

UNILEÃO  
CENTRO UNIVERSITÁRIO  
CURSO DE TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

BERNARDO VICTOR PIRES DA SILVA

BRENDA MARIA DE SOUSA LIMA

GABRIEL NICACIO FIRMINO

ITALO VINICIUS ARAUJO DA SILVA

IVAN LUÍS RODRIGUES NERY

LUCAS VINICIUS ALENCAR ALVES

**R-EAT**

JUAZEIRO DO NORTE - CE  
2020

BERNARDO PIRES  
BRENDA MARIA DE SOUSA LIMA  
GABRIEL NICACIO FIRMINO  
ITALO VINICIUS ARAUJO DA SILVA  
IVAN LUÍS RODRIGUES NERY  
LUCAS VINICIUS ALENCAR ALVES

**R-EAT**

Trabalho de Conclusão de Curso apresentado à coordenação do Centro Universitário Leão Sampaio como requisito para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.  
Orientador: Prof. Carlos Barreto de Almeida.

JUAZEIRO DO NORTE - CE  
2020

## TERMO DE APROVAÇÃO

## SUMÁRIO

|   |    |
|---|----|
| DOCUMENTO DE VISÃO .....                      | 4  |
| 1 INTRODUÇÃO .....                            | 4  |
| 1.1 OBJETIVO .....                            | 4  |
| 1.2 ESCOPO.....                               | 4  |
| 1.3 DEFINIÇÕES, ACRÔNIMOS E ABREVIACÕES ..... | 4  |
| 2 POSICIONAMENTO .....                        | 5  |
| 2.1 OPORTUNIDADE DE NEGÓCIO .....             | 5  |
| 2.2 RELATO DO PROBLEMA.....                   | 5  |
| 2.3 DECLARAÇÃO DA POSIÇÃO DO PRODUTO .....    | 5  |
| 3 DESCRIÇÃO DOS ENVOLVIDOS E USUÁRIOS.....    | 6  |
| 3.1 RESUMO DOS USUÁRIOS.....                  | 6  |
| 4 CARACTERÍSTICAS DO PRODUTO .....            | 7  |
| 4.1 BATER PONTO DE REFEIÇÃO DIÁRIA .....      | 7  |
| 4.2 RELATÓRIOS DE REFEIÇÕES.....              | 7  |
| 5 RESTRIÇÕES .....                            | 7  |
| 5.1 LINGUAGEM DE PROGRAMAÇÃO .....            | 7  |
| 5.2 PADRÕES DE DESENVOLVIMENTO .....          | 8  |
| 6 REQUISITOS DE DOCUMENTAÇÃO .....            | 8  |
| 6.1 MANUAL DO USUÁRIO .....                   | 8  |
| 6.2 AJUDA ONLINE.....                         | 8  |
| DOCUMENTO DE REQUISITOS.....                  | 9  |
| 1. ESCOPO GERAL DO PRODUTO .....              | 9  |
| 1.1 DESCRIÇÃO DO PRODUTO .....                | 9  |
| 2. REQUISITOS ESPECÍFICOS .....               | 9  |
| 2.1 IDENTIFICAÇÃO DOS REQUISITOS .....        | 9  |
| 2.2 PRIORIDADES DOS REQUISITOS.....           | 10 |

|   |    |
|---|----|
| • Alta:.....  | 10 |
| 2.3 DESCRIÇÃO DOS REQUISITOS .....  | 10 |
| 3. Observações legais, de copyright e outras.....                           | 15 |
| 4 . Diagrama de Casos de Uso .....  | 16 |
| 5. Matriz de Rastreabilidade .....  | 16 |
| 6. Especificação de caso de usos.....                                       | 18 |
| 7. Diagrama de Classe .....   | 20 |
| Introdução.....   | 22 |
| Finalidade.....   | 22 |
| Escopo .....  | 22 |
| Definições, Acrônimos e Abreviações.....                                    | 22 |
| Referências.....  | 23 |
| 1.1 Visão Geral .....   | 24 |
| 2. Representação Arquitetural .....   | 24 |
| 3. Metas e Restrições da Arquitetura.....                                   | 25 |
| 4. Visão de Casos de Uso .....  | 26 |
| 4.1 Realizações de Casos de Uso.....  | 28 |
| 5. Visão Lógica .....   | 28 |
| 5.1 Visão Geral .....   | 28 |
| 5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura ..... | 29 |
| 6. Visão de Processos .....   | 30 |
| 7. Visão de Implantação .....   | 30 |
| 8. Visão da Implementação .....   | 31 |
| 8.1 Visão Geral .....   | 31 |
| 8.2 Camadas.....  | 31 |
| 9. Visão de Dados (opcional).....   | 32 |
| Tamanho e Desempenho .....  | 33 |

|   |    |
|---|----|
| Qualidade .....   | 33 |
| Introdução .....  | 34 |
| 1.1 Finalidade .....  | 34 |
| 1.2 Escopo .....  | 34 |
| 1.3 Público-alvo .....  | 35 |
| 1.4 Terminologia e Acrônimos do Documento .....                           | 35 |
| 1.5 Referências .....   | 36 |
| 1.6 Estrutura do Documento .....  | 36 |
| 2. Missão de Avaliação e Motivação dos Testes .....                       | 36 |
| 2.1 Informações Detalhadas .....  | 37 |
| 2.2 Missão de Avaliação .....   | 37 |
| 2.3 Motivadores dos Testes .....  | 37 |
| Itens-alvo dos Testes .....   | 38 |
| 3. Resumo dos Testes Planejados .....                                     | 39 |
| 3.1 Resumo das Inclusões dos Testes .....                                 | 39 |
| 3.2 Resumo dos Outros Candidatos a Possível Inclusão .....                | 39 |
| Abordagem dos Testes .....  | 39 |
| Catálogos Iniciais de Idéias de Teste e Outras Fontes de Referência ..... | 41 |
| 5.2 Técnicas e Tipos de Teste .....                                       | 41 |
| 5.2.1 Teste de Função .....   | 41 |
| 5.2.2 Teste da Interface do Usuário .....                                 | 42 |
| 5.2.3 Teste de Carga .....  | 42 |
| 5.2.4 Teste de Stress .....   | 43 |
| 5.2.5 Teste de Configuração .....   | 43 |
| Critérios de Entrada e de Saída .....                                     | 44 |
| Plano de Teste .....  | 44 |
| Critérios de Entrada de Plano de Teste .....                              | 44 |

