

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Projeto Integrado

Relatório Técnico

Quiosque

Milton França Filho

Belo Horizonte  
junho de 2022

## Projeto Integrado – Arquitetura de Software Distribuído

### *Sumário*

Projeto Integrado – Arquitetura de Software Distribuído	2
1. Introdução	3
2. Cronograma do Trabalho	4
3. Especificação Arquitetural da solução	4
3.1 Restrições Arquiteturais	4
3.2 Requisitos Funcionais	4
3.3 Requisitos Não-funcionais	5
3.4 Mecanismos Arquiteturais	5
4. Modelagem Arquitetural	5
4.1 Diagrama de Contexto	6
4.2 Diagrama de Container	6
4.3 Diagrama de Componentes	8
5. Prova de Conceito (PoC)	8
5.1 Integrações entre Componentes	9
5.2 Código da Aplicação	10
6. Avaliação da Arquitetura (ATAM)	13
6.1. Análise das abordagens arquiteturais	13
6.2. Cenários	13
6.3. Evidências da Avaliação	14
6.4. Resultados Obtidos	18
7. Avaliação Crítica dos Resultados	19
8. Conclusão	20

## **1. Introdução**

A aplicação Quiosque será responsável pela pesquisa, efetivação e controle das vendas de produtos através de QR code pix, para proprietário de quiosque nos mais diversos segmentos como: lanches, frutarias e acessórios para smartphone, utilizando os mais diversos dispositivos móveis e web. Esse documento serve de norteador do desenvolvimento e posteriormente, da sustentação do sistema Quiosque, apresentando através de suas visões, as decisões significativas tomadas e como operar. A aplicação Quiosque contempla em seu escopo a parametrização de produtos, pesquisa e controle de vendas via Qr Code pix.

Podemos considerar que “Com a chegada da pandemia de Coronavírus (Covid-19), muitos setores da economia acabaram sendo prejudicados, principalmente o comercial”( <https://g1.globo.com/sp/vale-do-paraiba-regiao/especial-publicitario/prefeitura-de-braganca-paulista/noticia/2021/03/16/feira-conheca-o-aplicativo-que-vem-auxiliando-os-feirantes-durante-a-pandemia.ghtml>).

Com esse cenário global, há os desafios de evitar aglomeração e manter este setor aquecido.

A solução é disponibilizar um aplicativo para auxiliar os microempreendedores a continuarem atendendo seus clientes de um jeito prático e fácil juntamente com a ação de evitar aglomeração, melhorando a qualidade no atendimento, recebimentos e com mais agilidade no atendimento.

O objetivo deste trabalho é apresentar a descrição do projeto arquitetural de uma aplicação chamada Quiosque para informatizar proprietários de quiosques, visando agilizar o atendimento e a tramitação de processos de pagamentos via Qr code Pix. O projeto visa facilitar atendimento aos clientes, evitando aglomeração, de forma segura, baixo custo e com acesso em qualquer via web e dispositivos móveis.

Os objetivos específicos propostos são:

- Criar um módulo para parametrizar os produtos como: cadastro, upload de imagens, descrição dos produtos e gravação do Qr code.
- Criar uma página principal para exibir os produtos para clientes consultar os produtos.

## 2. Cronograma do Trabalho

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

Datas		Atividade / Tarefa	Produto / Resultado
De	Até		
20 /06 / 2022	30 /06 /2022	1.Criação de página principal para exibir produtos	Página PWA principal
04 /07 /2022	22 /07 /2022	2.Módulo para parametrização	Formulário
18 / 07 /2022	28 /07 /2022	3.Modelagem do banco de dados	Base de dados
08/08/2022	19/08/2022	4.Salvar Qr Code para cada produto	Página PWA principal

## 3. Especificação Arquitetural da solução

Módulo de parametrização

O Sistema deve permitir a parametrização dos produtos, uploads das imagens, o usuário poderá atualizar e inserir novos registros.

Página principal

Será apresentado aos usuários da aplicação uma página principal com o catálogo de produtos, suas descrições e como realizar compras.

