-graduação *Lato Sensu* em Arquitetura de *Software* Distribuído**

Projeto Integrado

Relatório Técnico

<nome do sistema>

<nome(s) do(s) aluno(s)>

Belo Horizonte

<mês e ano>.

# Projeto Integrado – Arquitetura de Software Distribuído

**Sumário**

[Projeto Integrado – Arquitetura de Software Distribuído 2](#_Toc80562699)

[1. Introdução 3](#_Toc80562700)

[2. Cronograma do Trabalho 5](#_Toc80562701)

[3. Especificação Arquitetural da solução 6](#_Toc80562702)

[3.1 Restrições Arquiteturais 6](#_Toc80562703)

[3.2 Requisitos Funcionais 6](#_Toc80562704)

[3.3 Requisitos Não-funcionais 7](#_Toc80562705)

[3.4 Mecanismos Arquiteturais 7](#_Toc80562706)

[4. Modelagem Arquitetural 8](#_Toc80562707)

[4.1 Diagrama de Contexto 8](#_Toc80562708)

[4.2 Diagrama de Container 9](#_Toc80562709)

[4.3 Diagrama de Componentes 9](#_Toc80562710)

[5. Prova de Conceito (PoC) 11](#_Toc80562711)

[5.1 Integrações entre Componentes 11](#_Toc80562712)

[5.2 Código da Aplicação 11](#_Toc80562713)

[6. Avaliação da Arquitetura (ATAM) 13](#_Toc80562714)

[6.1. Análise das abordagens arquiteturais 13](#_Toc80562715)

[6.2. Cenários 13](#_Toc80562716)

[6.3. Evidências da Avaliação 14](#_Toc80562717)

[6.4. Resultados Obtidos 15](#_Toc80562718)

[7. Avaliação Crítica dos Resultados 16](#_Toc80562719)

[8. Conclusão 17](#_Toc80562720)

[Referências 18](#_Toc80562721)

## Introdução

Esta seção tem como propósito apresentar o contexto, o problema e os objetivos deste trabalho. Deve-se iniciar com uma descrição resumida da área de negócio e sua importância para, em seguida, apresentar o que o trabalho aborda. Considere escrever neste tópico (seção 1) **aproximadamente uma página a uma página e meia**, apresentando resumidamente estes aspectos.

Na parte inicial (contextualização) deve-se apresentar o contexto/escopo tratado no trabalho, a **área** ou **contexto maior** no qual ele se insere e por que ele é importante. A contextualização deve ser desenvolvida do geral/genérico para o particular/específico. A citação de pesquisas com dados quantitativos, devidamente referenciadas, é altamente recomendada. As pesquisas citadas devem ser semelhantes, mesmo que em parte, ao trabalho elaborado por você. *(Utilize 1 parágrafo para isto)*

Em seguida, deve-se descrever de forma detalhada o **problema** que o projeto pretende resolver. O problema deve ser apresentado de forma sucinta, completa e clara, pois é necessário conhecê-lo para poder desenvolver a proposta de solução para ele. Atente para o fato de que a descrição do problema não mostra como o mesmo será revolvido, nem aponta para sua solução. É a chamada “dor do cliente”. *(Utilize 1 parágrafo para isto)*

Descrito o problema, mostre a **motivação** para resolvê-lo. Essa motivação pode se caracterizar de diversas formas: acadêmica, mercadológica, social etc. Considere que é neste parágrafo que o trabalho será justificado, de modo que se apresente a importância de sua realização. Não utilize uma motivação pessoal, que é irrelevante aqui. Apresentar resultados esperados e/ou dados quantitativos aqui também é uma boa prática. Por exemplo, podem ser mostrados benefícios como: economia de recursos, maior agilidade nas decisões, melhor qualidade nos processos, etc... *(Utilize 1 parágrafo para isto)*

Apresente agora o **objetivo geral** e alguns (pelo menos 3) **objetivos específicos**. Essas definições servirão para mostrar o objetivo maior/final relacionado à sua descrição arquitetural e como ele poderá ser atingido (metas parciais).

