**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Projeto Integrado

Relatório Técnico

Rede Social de Distribuição de Cestas Básicas

Txai Vieira Garcia

Belo Horizonte

Junho, 2022

# Projeto Integrado – Arquitetura de Software Distribuído

**Sumário**

[Projeto Integrado – Arquitetura de Software Distribuído 2](#_Toc106217082)

[1. Introdução 3](#_Toc106217083)

[2. Cronograma do Trabalho 5](#_Toc106217084)

[3. Especificação Arquitetural da solução 7](#_Toc106217085)

[3.1 Restrições Arquiteturais 7](#_Toc106217086)

[3.2 Requisitos Funcionais 7](#_Toc106217087)

[3.3 Requisitos Não-funcionais 9](#_Toc106217088)

[3.4 Mecanismos Arquiteturais 9](#_Toc106217089)

[4. Modelagem Arquitetural 10](#_Toc106217090)

[4.1 Diagrama de Contexto 10](#_Toc106217091)

[Link para o vídeo de apresentação da Etapa 1: 11](#_Toc106217092)

[Etapa 2 – Pendente 11](#_Toc106217093)

[Etapa 3 – Pendente 11](#_Toc106217094)

[Referências 11](#_Toc106217095)

## Introdução

Ocupando a nona posição no ranking global, o Brasil figura entre os dez países mais desiguais do mundo.

Esta situação agravou-se com o surgimento da pandemia de COVID-19, a qual trouxe, além das implicações sanitárias, consequências econômicas e sociais, como o aumento do desemprego, que em 2020 atingiu 14,4%, o que elevou ainda mais a situação de desigualdade do país e trouxe muitas famílias a uma situação de vulnerabilidade social, sendo que, segundo o último levantamento, 27.7% dos brasileiros vivenciavam situação de insegurança alimentar grave ou moderada em 2020. Neste contexto, torna-se ainda mais importante o trabalho de distribuição de cestas básicas, promovido, geralmente, por instituições religiosas, organizações não governamentais e por iniciativas comunitárias ou individuais.

Este trabalho envolve a participação de três atores: os distribuidores, que concentram as atividades de arrecadação de alimentos, montagem e distribuição das cestas; os doadores, que são as pessoas interessadas em doar alimentos e os recebedores, que são as famílias auxiliadas. Não havendo meios de coordenar estes agentes, é comum encontrar alguns problemas principais em todas as iniciativas de distribuição de cestas:

* A divulgação da arrecadação e da distribuição é restrita;
* Os doadores, por não terem informações suficientes, acabam doando itens de maneira desproporcional (e.g. doam muito feijão, mas o distribuidor está com falta de arroz), causando desperdício;
* Em cada local é necessário que o recebedor faça um cadastro e passe por uma avaliação, o que é um desperdício de tempo para os recebedores e para os distribuidores;
* A demanda de cestas geralmente excede a oferta destas, e sem um meio de divulgar esta informação, muitos recebedores acabam indo ao local de distribuição sem saber se poderão receber uma cesta.

Uma plataforma, que permitisse a centralização de informações e a interação entre todos os três atores envolvidos no processo, permitiria redução dos desperdícios de recursos e permitiria que mais famílias fossem auxiliadas, sobretudo em situações de picos de demanda, como no caso de desastres ou eventos como a recente pandemia. Além disso, as informações coletadas, poderiam servir de auxílio para estudos e na tomada de decisões de ações de combate à fome.

O objetivo do presente trabalho é apresentar a descrição arquitetural de uma plataforma deste tipo, denominada ReCeBa (Rede social de distribuição de Cestas Básicas), que deve atuar como centralizadora a fim de resolver os problemas elencados acima.