### 3.1 Restrições Arquiteturais

R1: O software deve ser desenvolvido em JavaScript, com HTML e CSS;
R2: As APIs devem seguir o padrão ReSTful.
R3: O backend deve ser desenvolvido em java
R4: O backend deve ser implementado através do framework Quarkus
R5: O backend deve ser adotado o padrão arquitetural hexagonal
R6: Banco de dados deve ser NoSQL
R7: O sistema deve ser distribuído através do Whatsap

### 3.2 Requisitos Funcionais

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	O sistema deve permitir o cadastramento de produtos	B	A
RF02	O sistema deve permitir o upload de imagens	A	B

RF03	O Sistema deve permitir visualizar todos os produtos disponíveis para compra	A	A
RF04	O Sistema deve permitir a pesquisa pelo nome do produto	M	B
RF05	O Sistema deve permitir realizar pagamentos via qr code pix	A	B
RF06	O sistema deve ser capaz de realizar pesquisa em 30 segundos		

\*B=Baixa, M=Média, A=Alta.

### 3.3 *Requisitos Não-funcionais*

ID	Descrição	Prioridade B/M/A
RNF01	O sistema deve ser apresentar disponibilidade 24 X 7 X 365	A
RNF02	O sistema deve ser capaz de subir imagem no tempo máximo 1 minuto	B
RNF03	O Sistema deve ser capaz exibir QR-Code para pagamento via pix	A

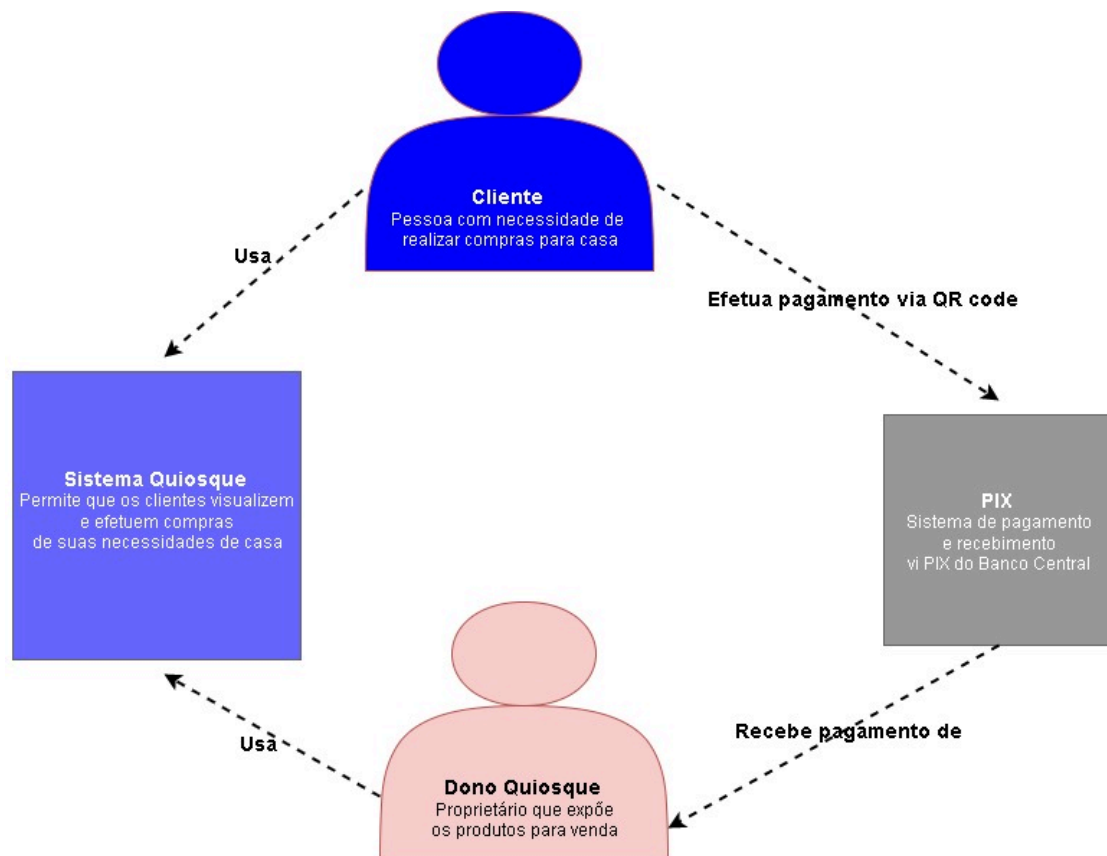
### 3.4 *Mecanismos Arquiteturais*

Análise	Design	Implementação
Persistência	ORM	Hibernate
Front end	Page Web Application	Node
Back end	Java	Rest
Versionamento	Versionamento Código	Git
Log do sistema	Framework de log	Log4j
Documentação	Documentação de API	Swagger

## 4. *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), ou seja, será exibido o App que será construído e como será encaixado no mundo em termos das pessoas que o utilizam e em relação aos outros sistemas de software necessário para interação.