|  |    |
|--|----|
| <i>Critérios de Saída de Plano de Teste</i> .....  | 44 |
| <i>Critérios de Suspensão e de Reinício</i> .....  | 44 |
| Ciclos de Teste.....   | 45 |
| <i>Critérios de Entrada de Ciclo de Teste</i> .....  | 45 |
| <i>Critérios de Saída de Ciclo de Teste</i> .....  | 45 |
| <i>Término Anormal do Ciclo de Teste</i> .....   | 45 |
| Produtos Liberados .....   | 45 |
| Fluxo de Trabalho de Teste .....   | 45 |
| Necessidades Ambientais.....   | 45 |
| Hardware Básico do Sistema.....  | 46 |
| Elementos de Software Básicos do Ambiente de Teste.....  | 46 |
| Ferramentas de Produtividade e de Suporte.....   | 46 |
| Configurações do Ambiente de Teste .....   | 47 |
| Responsabilidades, Perfil da Equipe e Necessidades de Treinamento.....   | 47 |
| Pessoas e Papéis.....  | 47 |
| Procedimentos e Processos de Gerenciamento .....   | 48 |
| Em caso de falha na execução do Plano de Teste, deverão ser documentadas as falhas e/ou modificar partes e fornecer a revisão adequada do documento com a finalidade de continuar com os testes..... | 48 |
| Avaliação dos Produtos Liberados deste Plano de Teste.....   | 48 |
| Relato de Problemas, Seleção de Pessoas para Resolvê-los e Busca de Soluções.....  | 49 |





# DOCUMENTO DE VISÃO

## 1 INTRODUÇÃO

### 1.1 OBJETIVO

O projeto tem como principal objetivo oferecer a empresas um novo método de analisar e organizar os benefícios alimentícios de seus colaboradores, onde será feito o contrato com um ou mais restaurante parceiro para que os usuários do aplicativo tenham uma maior liberdade de escolha. E também como forma de evitar fraudes ou perda de dados, será feito um acordo entre a empresa e o(s) restaurante(s) parceiro(s).

### 1.2 ESCOPO

O projeto visa, uma nova forma de gerar o benefício de vale alimentação dos colaboradores dentro da empresa, ou seja a empresa fará um contrato com um ou mais estabelecimento, sendo assim evitando possíveis fraudes no sistema, e contudo oferecendo aos usuários do aplicativo uma carteira virtual, que será depositado uma quantia determinada pela regra de negócio da empresa para com seus colaboradores, onde o mesmo possa utilizar de seu benefício a fim de escolher o local que será feito a sua refeição.

### 1.3 DEFINIÇÕES, ACRÔNIMOS E ABREVIATURAS

GP - Gerente de Projeto

responsável por planejar e controlar a execução do projeto

MVC - Model View Control

padrão de arquitetura de software

## SCRUM

Significa que o software é construído e entregue em pedaços.

JS - JavaScript

JavaScript é uma linguagem de programação interpretada estruturada

## 2 POSICIONAMENTO

### 2.1 OPORTUNIDADE DE NEGÓCIO

Tendo em vista empresas da região que procuram por restaurantes parceiros, visamos produzir uma aplicação mobile e web para melhorar a organização dos processos e de análise de consumos da empresa. Onde muitas dessas empresas ainda utilizam planilhas ou bloco de notas.

### 2.2 RELATO DO PROBLEMA

|                               |   |
|-------------------------------|---|
| <b>O PROBLEMA DE:</b>         | gastos com cartão físico e fraudes dos colaboradores da empresa   |
| <b>AFETA:</b>                 | Administração da empresa.   |
| <b>IMPACTO E O SEGUINTE:</b>  | Algumas empresas acabam perdendo informação, sobre alguma refeição que foi feita naquele dia, podendo haver falsa informação sobre a refeição permitida por parte do colaborador, ou até mesmo desvio de dinheiro.                    |
| <b>UMA BOA SOLUÇÃO SERIA:</b> | Criar um sistema que faça o cadastro de refeições feitas e um relatório de clientes mensais e semanais, causando assim uma facilidade para as empresas terem um relatório mensal exato das refeições que são servidas no restaurante. |

**Tabela 1** - Declaração do problema.

### 2.3 DECLARAÇÃO DA POSIÇÃO DO PRODUTO

Uma declaração da posição do produto comunica a intenção do aplicativo e a importância do projeto para todo o pessoal interessado.

|                       |  |
|-----------------------|--|
| <b>PARA:</b>          | Empresas que pagam ou querem começar a pagar o benefício aos funcionários.   |
| <b>QUEM:</b>          | Empresas que precisam agilizar o processo de análise de refeições consumidas, podendo evitar o uso de sistemas de planilhas ou blocos de nota.   |
| <b>O REAT:</b>        | É um sistema que tem como objetivo promover a organização de dados, facilitando o cadastro de refeições em larga escala de funcionários, sendo assim gerando e exibindo para o administrador todas as refeições consumidas ao mês. |
| <b>QUE:</b>           | Agiliza o controle e checagem de refeições dos funcionários.   |
| <b>DIFERENTE:</b>     |  |
| <b>NOSSO PRODUTO:</b> | Foi desenvolvido para gerenciar os gastos em refeições dos funcionários, onde o administrador terá as informações da quantidade de refeições feitas e o gasto total ao mês.  |