Os objetivos específicos a serem alcançados são:

* Elencar as restrições arquiteturais e os requisitos necessários à construção do sistema;
* Desenhar uma arquitetura capaz de prover a segurança necessária dos dados dos usuários e de ser escalável, principalmente em picos de utilização, como durante a ocorrência de desastres;
* Desenhar uma interface que seja intuitiva e leve, considerando o público-alvo da aplicação;
* Demonstrar a viabilidade de uma plataforma do tipo, por meio de prova de conceito e posterior protótipo;
* Documentar a construção do sistema, a fim de permitir um trabalho de expansão deste no futuro;

## Cronograma do Trabalho

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 03/05/2022 | 07/05/2022 | 1. Levantamento de ideias de possíveis projetos | Tema do projeto integrado |
| 08/05/2022 | 10/05/2022 | 1. Estudo do contexto e problemática do tema proposto | Introdução deste documento |
| 08/05/2022 | 10/05/2022 | 1. Elaboração da introdução | Introdução deste documento |
| 11/05/2022 | 11/05/2022 | 1. Elaboração do cronograma | Cronograma definido |
| 12/05/2022 | 15/05/2022 | 1. Levantamento das restrições arquiteturais | Lista de restrições arquiteturais |
| 16/05/2022 | 20/05/2022 | 1. Levantamento dos requisitos funcionais | Lista de requisitos funcionais |
| 21/05/2022 | 24/05/2022 | 1. Levantamento dos requisitos não funcionais | Lista de requisitos não funcionais |
| 25/05/2022 | 30/05/2022 | 1. Definição dos mecanismos arquiteturais | Lista de mecanismos arquiteturais |
| 01/06/2022 | 03/06/2022 | 1. Elaboração do diagrama de contexto | Diagrama de contexto da solução |
| 04/06/2022 | 05/06/2022 | 1. Revisão deste documento | Etapa 1 do Relatório técnico revisada |
| 06/06/2022 | 15/06/2022 | 1. Elaboração do vídeo de apresentação | Vídeo de 3 minutos apresentando o contexto do projeto e a solução proposta |
| 20/06/2022 | 23/06/2022 | 1. Elaboração do diagrama de container | Diagrama de container da solução |
| 24/06/2022 | 26/06/2022 | 1. Elaboração do diagrama de componentes | Diagrama de componentes da solução |
| 27/06/2022 | 08/07/2022 | 1. Elaboração do wireframe da prova de conceito | Wireframe navegável da prova de conceito com a implementação de 3 requisitos funcionais |
| 09/07/2022 | 22/07/2022 | 1. Criação do front end da prova de conceito | Frontend disponibilizado em nuvem, implementado conforme wireframe |
| 23/07/2022 | 06/08/2022 | 1. Criação do back end da prova de conceito | Backend disponibilizado em nuvem, com todas as interfaces necessárias para a integração com o frontend |
| 07/08/2022 | 12/08/2022 | 1. Integração do backend e do frontend | Frontend e backend integrados |
| 13/08/2022 | 15/08/2022 | 1. Revisão do relatório técnico | Etapa 2 do relatório técnico revisada |
| 16/08/2022 | 18/08/2022 | 1. Levantamento dos atributos de qualidade e cenários da avaliação da arquitetura | Lista de itens de análise das abordagens arquiteturais |
| 19/08/2022 | 26/08/2022 | 1. Detalhamento dos cenários levantados na etapa anterior | Lista detalhada de cenários |
| 27/08/2022 | 16/09/2022 | 1. Levantamento das evidências relativas a cada cenário | Lista de evidências |
| 17/09/2022 | 20/09/2022 | 1. Construção da seção de resultados obtidos | Texto e tabela de resultados obtidos, evidenciando RNFs homologados e pontos fortes e fracos da arquitetura |
| 21/09/2022 | 23/09/2022 | 1. Avaliação dos resultados obtidos | Quadro resumo contendo as avaliações |
| 24/09/2022 | 27/09/2022 | 1. Elaboração da conclusão | Conclusão do relatório |
| 28/09/2022 | 30/09/2022 | 1. Revisão do relatório técnico | Relatório técnico completo e revisado |
| 01/10/2022 | 15/10/2022 | 1. Elaboração do vídeo explicativo | Vídeo de 5 minutos apresentando o projeto implementado e a arquitetura desenhada |