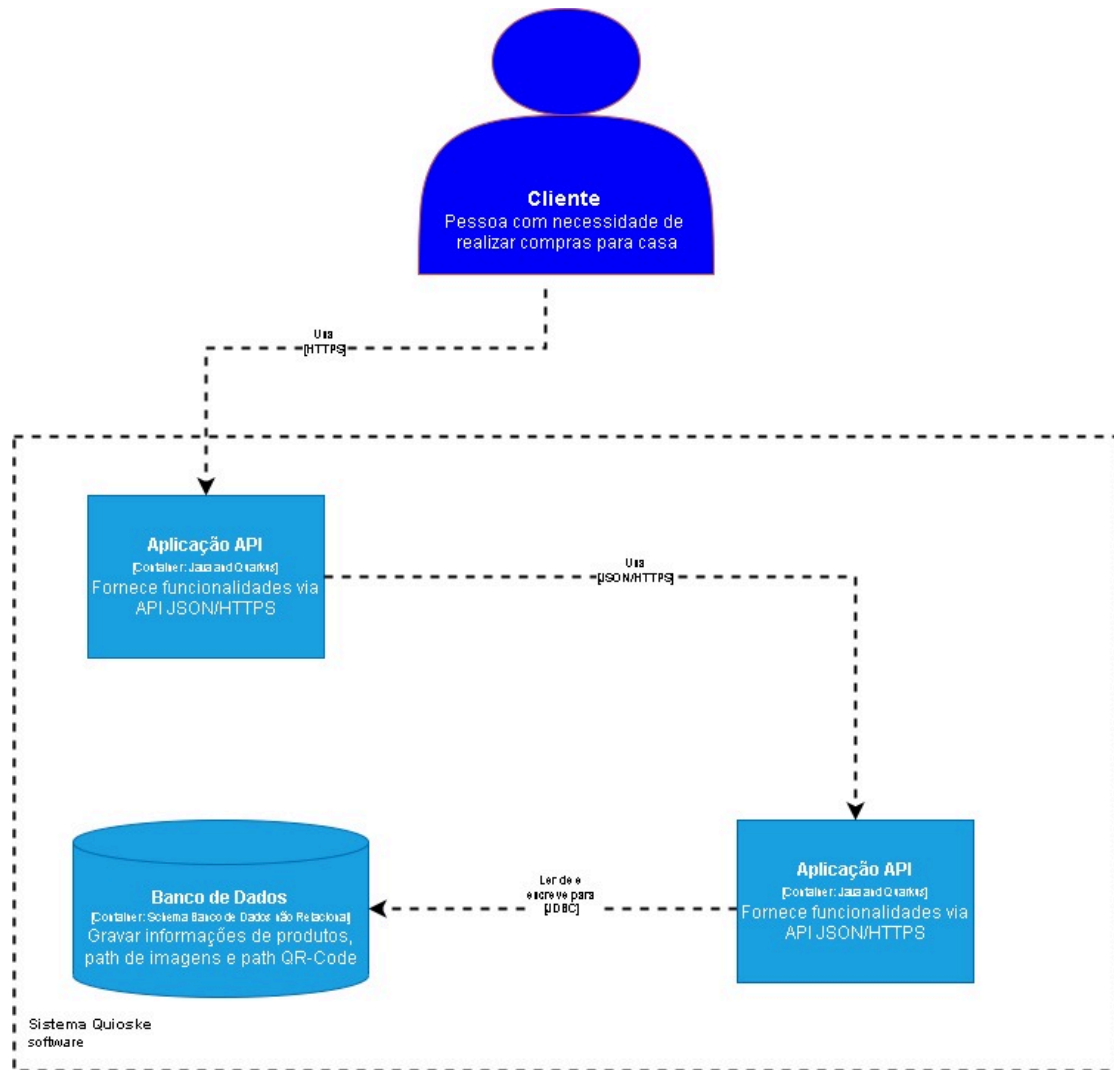
#### 4.1 Diagrama de Contexto



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

Link vídeo: <https://youtube.com/shorts/Aa10FaHJ-C8>

#### 4.2 Diagrama de Container



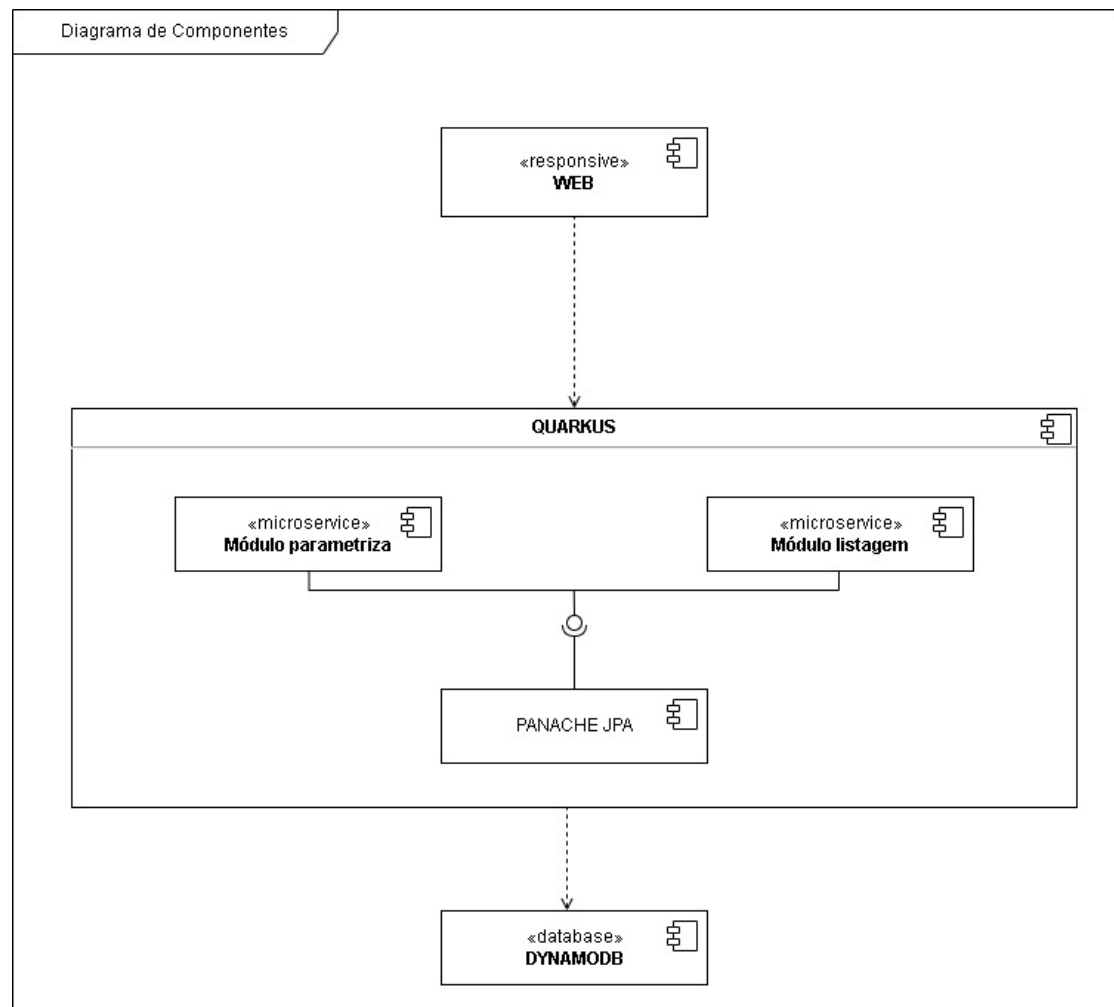
**Diagrama de Container para o Sistema Quioske**

O diagrama de container para o sistema Quioske

Última modificação: Sábado 13 de agosto de 2022 21:37

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

### 4.3 Diagrama de Componentes



*Figura 3 – Diagrama de Componentes.*

Conforme diagrama apresentado na Figura 3, as entidades participantes da solução são:

Componente Web – Responsável pela apresentação e exposição dos produtos, componente responsável pela interação do usuário com a aplicação.

- Componente Quarkus – Este componente vai disponibilizar as implementações dos endpoints que transitarão as informações do componente Web até o componente Dynamo, e o caminho inverso. Internamente teremos o módulo de cada Api e o módulo da camada de acesso a dados, o Panache JPA.
- Componente Dynamo – Onde será gravado a parametrização dos produtos e de onde será buscados as mesmas informações.

## 5. Prova de Conceito (PoC)

Nesta seção será detalhada a prova de conceito arquitetural. Foi gerado um protótipo que permite avaliar as funcionalidades relativamente aos requisitos arquiteturais definidos, segundo o modelo ATAM (seção 6).