Exemplo:

“O objetivo deste trabalho é apresentar a descrição do projeto arquitetural de uma aplicação XXXXXX.”

“Os objetivos específicos propostos são:

* Realizar um estudo de mercado sobre a área de negócio da aplicação proposta;
* Descrever os requisitos da aplicação de forma resumida, clara e objetiva;
* Etc...

## Cronograma do Trabalho

A seguir é apresentado o cronograma proposto para as etapas deste trabalho.

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 1. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 2. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 3. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 4. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 5. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 6. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 7. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 8. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 9. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 10. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 11. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 12. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 13. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 14. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 15. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 16. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 17. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 18. |  |

Obs: acrecente ou retire linhas, caso seja necessário.

## Especificação Arquitetural da solução

Esta seção apresenta a especificação básica da arquitetura da solução a ser desenvolvida, incluindo diagramas, restrições e requisitos definidos pelo autor, tal que permitem visualizar a macroarquitetura da solução.

## Restrições Arquiteturais

Enumere aqui as restrições arquiteturais relativas ao seu projeto. Lembre-se que as restrições arquiteturais geralmente não são consideradas requisitos uma vez que limitam a solução candidata mas não constituem funcionalidades ou necessidades a serem satisfeitas. Ou seja, diferentemente dos requisitos elas servem para **impor restrições que obrigatoriamente precisam ser satisfeitas**. Exemplos de restrições impostas ao projeto que afetam sua arquitetura podem ser, por ex.:

R1: O software deve ser desenvolvido em Python, com o *framework* Django;

R2: As APIs devem seguir o padrão ReSTful.

R3: xxxxxxxxx

## Requisitos Funcionais

Enumere todos os requisitos funcionais previstos para a sua aplicação. Lembre-se de listar todos os requisitos necessários para garantir cobertura arquitetural, segundo a especificação fornecida. Esta seção deve conter uma lista de requisitos visando produzir a **modelagem completa** da solução.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| RF01 | O sistema deve permitir o auto cadastramento do usuário | B | A |
| RF02 | ... |  |  |
| ... |  |  |  |
| ... |  |  |  |
| ... |  |  |  |

\*B=Baixa, M=Média, A=Alta.

**Obs:** acrescente quantas linhas forem necessárias.

## Requisitos Não-funcionais

Enumere todos os requisitos não-funcionais previstos para a sua aplicação. Entre os requisitos não funcionais inclua todos os requisitos que julgar importantes do ponto de vista arquitetural, ou seja, os requisitos que terão impacto na definição da arquitetura. Os requisitos devem ser descritos de forma completa e preferencialmente quantitativa (ex: tempo de resposta de “x” segundos).

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| RNF01 | O sistema deve ser apresentar disponibilidade 24 X 7 X 365 | A |
| RNF02 | ... |  |
| ... |  |  |
| ... |  |  |

**Obs**: acrescente quantas linhas forem necessárias.

## Mecanismos Arquiteturais

Esta seção deve apresentar uma visão geral dos mecanismos que compõem a arquitetura do software, baseando-se em três estados: (1) análise, (2) design e (3) implementação. Em termos de Análise devem ser listados os aspectos gerais que compõem a arquitetura do software, como: persistência, integração com sistemas legados, geração de logs do sistema, ambiente de front end, tratamento de exceções, formato dos testes, formato de distribuição/implantação (deploy), dentre outros. Em Design deve-se identificar o padrão tecnológico a seguir para cada mecanismo identificado na análise. Em Implementação deve-se identificar o produto a ser utilizado na solução.

|  |  |  |
| --- | --- | --- |
| **Análise** | ***Design*** | **Implementação** |
| Persistência | ORM | Hibernete |
| Front end | Single Page Application | Angular |
| Back end |  |  |
| Integração |  |  |
| Log do sistema |  |  |
| Teste de Software |  |  |
| Deploy |  |  |
| ... |  |  |

## Modelagem Arquitetural

Coloque um texto explcativo aqui, tal como: “Esta seção apresenta a modelagem arquitetural da solução proposta, de forma a permitir seu completo entendimento visando à implementação da prova de conceito (seção X).”