## 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 macro arquitetura da solução.

## Restrições Arquiteturais

As restrições arquiteturais foram levantadas, considerando como público-alvo pessoas de baixo poder aquisitivo. Além disso, também foi considerado que o projeto não possui um objetivo comercial, então deve permitir envolvimento da comunidade para expansão e manutenção.

|  |  |
| --- | --- |
| **ID** | **Descrição** |
| R1 | Por não ser uma aplicação com clientes empresariais, o sistema deve permitir acesso público através da internet |
| R2 | O público-alvo possui, majoritariamente, acesso à internet através de celulares. Então o sistema deve prover uma solução de frontend mobile, ou através de aplicativos ou através de web apps |
| R3 | Em muitos lugares, o acesso à internet pode ser instável. Portanto, o sistema deve prover algumas funcionalidades offline |
| R4 | Serão trafegados dados pessoais identificáveis, portanto o sistema deve prover mecanismos de segurança para proteção destes dados |
| R5 | Não dispondo de investimentos, o sistema deve utilizar ao máximo tecnologias open source e gratuitas |
| R6 | O desenvolvimento do sistema deve utilizar controle de versão através de repositório Git público e deve ser estruturado para permitir o envolvimento da comunidade open source |

## Requisitos Funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| RF01 | O sistema deve permitir o acesso de usuários não cadastrados | B | A |
| RF02 | O sistema deve permitir o auto cadastramento do usuário | B | A |
| RF03 | O sistema deve permitir que usuários cadastrem iniciativas de distribuição de cestas (IDC) | M | A |
| RF04 | As IDCs devem conter nome, endereço, meta de cestas do mês, quantidade de cestas disponíveis, itens faltantes, composição das cestas e data de entrega para o mês corrente | B | A |
| RF05 | As IDCs devem ser acessíveis por usuários cadastrados e não cadastrados | B | A |
| RF06 | O sistema deve permitir que os criadores das IDCs definam outros usuários como administradores destas últimas. Os administradores terão as mesmas atribuições e permissões dos criadores | A | B |
| RF07 | O sistema deve permitir que os administradores das IDCs cadastrem metas de cestas básicas | B | M |
| RF08 | O sistema deve permitir que os administradores das IDCs cadastrem composições de cestas básicas | B | M |
| RF09 | O sistema deve permitir que os administradores das IDCs cadastrem itens recebidos por doação | B | M |
| RF10 | O sistema deve montar cestas básicas, automaticamente, conforme a composição da cesta, a meta de distribuição e a quantidade de itens recebidos | M | B |
| RF11 | O sistema deve exibir as IDCs cadastradas através de mapa e lista | A | B |
| RF12 | O sistema deve permitir que usuários cadastrados, criem pedidos de recebimento de cestas básicas | B | M |
| RF13 | O sistema deve permitir que o usuário envie o pedido para até 5 IDCs | M | M |
| RF14 | O sistema deve exibir, para os administradores das IDCs, os pedidos de recebimento pendentes | B | M |
| RF15 | O sistema deve permitir que os administradores das IDCs aceitem o pedido de recebimento, rejeitem-no ou o coloquem em fila de espera | M | M |
| RF16 | Ao aceitar um pedido de recebimento, o administrador da IDC terá 24h de tolerância para poder desfazer a ação | A | B |
| RF17 | O sistema deve avisar o administrador da IDC caso ele aceite um pedido de recebimento e não tenha cestas suficientes para atender o pedido | B | B |
| RF18 | Quando uma cesta for completada, o sistema deve exibir a fila de pedidos pendentes para que ela possa ser reavaliada pelo administrador | M | B |
| RF19 | O sistema deve notificar o usuário por e-mail, notificação push e SMS quando um pedido de recebimento for aceito | A | A |
| RF20 | O sistema deve notificar o usuário por e-mail, notificação push e SMS 3 dias antes da data de entrega da cesta básica | A | M |
| RF21 | O sistema deve destacar, para o administrador da IDC, usuários que já tiveram seus pedidos aceitos por outras IDCs | M | A |
| RF22 | O sistema deve retirar esta marcação um mês após o recebimento da cesta pelo usuário | M | A |
| RF23 | O sistema deve permitir que o administrador de IDC registre o recebimento da cesta pelo usuário | B | A |

\*B=Baixa, M=Média, A=Alta.