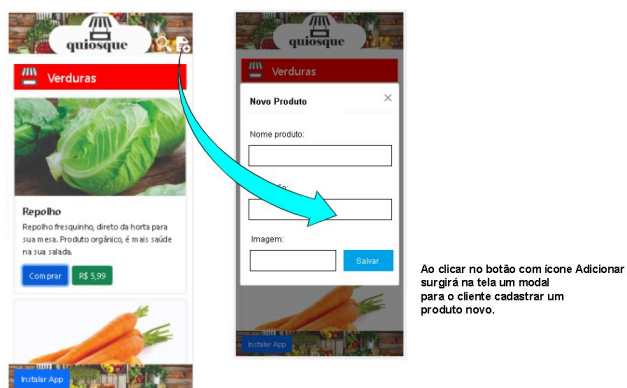
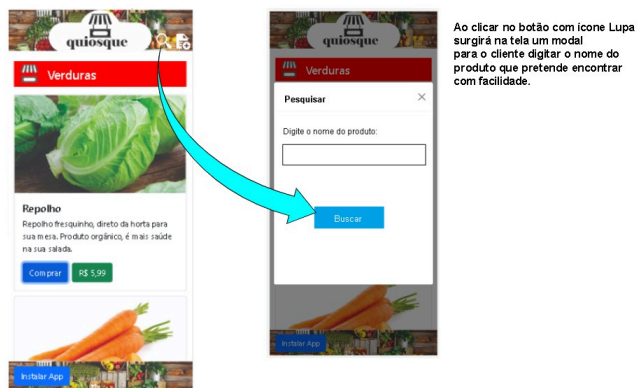
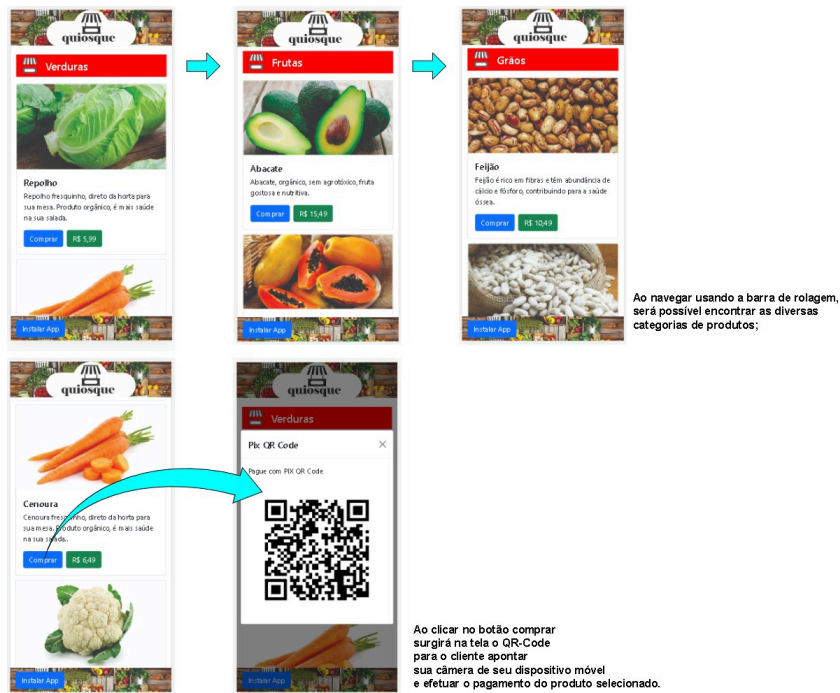


Segue abaixo, **link do vídeo de apresentação do protótipo** criado.