Para esta modelagem arquitetural optou-se por utilizar o modelo C4 para documentação de arquitetura de software. Mais informações a respeito podem ser encontradas aqui: <https://c4model.com/> e aqui: <https://www.infoq.com/br/articles/C4-architecture-model/>. Dos quatro nível que compõem o modelo C4 três serão apresentados aqui e somente o Código será apresentado na próxima seção (5).

## 4.1 Diagrama de Contexto

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



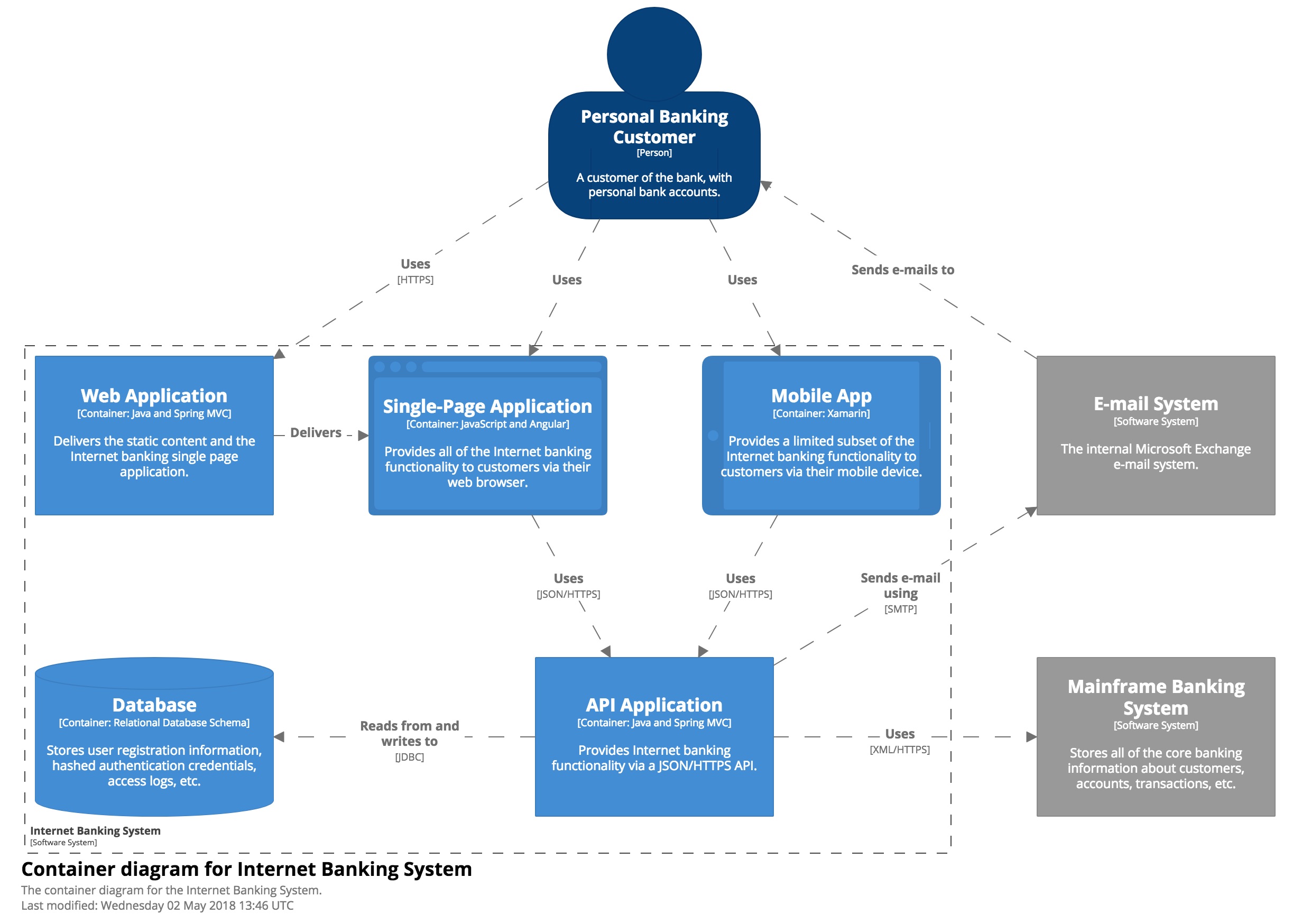
**Figura 1 - Visão Geral da Solução. Fonte:** <https://www.infoq.com/br/articles/C4-architecture-model/>