**Tabela 2** - Declaração da posição do produto.

### 3 DESCRIÇÃO DOS ENVOLVIDOS E USUÁRIOS

#### 3.1 RESUMO DOS USUÁRIOS

| Nome            | Descrição  | Responsabilidade   |
|-----------------|--|--|
| Administrador   | Responsável por manter cadastros e configurações do sistema. | <ul style="list-style-type: none"> <li>- manter colaboradores.</li> <li>- Gerenciar departamento ao qual o colaborador foi designado.</li> </ul> |
| Estabelecimento | Ator responsável por gerar o                                 | <ul style="list-style-type: none"> <li>- Gera o Qr Code.</li> <li>- Validar</li> </ul>   |

|             |   |  |
|-------------|---|--|
|             | QrCode                                  | disponibilidade do colaborador para a refeição.  |
| Funcionário | Ator responsável pela leitura do QrCode | <ul style="list-style-type: none"> <li>- faz a leitura do Qr Code.</li> <li>- Lista refeições feitas.</li> </ul> |

**Tabela 3** - Resumo dos Usuários.

## **4 CARACTERÍSTICAS DO PRODUTO**

### **4.1 BATER PONTO DE REFEIÇÃO DIÁRIA**

Possibilita o funcionário ler o Qr Code para efetuar o pagamento da refeição feita no restaurante parceiro da empresa.

### **4.2 RELATÓRIOS DE REFEIÇÕES**

Permite que o gestor do sistema gere um relatório, apresentando o número de refeições semanais e mensais de registros feitos pela a leitura dos Qr Codes.

## **5 RESTRIÇÕES**

A REAT fornecerá todo o material de consulta relacionado ao escopo do projeto que se fizer necessário, dentro dos prazos acordados com a empresa contratada no início dos trabalhos. sempre que houver um artefato de levantamento de requisito e/ou especificação funcional assinado pela REAT, a homologação do produto deste artefato se dará com base no conteúdo deste documento.

### **5.1 LINGUAGEM DE PROGRAMAÇÃO**

A linguagem de programação será javascript tanto para Front-end quanto para Back-end.

## **5.2 PADRÕES DE DESENVOLVIMENTO**

O padrão utilizado para o desenvolvimento do projeto será o modelo MVC.

Será utilizado a metodologia Scrum abordando a orientação a objetos em sua concepção. Também será utilizado a ferramenta clickup onde o GP poderá realizar o controle das tarefas dos integrantes. Como plataforma de nuvem para o código será utilizado Gitlab.

## **6 REQUISITOS DE DOCUMENTAÇÃO**

### **6.1 MANUAL DO USUÁRIO**

O manual será de simples porém muito objetivo e de fácil entendimento onde o responsável por manusear a software não terá muitos problemas, entretanto irá requerir um bom tempo do usuário, pois será exibido as principais funcionalidades do sistema bem detalhada.

### **6.2 AJUDA ONLINE**

O suporte será realizado pelo e-mail da empresa ou entrando em contato.

# DOCUMENTO DE REQUISITOS

## 1. ESCOPO GERAL DO PRODUTO

### 1.1 DESCRIÇÃO DO PRODUTO

O projeto tem como objetivo gerenciar o benefício alimentício de funcionários de uma empresa, disponibilizando uma carteira virtual onde é depositado o saldo no período acordado com o colaborador. Por meio de nossa tecnologia buscamos facilitar a distribuição e legalização desses recursos além de sanar problemas de improbidade, custo com cartões físicos e manutenções, além de apoiar o crescimento do comércio local oferecendo aos usuários finais uma gama de estabelecimentos parceiros para que eles possam realizar suas refeições livremente.

## 2. REQUISITOS ESPECÍFICOS

### 2.1 IDENTIFICAÇÃO DOS REQUISITOS

Por convenção e para facilitar a identificação dos casos de uso junto aos atores e contextos, a referência é feita de acordo com o esquema abaixo:

**RF - Requisito Funcional:** Define função do sistema de software ou seu componente específico que definem o que o sistema idealmente será capaz de realizar.

**RN - Regras de Negócio:** Os requisitos de domínio são derivados do domínio da aplicação do sistema que podem ser novos requisitos funcionais em si, podem restringir os requisitos funcionais do sistema ou estabelecer como devem ser executadas ações específicas.

**RNF -Requisitos Não Funcionais:** São os requisitos relacionados ao uso da aplicação e termos de desempenho, usabilidade, segurança, disponibilidade, manutenibilidade e tecnologia envolvidas.

## 2.2 PRIORIDADES DOS REQUISITOS

- **Alta:**

São requisitos no qual o sistema precisa para entrar em funcionamento. Os requisitos essenciais são de suma importância para o funcionamento do software.

- **Média:**

É o requisito no qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.

- **Baixa:**

É o requisito que não compromete as funcionalidades básicas do sistema, ou seja, o sistema pode funcionar de forma razoável sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

## 2.3 DESCRIÇÃO DOS REQUISITOS

Este projeto de desenvolvimento de software exige que seja identificado requisitos do tipo: Requisitos Funcionais, Requisitos de Negócio (domínio) e Requisitos Não Funcionais.