### **5.1    *Integrações* entre Componentes**

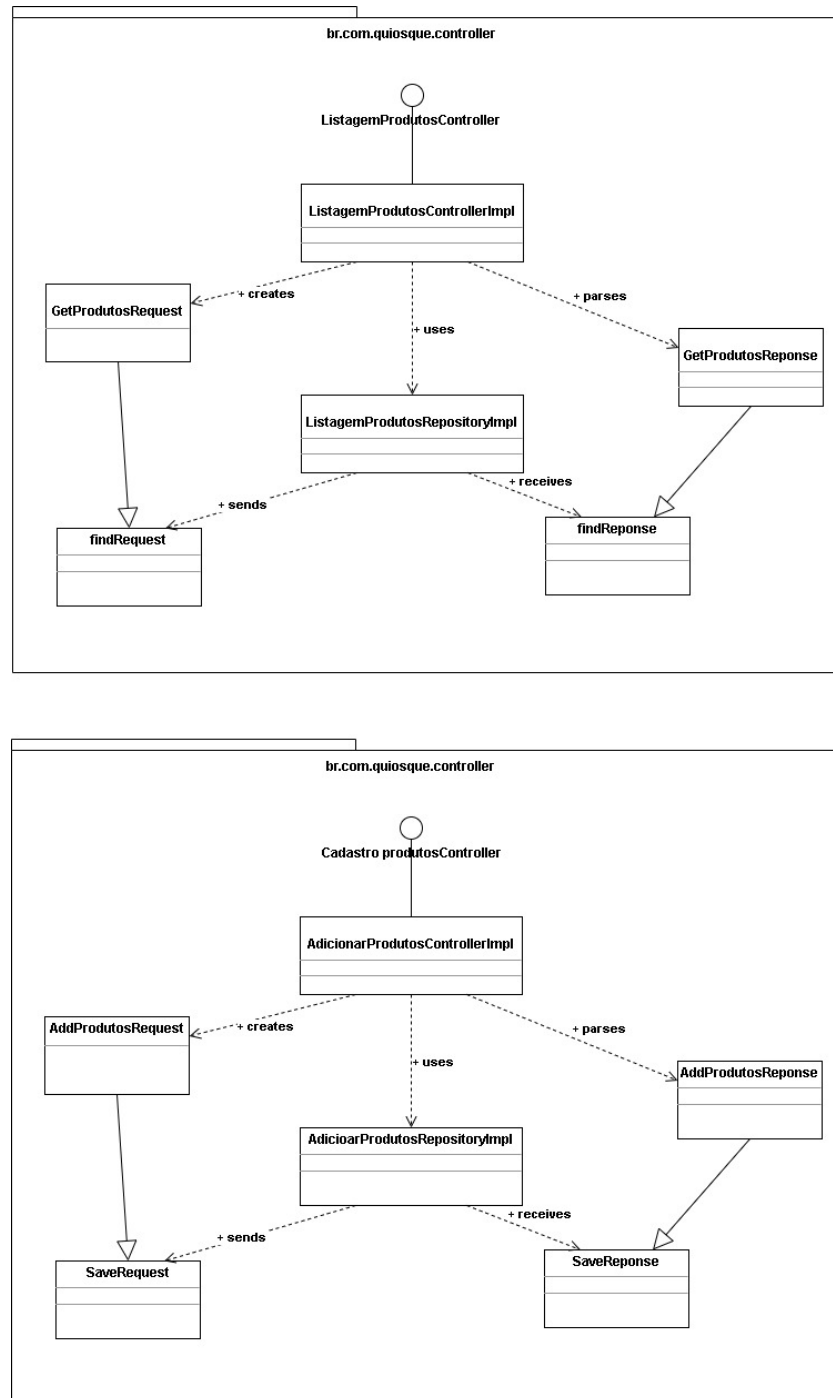
Protótipo navegável e interativo do sistema, que apresente as interfaces e as integrações (protocolos, *middlewares*, padrão de troca de dados, etc) entre **os três requisitos prioritários selecionados**.

## Quiosque



## 5.2 Código da Aplicação

Nesta seção é indicado, segundo o **padrão arquitetural C4**, a estrutura de código da aplicação.



**Figura 4 – Estrutura de código da aplicação.**

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

- ListagemProdutosControllerImpl, responsável por expor o end-point para o App consumir.

## Quiosque

- ListagemProdutosRepositoryImpl, contém a implementação dos métodos findRequest e findResponse, para controlar requisição e resposta à listagem de produtos.
- AdicionarProdutosControllerImpl, responsável por expor e receber dados de cadastro de produtos, vindo do App.
- AdicionarProdutosRepository, contém a implementação dos métodos saveRequest e saveResponse, para controlar o fluxo de dados que serão persistido no banco de dados.

link vídeo:

**<https://youtu.be/MK7pHXCZwNI>**

link repositório:

**<https://github.com/FrancaFilho/quioske>**

link acesso à página principal:

**<https://quiosque.advancerastreamento.com.br/>**

## 6. 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

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

Atributos de Qualidade	Cenários	Importância	Complexidade
Usabilidade	Cenário 1: O sistema deve prover boa usabilidade.	M	B
Manutenibilidade	Cenário 2: O sistema deve ter a manutenção facilitada.	M	M
Segurança	Cenário 3: O sistema deve prover finalização da compra confiável.	A	M
Desempenho	Cenário 4: O sistema deve prover rapidez em em todo seu processo.	A	M

### 6.2. Cenários

Cenário 1 - 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 2 - Manutenibilidade: Havendo a necessidade de alterar o destino do consumo dos dados em JSON somente será necessário fazer alteração da url da funcionalidade, facilitando a manutenção e os testes.