*Obs: esta é uma figura da internet. Substitua-a por outra elaborada por você, que seja adequada ao seu projeto. Lembre-se que cada arquitetura é única.*

A figura 1 mostra a especificação o diagrama geral da solução proposta, com todos seus principais módulos e suas interfaces...

## 4.2 Diagrama de Container

Apresente o Diagrama de Container da aplicação, indicando como os componentes (aplicativos, armazenamentos de dados, microservices, etc.) que compõem esse sistema de software estão distribuídos e organizados. Lembre-se que as decisões de tecnologia que você tomou devem ser contempladas nesse diagrama.



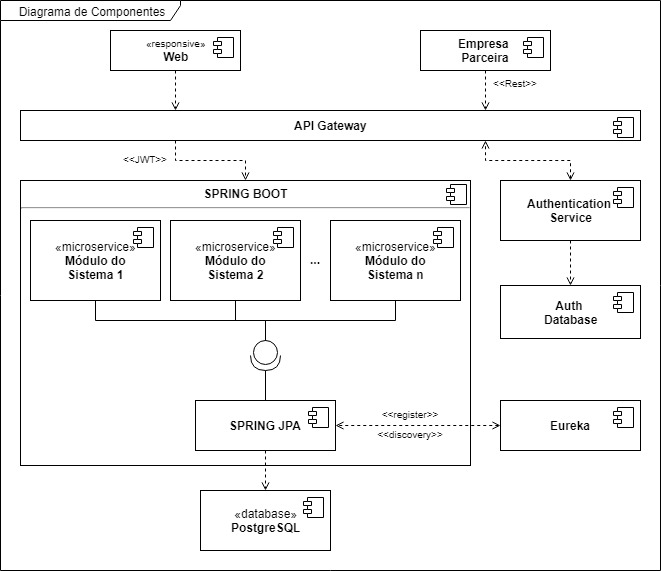
**Figura 2 – Diagrama de container. Fonte:** <https://www.infoq.com/br/articles/C4-architecture-model/>

*Obs: esta é uma figura da literatura. Substitua-a por outra elaborada por você, que seja adequada ao seu projeto. Lembre-se que cada arquitetura é única.*

A figura 2 apresenta os *containers* da aplicação...

## 4.3 Diagrama de Componentes

Apresente o Diagrama de Componentes da aplicação (baseado na UML), indicando os elementos da arquitetura e as interfaces entre eles. Liste os estilos/padrões arquiteturais utilizados e faça uma descrição sucinta dos componentes indicando o papel de cada um deles dentro da arquitetura/estilo/padrão arquitetural. Indique também quais componentes serão reutilizados (navegadores, SGBDs, middlewares, etc), quais componentes serão adquiridos por serem proprietários e quais componentes precisam ser desenvolvidos.



**Figura 3 – Diagrama de Componentes (exemplo). Fonte: imagens Google.**

*Obs: esta é uma figura da internet. Substitua-a por outra elabora por você, que seja adequada ao seu projeto. Lembre-se que cada arquitetura é única.*

*Apresente uma descrição detalhada dos artefatos que constituem o Diagrama de Componentes. Lembre-se de apresentar no diagrama todos os componentes da aplicação.*

*Ex: conforme diagrama apresentado na Figura 3, as entidades participantes da solução são:*