### 2.3.1 REQUISITOS FUNCIONAIS

| SIGLA | NOME | DESCRIÇÃO | PRIORIDADE | DEPENDE DE |
|-------|------|-----------|------------|------------|
|-------|------|-----------|------------|------------|

|        |                        |  |       |                  |
|--------|------------------------|--|-------|------------------|
| RF-001 | Manter Usuário         | Tem o objetivo de criar, consultar, atualizar e deletar todos os usuários vinculados a uma empresa que serão cadastrado no sistema.                      | Alta  | RF-003<br>RF-002 |
| RF-002 | Manter Restaurante     | Cadastro um estabelecimento que oferecerá serviço no sistema, esse requisito tem função de criar, consultar, atualizar e deletar.                        | Alta  |                  |
| RF-003 | Manter Empresa         | Cadastra empresas que oferecem benefício aos seus colaboradores por meio do sistema, esse requisito tem função de criar, consultar, atualizar e deletar. | Alta  |                  |
| RF-004 | Listar Parceiros       | O sistema deverá apresentar ao usuário quais estabelecimentos estão disponíveis para uso do benefício  | Média | RF-002           |
| RF-005 | Adicionar Saldo        | Esta função permite que a empresa faça o lançamento do benefício para seus colaboradores mediante suas restrições.                                       | Alta  | RF-001           |
| RF-006 | Indicar Valor Sugerido | A empresa deve indicar o preço sugerido de uso diário para seus colaboradores  |       | RF-001<br>RF-005 |



|        |                                      |  |      |                                      |
|--------|--------------------------------------|--|------|--------------------------------------|
| RF-007 | Ver Valor Sugerido                   | O usuário (colaborador) deve ver o valor sugerido de uso diário.   |      | RF-005<br>RF-006<br>RF-001           |
| RF-008 | Ver Saldo                            | O usuário (colaborador) tem acesso ao valor atual do benefício em sua carteira.  | Alta | RF-005<br>RF-006<br>RF-001           |
| RF-009 | Efetuar Pagamento                    | O usuário (colaborador) irá escanear o Qr code gerado pelo estabelecimento ou aprovar a solicitação de pagamento do restaurante.   | Alta | RF-005<br>RF-002<br>RF-001           |
| RF-010 | Receber Pagamento                    | Esta função permite que os restaurantes gerem um QR code ou digite o id do colaborador a fim de receber o pagamento de acordo com o consumo dos usuários em seu estabelecimento. | Alta | RF-005<br>RF-002<br>RF-001           |
| RF-011 | Histórico de Transações Pagamentos   | Permite ao usuário (colaborador) verificar detalhes de seus últimos pagamentos.  | Alta | RF-009<br>RF-010<br>RF-001<br>RF-002 |
| RF-012 | Histórico de Transações Recebimentos | Permite ao usuário (empresa) verificar detalhes de seus últimos recebimentos.  | Alta | RF-009<br>RF-010<br>RF-001<br>RF-002 |

|        |                                   |   |      |                                      |
|--------|-----------------------------------|---|------|--------------------------------------|
| RF-013 | Emitir Comprovante de Pagamento   | Após o ato de pagamento o sistema deverá emitir um comprovante com dados do pagante, cobrador, serviço, valor, etc.   | Alta | RF-001<br>RF-002<br>RF-009<br>RF-010 |
| RF-014 | Emitir Comprovante de Recebimento | Após o ato de recebimento o sistema deverá emitir um comprovante com dados do pagante, cobrador, serviço, valor, etc. | Alta | RF-001<br>RF-002<br>RF-009<br>RF-010 |

**Tabela 1** – Requisitos funcionais.

### 2.3.2 REQUISITOS DE NEGÓCIO (DOMÍNIO)

| SIGLA  | NOME                      | DESCRIÇÃO   | PRIORIDADE |
|--------|---------------------------|---|------------|
| RN-001 | Restrição de Restaurantes | Os usuários cadastrados (colaboradores) só podem realizar refeições nos restaurantes parceiros da Reat. | Alta       |
| RN-002 | Expirar Benefício         | O saldo em carteiras é renovado mensalmente, assim valores do mês anterior são removidos.               | Alta       |

|        |                                  |  |      |
|--------|----------------------------------|--|------|
| RN-003 | Abatimento de Preço para Empresa | Se houver valores expirados do saldo dos colaboradores no ciclo anterior, esse valor é abatido, gerando um desconto no valor que será pago no novo lançamento. | Alta |
|--------|----------------------------------|--|------|

**Tabela 2** – Requisitos de negócio.

### 2.3.3 Requisitos não funcionais

| SIGLA   | NOME                       | DESCRIÇÃO  | PRIORIDADE | CATEGORIA           |
|---------|----------------------------|--|------------|---------------------|
| RNF-001 | Diferentes Ambientes       | O sistema atua em dois ambientes distintos ,aplicação mobile para o (usuário e operadores) e aplicação web(administradores). | Alta       | Portabilidade       |
| RNF-002 | Promover Facilidade de Uso | A aplicação necessita de uma interface eficiente e focada na experiência do usuário.   | Alta       | Usabilidade         |
| RNF-03  | Verificar Usuario          |  | Alta       | Segurança de Acesso |

|        |                       |  |      |                           |
|--------|-----------------------|--|------|---------------------------|
| RNF-04 | Facilitar Aprendizado | O sistema deve ser intuitivo e possibilitar que novos usuários consigam identificar as principais funções com pouca dificuldade.           | Alta | Facilidade de Aprendizado |
| RNF-05 | Permitir Mudanças     | O sistema como um todo deve ser construído para permitir alterações e possíveis implementações sem afetar diretamente o seu funcionamento. | Alta | Manutenibilidade          |

**Tabela 3** – Requisitos não funcionais.

**Categoria:** Segurança de Acesso, Interoperabilidade, Confiabilidade, Facilidade de Operação, Facilidade de Aprendizado, Atratividade, Eficiência em Relação ao Tempo, Eficiência em Relação a Recursos, Manutenibilidade, Portabilidade.