Cenário 3 - Segurança: Ao acessar o APP da aplicação Quiosque, o mesmo deve exibir somente a imagem do QR-Code do pix para garantir um pagamento.

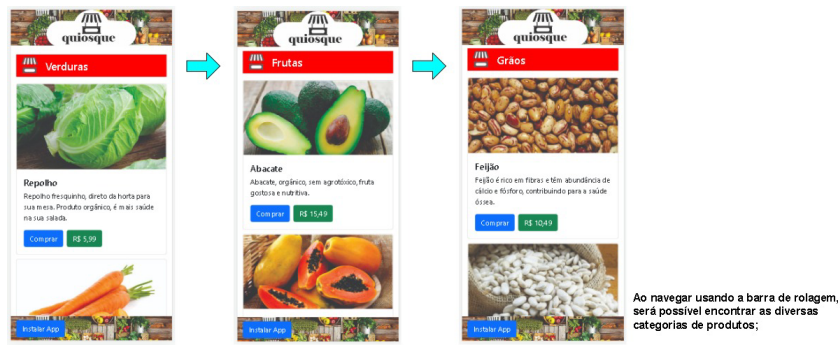
Cenário 4 - Desempenho: Ao acessar o APP da aplicação Quiosque, o mesmo deve garantir rapidez e performance ao alternar de telas(modais) dentro da aplicação.

### 6.3. Evidências da Avaliação

Medidas registradas na coleta de dados.

Atributo de Qualidade:	Usabilidade
Requisito de Qualidade:	O sistema deve prover boa usabilidade.
Preocupação:	
O sistema deve ter fácil navegação apresentando boa usabilidade, facilidade ao acessar as funcionalidades pelo usuário.	
Cenário(s):	
Cenário 1	
Ambiente:	
Sistema em operação normal	
Estímulo:	
O usuário navegará rolando a tela para baixo, visualizando produtos e novas categorias.	
Mecanismo:	
Criar um design intuitivo com cores que facilite a navegação	
Medida de resposta:	
Navegação por todas as categorias	
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

*Segue evidência abaixo:*



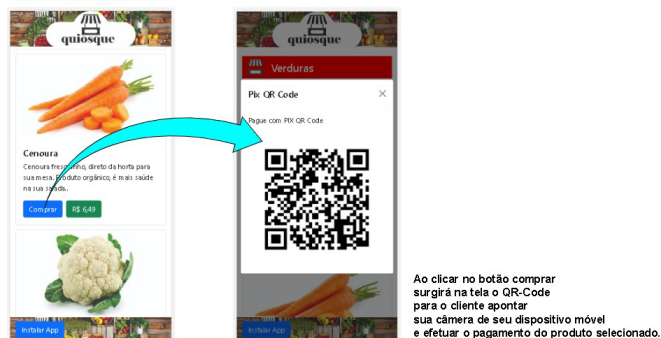
Atributo de Qualidade:	Manutenibilidade
Requisito de Qualidade:	O sistema deve prover boa manutenibilidade.
Preocupação:	
O sistema deve permitir com facilidade alterar base de consumo dos dados de produtos, imagens e preços.	
Cenário(s):	
Cenário 2	
Ambiente:	
Sistema em desenvolvimento	
Estímulo:	
Alterar base dos dados sem alterar a estrutura de design e formatação da aplicação.	
Mecanismo:	
Alterar url dos métodos para redirecionar o consumo do rest	
Medida de resposta:	
Não há	
Considerações sobre a arquitetura:	
Riscos:	Não há
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Atributo de Qualidade:	Segurança
Requisito de Qualidade:	O sistema deve prover segurança.
Preocupação:	

## Quiosque

O sistema deve permitir que ao clicar no botão comprar seja exibido somente o QR-Code Pix para concluir o pagamento.	
Cenário(s):	
Cenário 3	
Ambiente:	
Sistema em operação normal	
Estímulo:	
Clicar no botão comprar do produto desejado.	
Mecanismo:	
Clicar no botão comprar	
Medida de resposta:	
Exibir modal com QR-Code Pix	
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