## Requisitos Não-funcionais

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| RNF01 | O sistema deve estar disponível das 6h às 24h (18h) durante os sete dias da semana | A |
| RNF02 | A interface de usuário deve ser acessível, devendo ser atendidos os níveis A e AA definidos no WCAG 2.1 | M |
| RNF03 | O sistema deve atender até 2000 usuários simultaneamente sem degradação da performance | B |
| RNF04 | A interface web deve rodar nas versões mais recentes do Microsoft Edge, Mozilla Firefox e Google Chrome e deve possuir design responsivo | B |
| RNF05 | O tempo de resposta da interface de usuário deve ser de até 1 segundo 90% do tempo | M |
| RNF06 | O desenvolvimento do sistema deve atender o nível 2 do Application Security Verification Standard v4.0 da OWASP a fim de evitar falhas de segurança | M |
| RNF07 | 100% dos dados pessoais identificáveis devem ser encriptados em repouso e em trânsito | A |

## Mecanismos Arquiteturais

|  |  |  |
| --- | --- | --- |
| **Análise** | ***Design*** | **Implementação** |
| Persistência | ORM | Hibernate |
| Persistência | Banco de dados relacional | PostgreSQL |
| Front end | Single Page Application | Vue.js |
| Front end | Mobile | Flutter |
| Back end | Micro serviços | Spring Boot |
| Desenvolvimento | Plataforma de desenvolvimento orientada a objetos | Java |
| Integração | API Restful | HTTP+JSON |
| Integração | API Gateway | Kong |
| Autorização e Autenticação | OAuth2 | Keycloack |
| Log do sistema | Plataforma de gerenciamento de logs | Graylog |
| Teste de Software | Testes de unidade | JUnit |
| Deploy | Conteinerização | Docker/Kubernets |
| Deploy | CI/CD | Jenkins |

## Modelagem Arquitetural

Esta seção apresenta a modelagem arquitetural da solução proposta, de forma a permitir seu completo entendimento visando à implementação da prova de conceito (seção 5).

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

Diagrama

Descrição gerada automaticamente

**Figura 1 - Visão Geral da Solução**

A figura 1 mostra o diagrama de contexto da solução proposta, com todos os usuários envolvidos e suas interações com o sistema. Nele é possível perceber os três perfis presentes no sistema: distribuidores, recebedores e visitantes, bem como suas atribuições. Também é possível notar que o ReCeBa atua como um centralizador destas interações.

## Link para o vídeo de apresentação da Etapa 1:

<https://youtu.be/LrCGrWfXn34>

## Etapa 2 – Pendente

## Etapa 3 – Pendente

## Referências

AGÊNCIA O GLOBO**. Brasil é nono país mais desigual do mundo, diz IBGE.** São Paulo, 12 de Novembro de 2020. Disponível em: <https://exame.com/economia/brasil-e-nono-pais-mais-desigual-do-mundo-diz-ibge/>. Acessado em: 08 de Maio de 2022.

UOL. **Desemprego no Brasil cai a 13,2%, mas renda tem a maior queda da história.** São Paulo, 27 de Outubro de 2021. Disponível em: <https://economia.uol.com.br/noticias/redacao/2021/10/27/pnad-ibge-desemprego-brasil.htm>. Acessado em: 08 de Maio de 2022.

PELLEGRINI, A. **Qual o quadro de insegurança alimentar no Brasil da pandemia.** São Paulo, 13 de Abril de 2021. Disponível em: <https://www.nexojornal.com.br/expresso/2021/04/13/Qual-o-quadro-de-inseguran%C3%A7a-alimentar-no-Brasil-da-pandemia>. Acessado em: 08 de Maio de 2022.

BROWN, S. **O modelo C4 de documentação para Arquitetura de Software**. 01 de Agosto de 2018. Disponível em: <https://www.infoq.com/br/articles/C4-architecture-model/>. Acessado em 01 de Junho de 2022.