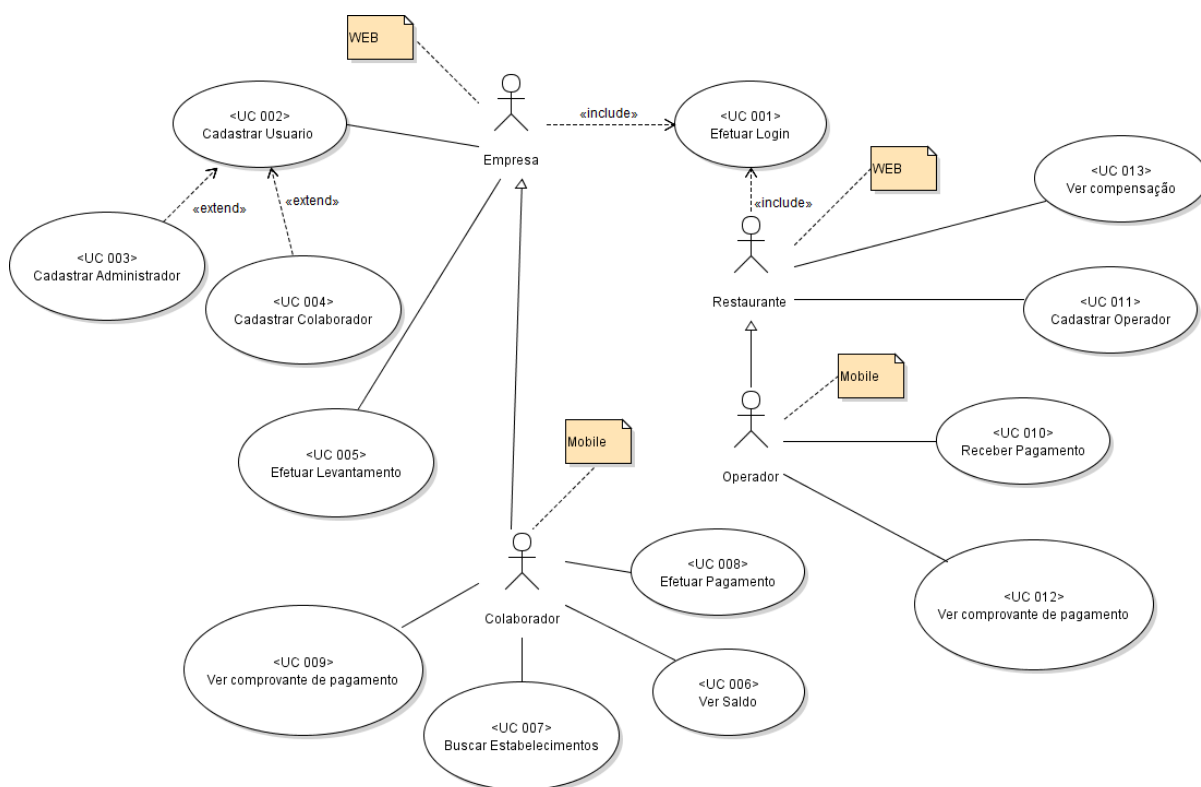
### 3. Observações legais, de copyright e outras

Art. 2º O regime de proteção à propriedade intelectual de programa de computador é o conferido às obras literárias pela legislação de direitos autorais e conexos vigentes no País, observado o disposto nesta Lei.

§ 2º Fica assegurada a tutela dos direitos relativos a programa de computador pelo prazo de cinquenta anos, contados a partir de 1º de janeiro do ano subsequente ao da sua publicação ou, na ausência desta, da sua criação.

§ 5º Inclui-se dentre os direitos assegurados por esta Lei e pela legislação de direitos autorais e conexos vigentes no País aquele direito exclusivo de autorizar ou proibir o aluguel comercial, não sendo esse direito exaurível pela venda, licença ou outra forma de transferência da cópia do programa.

#### 4 . Diagrama de Casos de Uso



+

#### 5. Matriz de Rastreabilidade



## 6. Especificação de caso de usos

| DATA                       | VERSÃO   | DESCRIÇÃO     |  | AUTORES        |
|----------------------------|--|---------------|--|----------------|
|                            | 1.0  | Efetuar Login |  | Bernardo Pires |
| <b>STATUS</b>              | ( x ) Pendente ( ) Revisado ( ) Realizado<br>( ) Excluído/Cancelado  |               |  |                |
| <b>USE CASE - UC</b>       | UC 001   |               |  |                |
| <b>REQUISITO FUNCIONAL</b> | Efetuar Login  |               |  |                |
| <b>DESCRIÇÃO DO UC</b>     | Caso de uso responsável por conceder acesso único mediante ao tipo de usuário além de certificar demais ações no sistema   |               |  |                |
| <b>ATORES ENVOLVIDOS</b>   | Administrador, Colaborador, Restaurante, Operador  |               |  |                |
| <b>PRÉ-CONDIÇÃO</b>        | Usuários cadastrados, Realizar Login   |               |  |                |
| <b>PÓS-CONDIÇÃO</b>        | Cadastro Realizado, Uso das demais funções   |               |  |                |
| <b>FLUXO PRINCIPAL</b>     | <b>AÇÕES DO ATOR</b>   |               | <b>AÇÕES DO SISTEMA</b>  |                |
|                            | 1. O usuário do tipo Operador deverá realizar login.<br>2. O usuário do tipo Administrador deverá realizar login.<br>3. O usuário do tipo Colaborador deverá realizar login.<br>4.<br>5. |               | 1. Verificar credenciais de login<br>2. Certificar usuário.<br>3. Permitir ou não o acesso<br>4. Validar Login |                |
| <b>FLUXO ALTERNATIVO</b>   | <b>AÇÕES DO ATOR</b>   |               | <b>AÇÕES DO SISTEMA</b>  |                |
|                            | 1.<br>2.<br>3.   |               | 1.<br>2.<br>3.   |                |