Segue evidência abaixo:



Atributo de Qualidade:	Desempenho
Requisito de Qualidade:	O sistema deve prover desempenho.
Preocupação:	
Ao acessar o APP da aplicação Quiosque, o mesmo deve garantir rapidez e performance ao alternar de telas(modais).	
Cenário(s):	

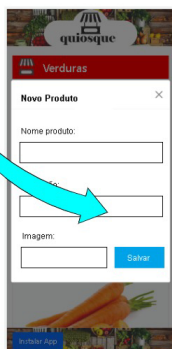


Cenário 4	
Ambiente:	
Sistema em operação normal	
Estímulo:	
Clicar no botão comprar do produto desejado e navegar para outras categorias.	
Mecanismo:	
Clicar no botão comprar e rolar tela	
Medida de resposta:	
Não há	
Considerações sobre a arquitetura:	
Riscos:	Alguma instabilidade na rede pode deixar a conexão lenta ou mesmo a perda de pacotes.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Segue evidência abaixo:



Ao clicar no botão com ícone Lupa surgirá na tela um modal para o cliente digitar o nome do produto que pretende encontrar com facilidade.



Ao clicar no botão com ícone Adicionar surgirá na tela um modal para o cliente cadastrar um produto novo.

## 6.4. Resultados Obtidos

Resultados da arquitetura produzida, indicando seus pontos fortes e suas limitações.

Requisitos Não Funcionais	Teste	Homologação
RNF01: O sistema deve ser apresentar disponibilidade 24 X 7 X 365	OK	OK
RNF02: O sistema deve ser capaz de subir imagem no tempo máximo 1 minuto	OK	OK
RNF03: O sistema deve ser capaz de realizar pesquisa em 30 segundos	OK	OK
RNF04: O Sistema deve ser capaz exibir QR-Code para pagamento via pix	OK	OK

## 7. Avaliação Crítica dos Resultados

Os principais pontos positivos e negativos da arquitetura proposta.

Ponto avaliado	Descrição
Usabilidade	A facilidade ao acessar as funcionalidades pelo usuário.
Prover manutenibilidade.	A alteração da base de consumo dos dados se torna simples, quando trabalha-se com Microserviços Rest.
Prover segurança	Ao permitir efetuar a compra através da exibição do QR-Code Pix para concluir o pagamento.
Prover desempenho	Ao garantir rapidez e performance ao acessar funcionalidades ou tela (modais).
Distribuição	Será por compartilhamento do link, ao abrir a aplicação pela primeira vez será disponibilizado um botão para instalar o App.
Curva de Aprendizado	Ao adotar esta arquitetura será necessário que a equipe de desenvolvimento adquira novos conhecimentos ou habilidades, em se tratando de Quarkus e programação assíncrona. A curva de aprendizado será em um tempo médio de aproximadamente 3 (três) meses para capacitação, através de documentação oficial, playlists públicas (vídeo-aulas) e cursos em plataformas específicas.
Dependências Externas	O front-end possui uma dependência com back-end, que se encontra em outro servidor, podendo em alguma ocasião se encontrar indisponível. Indisponibilidade esta que poderá ser por algumas das causas: Atualização de versão, manutenção da base de dados ou implementação de solução segurança.

## **8. Conclusão**

Neste documento de Arquitetura de Software Distribuído foi aprendido a importância de se ter um relatório que facilite a implementação com visões de diagramas, requisitos funcionais e requisitos não funcionais, também ficou evidente a lição aprendida de com realizar os cenários de teste e poder avaliá-los.

Com esse projeto foi possível aplicar arquitetura adequada para se ter um App consumindo um Microserviço Rest, produzido por um backend estruturado com o padrão arquitetural Hexagonal.

Este projeto poderá evoluir para rodar em sistemas android (APK extensão do executável) e ios (IPA extensão do executável), com isso poder disponibilizar nos repositórios playStore e appleStore, para isto deve-se adotar flutter como linguagem de programação no frontend e poder gerar códigos nativos para os dois tipos de sistemas operacionais.

Lições aprendidas:

1. Produzir documento ASD;
2. Importância do padrão arquitetural;
3. Padrão BFF no frontend;
4. Microserviços rest como backend;
5. Trabalhar orientado aos requisitos, diagramas e cenários de testes;
6. Avaliar os resultados.

Link vídeo apresentação final:

<https://youtu.be/bXATdn1rzwk>