* *Componente 1 - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras nunc magna, accumsan eget porta a, tincidunt sed mauris. Suspendisse orci nulla, sagittis a lorem laoreet, tincidunt imperdiet ipsum. Morbi malesuada pretium suscipit.*
* *Componente 2 - Praesent nec nisi hendrerit, ullamcorper tortor non, rutrum sem. In non lectus tortor. Nulla vel tincidunt eros.*

## Prova de Conceito (PoC)

Nesta seção deve ser detalhada a prova de conceito arquitetural produzida. Não é necessário desenvolver todo o escopo da aplicação proposta, mas deve-se gerar um protótipo que permita avaliar as funcionalidades relativamente aos requisitos arquiteturais definidos, segundo o modelo ATAM (seção 6).

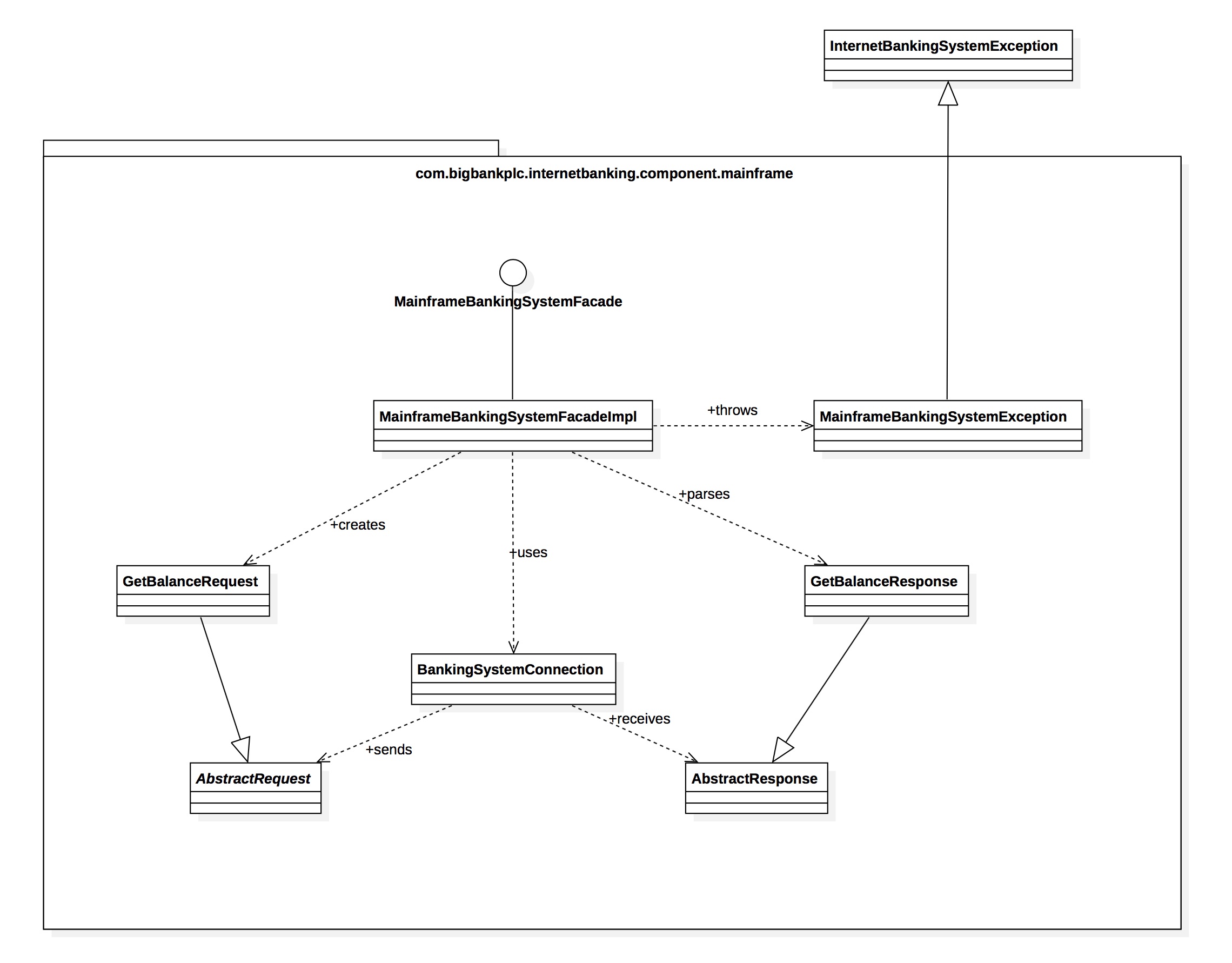
Deve-se produzir também um **vídeo de apresentação do protótipo** criado, disponibilizando-o a os professores envolvidos no processo de avaliação do trabalho.