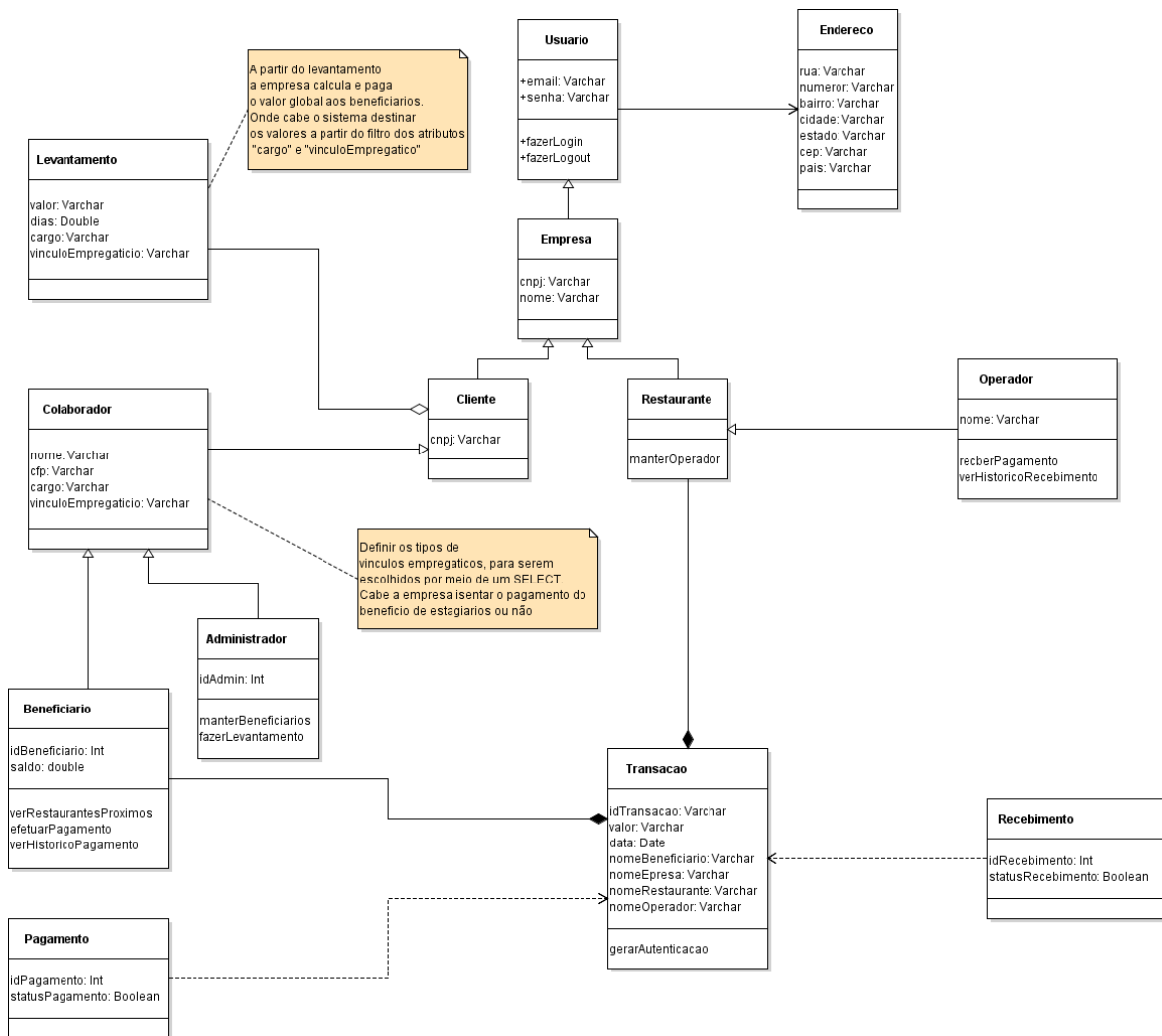
|                         |   |   |
|-------------------------|---|---|
|                         | 4.<br>5.  | 4.<br>5.  |
| <b>FLUXO DE EXCEÇÃO</b> | <b>AÇÕES DO ATOR</b>                                  | <b>AÇÕES DO SISTEMA</b>   |
|                         | 1. Solicitar novas credencial<br>2.<br>3.<br>4.<br>5. | 1. Alterar credenciais no banco.<br>2. Informar ao usuário.<br>3.<br>4.<br>5. |
| <b>PROTÓTIPO</b>        |   |   |

| <b>DATA</b>                | <b>VERSÃO</b>  | <b>DESCRIÇÃO</b>   | <b>AUTORES</b> |
|----------------------------|--|--|----------------|
|                            | 1.0  | Efetuar Levantamento   | Bernardo Pires |
| <b>STATUS</b>              | <input checked="" type="checkbox"/> Pendente <input type="checkbox"/> Revisado <input type="checkbox"/> Realizado<br><input type="checkbox"/> Excluído/Cancelado |  |                |
| <b>USE CASE - UC</b>       | UC 005   |  |                |
| <b>REQUISITO FUNCIONAL</b> | Efetuar Levantamento   |  |                |
| <b>DESCRIÇÃO DO UC</b>     | Caso de uso responsável pelo lançamento do benefício por meio de saldo na carteira virtual dos colaboradores.  |  |                |
| <b>ATOES ENVOLVIDOS</b>    | Administrador, Colaborador   |  |                |
| <b>PRÉ-CONDIÇÃO</b>        |  |  |                |
| <b>PÓS-CONDIÇÃO</b>        | Saldo adicionado nas contas  |  |                |
| <b>FLUXO PRINCIPAL</b>     | <b>AÇÕES DO ATOR</b>   | <b>AÇÕES DO SISTEMA</b>  |                |
|                            | 1. Selecionar valor referente ao tipo de colaborador<br>2. Definir dias abonados<br>3. Definir prazo de expiração<br>4. Definir sugestão de uso diário           | 1. Calcular valor total<br>2. Direcionar saldo a carteira do colaborador<br>3. Certificar levantamento |                |



|                          |                            |                            |
|--------------------------|----------------------------|----------------------------|
| <b>FLUXO ALTERNATIVO</b> | <b>AÇÕES DO ATOR</b>       | <b>AÇÕES DO SISTEMA</b>    |
|                          | 1.<br>2.<br>3.<br>4.<br>5. | 1.<br>2.<br>3.<br>4.<br>5. |
| <b>FLUXO DE EXCEÇÃO</b>  | <b>AÇÕES DO ATOR</b>       | <b>AÇÕES DO SISTEMA</b>    |
|                          | 1.<br>2.<br>3.<br>4.<br>5. | 1.<br>2.<br>3.<br>4.<br>5. |
| <b>PROTÓTIPO</b>         |                            |                            |

## 7. Diagrama de Classe



# DOCUMENTO DE ARQUITETURA DE SOFTWARE

## **Introdução**

Este documento fornece uma visão arquitetural abrangente do sistema, usando diversas visões de arquitetura para representar diferentes aspectos do sistema. Ele pretende capturar e transmitir as decisões arquiteturas significativas que foram tomadas em relação ao sistema.

## *Finalidade*

O projeto tem como objetivo gerenciar o benefício alimentício de funcionários de uma empresa, disponibilizando uma carteira virtual onde é depositado o saldo no período acordado com o colaborador. Por meio de nossa tecnologia buscamos facilitar a distribuição e legalização desses recursos além de sanar problemas de improbidade, custo com cartões físicos e manutenções, além de apoiar o crescimento do comércio local oferecendo aos usuários finais uma gama de estabelecimentos parceiros para que eles possam realizar suas refeições livremente.

## *Escopo*

Este Documento foi gerado diretamente a partir do Modelo de Análise e Design do Reat implementado no RUP aliado aos processos ágeis do SCRUM. A maior parte das seções foi extraída do template fornecido pela IBM, implementado pela equipe San Corporation como em uma metodologia híbrida entre SCRUM e RUP aliado ao gabarito de Documento de Arquitetura de Software.

## *Definições, Acrônimos e Abreviações*

1. MVC ( Model View Controller) : Arquitetura de software utilizada em

sistemas que desejam separar a modelagem de dados, interface e processamento de requisições em camadas independentes.

2. **SCRUM**: Scrum é uma **metodologia ágil** para gestão e planejamento de projetos de software. No Scrum, os projetos são divididos em ciclos (tipicamente mensais) chamados de **Sprints**. O **Sprint** representa um Time Box dentro do qual um conjunto de atividades deve ser executado. **Metodologias ágeis** de desenvolvimento de software são iterativas, ou seja, o trabalho é dividido em iterações, que são chamadas de Sprints no caso do Scrum.
3. **MongoDB**: **MongoDB** é um software de banco de dados orientado a documentos livre, de código aberto e multiplataforma, escrito na linguagem C++. Classificado como um programa de banco de dados NoSQL, o MongoDB usa documentos semelhantes a JSON com esquemas. É desenvolvido pela MongoDB Inc. e publicado sob uma combinação da GNU Affero General Public License e Licença Apache.
4. **NoSQL**: **NoSQL** (originalmente se referindo a "no SQL": "não SQL" ou "não relacional", posteriormente estendido para **Not Only SQL** - Não Somente SQL) é um termo genérico que representa os bancos de dados não relacionais. Uma classe definida de banco de dados que fornecem um mecanismo para armazenamento e recuperação de dados que são modelados de formas diferentes das relações tabulares usadas nos bancos de dados relacionais.
5. **Banco relacional** : Um **banco de dados relacional** é um banco de dados que modela os dados de uma forma que eles sejam percebidos pelo usuário como tabelas, ou mais formalmente relações.
6. **Modelo Relacional**: O **modelo relacional** é um modelo de dados representativo (ou de implementação), adequado a ser o modelo subjacente de um Sistema Gerenciador de Banco de Dados (SGBD), que se baseia no princípio de que todos os dados estão armazenados em tabelas (ou, matematicamente falando, relações). Toda sua definição é teórica e baseada na lógica de predicados e na teoria dos conjuntos.