## Integrações entre Componentes

Desenvolva um protótipo navegável e interativo do sistema, que apresente as interfaces (pode-se utilizar *mock objects*) e as integrações (protocolos, *middlewares*, padrão de troca de dados, etc) entre **os três requisitos prioritários selecionados**. A forma de comunicação entre os componentes por meio dessas integrações deve ser completamente detalhada. Pode-se utilizar alguma ferramenta para a apresentação dessas integrações, além de simulações dos mecanismos de comunicação em ambiente distribuído.

## Código da Aplicação

Nesta seção você deve indicar, segundo o **padrão arquitetural C4**, a estrutura de código da sua aplicação. Exemplo:



**Figura 4 – Estrutura de código da aplicação (exemplo). Fonte:** <https://www.infoq.com/br/articles/C4-architecture-model/>

*Obs: esta é uma figura da internet. Substitua-a por outra elabora por você, que seja adequada ao seu projeto. Lembre-se que cada arquitetura é única.*

A estrutura da aplicação mostrada na Figura 4 apresenta os componentes de código e suas funções no software implementado:

- xxxxxxxxxxxxxx

- xxxxxxxxxxxxxx

- etc...

Indique o ***link* do vídeo e do repositório** (como o [GitHub](http://github.com), [Bitbucket](https://bitbucket.org/product/), etc) onde seu protótipo funcional está disponível. Deve ser disponibilizado um *link* que forneça acessa à página principal da sua aplicação (menu principal), bem como usuário e senha de acesso para eventuais testes.

Como indicado no início desta seção 5, grave um **vídeo de apresentação do seu projeto**. Espera-se a produção de um vídeo sintético, de **no máximo**, **5 minutos**, no formato **MP4** ou outro de ampla utilização, apresentando o projeto e a solução desenvolvida. Pede-se que o aluno coloque o foco da apresentação nas questões arquiteturais mais relevantes, que envolvem o atendimento aos requisitos arquiteturais definidos na seção 3.3.

## Avaliação da Arquitetura (ATAM)

A avaliação da arquitetura desenvolvida neste trabalho é abordada nesta seção visando avaliar se ela atende ao que foi solicitado pelo cliente, segundo o método ATAM.

## 6.1. Análise das abordagens arquiteturais

Apresente aqui um breve resumo das principais características da proposta arquitetural. Para isto, utilize o método Architecture Tradeoff Analysis Method (ATAM), no qual são utilizados cenários para fazer essa análise.

Exemplo:

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributos de Quailidade** | **Cenários** | **Importância** | **Complexidade** |
| Interoperabilidade | Cenário 1: O sistema deve se comunicar com sistemas de outras tecnologias. | A | M |
| Usabilidade | Cenário 2: O sistema deve prover boa usabilidade. | M | B |
| Manutenibilidade | Cenário 3: O sistema deve ter a manutenção facilitada. | M | M |

## 

## 6.2. Cenários

Mostre os cenários utilizados na realização dos testes da sua aplicação. Escolha cenários de testes que demonstrem os requisitos não funcionais (atributos de qualidade) sendo satisfeitos. Priorize os cenários para a avaliação segundo critérios quantitativos ou qualitativos.

*Exemplos de cenários:*

*Cenário 1 - Interoperabilidade: Ao acessar a URL do serviço de informações gerenciais via HTTP GET, o mesmo deve retornar as informações no formato JSON.*

*Cenário 2 - Usabilidade: Ao navegar na tela, o sistema deve apresentar boa usabilidade. A navegação deve apresentar facilidade e o acesso as funcionalidades deve ser bem objetivo para a função que precisar ser realizada, o usuário deve ser capaz de efetuar uma compra em no máximo 5 minutos, assim garantindo a agilidade e a usabilidade para ficar de acordo com um dos requisitos não funcionais.*

*Cenário 3 - Manutenibilidade: Havendo a necessidade de alterar o gateway de pagamento somente será necessário fazer alteração no broker da funiconalidade de pagamento, facilitando a manutenção e os testes.*

## 6.3. Evidências da Avaliação

Apresente as medidas registradas na coleta de dados. Para o que não for possível quantificar apresente uma justificativa baseada em evidências qualitativas que suportem o atendimento ao requisito não-funcional.

|  |  |
| --- | --- |
| Atributo de Qualidade: | Interoperabilidade |
| Requisito de Qualidade: | O sistema deve se comunicar com outras tecnologias. |
| Preocupação: | |
| O sistema deve ter como resposta a uma requisição uma saída de fácil leitura por outro componente. | |
| Cenário(s): | |
| Cenário 1 | |
| Ambiente: | |
| Sistema em operação normal | |
| Estímulo: | |
| O sistema de monitoramento envia uma requisição para o serviço REST do módulo de informações gerenciais. | |
| Mecanismo: | |
| Criar um serviço REST para atender às requisições do sistema de monitoramento | |
| Medida de resposta: | |
| Retornar os dados requisitados no formato JSON | |
| Considerações sobre a arquitetura: | |
| Riscos: | Alguma instabilidade na rede pode deixar a conexão lenta ou mesmo a perda de pacotes. |
| Pontos de Sensibilidade: | Não há |
| Tradeoff: | Não há |

Acrescente imagens e descreva os testes realizados, de tal forma que se comprove a realização da avaliação.

Faça isto para todos os cenários apresentados no tópico 6.1.

## 6.4. Resultados Obtidos

Apresente os resultados da arquitetura produzida, indicando seus pontos fortes e suas limitações. A título de sugestão construa uma tabela apresentando esses resultados, como no exemplo que segue:

|  |  |  |
| --- | --- | --- |
| **Requisitos Não Funcionais** | **Teste** | **Homologação** |
| RNF01: O sistema deve ... | OK | OK |
| RNF02: O sistema deve ... | OK | N.A. |
| RNF03: ... | OK | N.A. |

Obs: N.A.: não se aplica.

## Avaliação Crítica dos Resultados

Apresente aqui, de forma resumida, os principais pontos positivos e negativos da arquitetura proposta. Adote uma postura crítica que permita entender as limitações arquiturais, incluindo os prós e contras das tecnologias. Você pode utilizar o formato textual ou produzir um quadro resumo.

Ex. de quadro resumo:

|  |  |
| --- | --- |
| **Ponto avaliado** | **Descrição** |
| xxxxxxxxxxxxxxx | xxxxxxxxxxxxxxxxxxxx |
|  |  |
|  |  |

## Conclusão

Descreva, de forma sucinta, quais foram as lições aprendidas na execução do seu projeto arquitetural. Procure apresentá-las de tal forma que fiquem configurados os *trade-offs* da arquitetura produzida, como por exemplo, Segurança X Desempenho, Granularidade X Manutenibilidade, etc.

Aqui deve ser apresentado também tudo que se aprendeu com esse projeto, de modo a servir como ajuda para outros profissionais.

Também se faz necessário evidenciar as possibilidades de melhoria do projeto, caso se deseje dar continuidade a ele. Nesse sentido, indique possíveis ajustes ou melhorias arquiteturais, que possam vir a ser realizados futuramente.

Lições aprendidas (ex.):

1. xxxxxxxxxxxxxxxxx
2. xxxxxxxxxxxxxxxxx
3. xxxxxxxxxxxxxxxxx

## 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.