## Referências

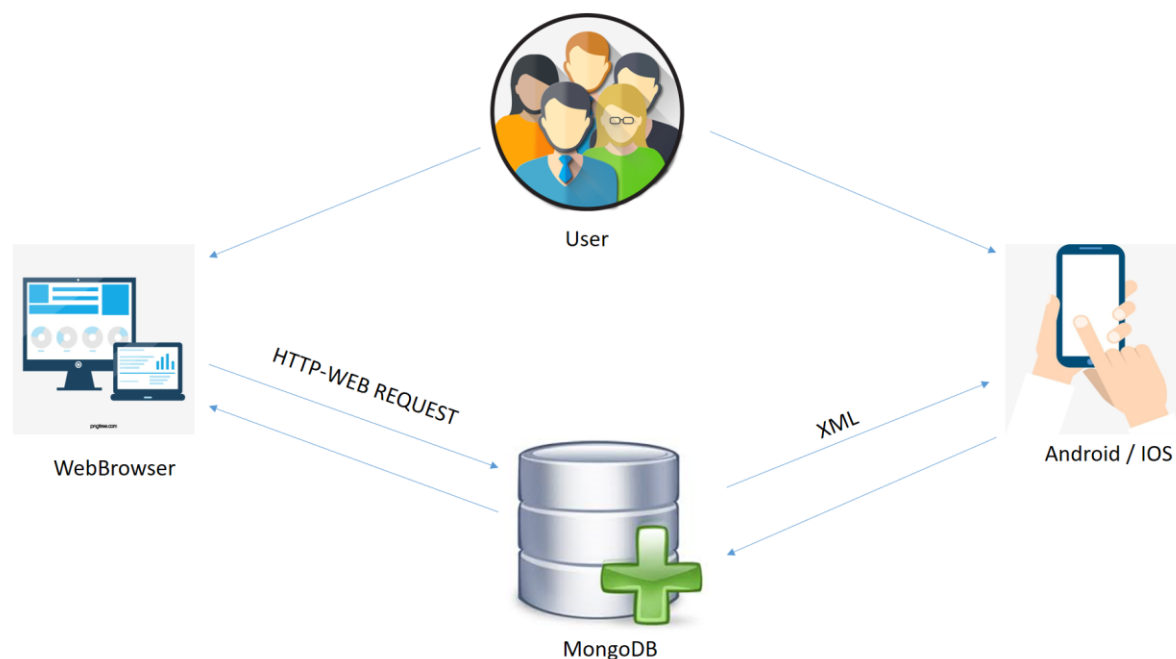
*[Esta subseção fornece uma lista completa dos documentos mencionados em outra parte do **Documento de Arquitetura de Software**. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]*

## 1.1 Visão Geral

Este documento está organizado em tópicos relacionados às diferentes visões arquiteturas, tais como as de diagramas de classes, de caso de uso, de entidade-relacional e de sequência. Sua sequência lógica é:

1. Introdução
2. Representação da Arquitetura
3. Restrição de Arquitetura e Metas
4. Visão de Casos de Uso
5. Visão Lógica
6. Visão de Implementação
7. Visão de Dados

## 2. Representação Arquitetural



Este documento apresenta a arquitetura como uma série de visualizações; visualização caso de uso, visualização lógica, visualização do processo e visualização da implementação. Não existe uma visualização de implementação separada descrita neste documento. Elas são visualizações em um modelo UML (linguagem de modelagem unificada) subjacente desenvolvido utilizando o Draw.io.

Na arquitetura apresentada acima mostra o funcionamento da arquitetura Reat a qual segue os seguintes passos:

- 1- O usuário executa o aplicativo em seu smartphone onde automaticamente aciona o *controller* para iniciar a *view* inicial para o usuário.
- 2- O smartphone deve estar conectado a internet para que assim possa ser realizado o login utilizando uma das três formas disponíveis para o usuário, onde são: 1 – Realizar login mediante cadastro de um e-mail, onde o usuário irá digitar alguns dados solicitados onde serão armazenados em nosso Webservice com a devida segurança, 2 – o usuário também poderá acessar o aplicativo Reat mediante a um acesso que empresa disponibiliza para o usuário.
- 3- O servidor terá todas as informações de cadastros Reat sendo executadas nele.
- 4- O aplicativo estará dividida em 3 camadas em um padrão determinado como MVC, propiciando assim facilidade na hora da atualização de dados no próprio aplicativo, além da implementação de novas funcionalidades.

### 3. Metas e Restrições da Arquitetura

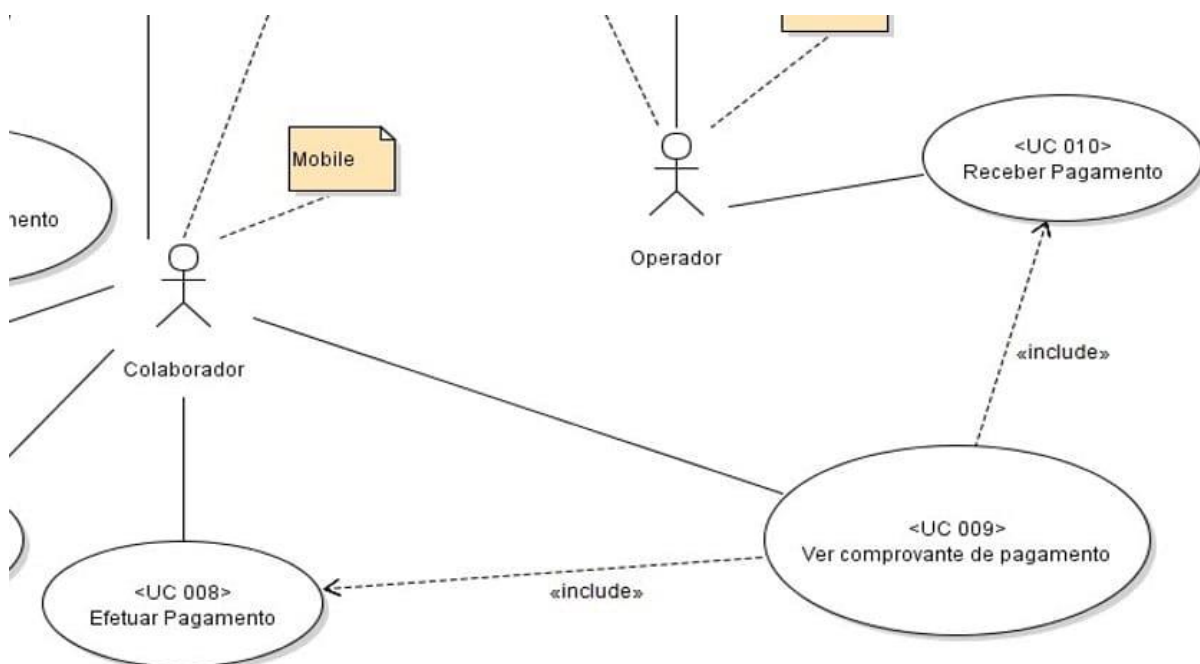
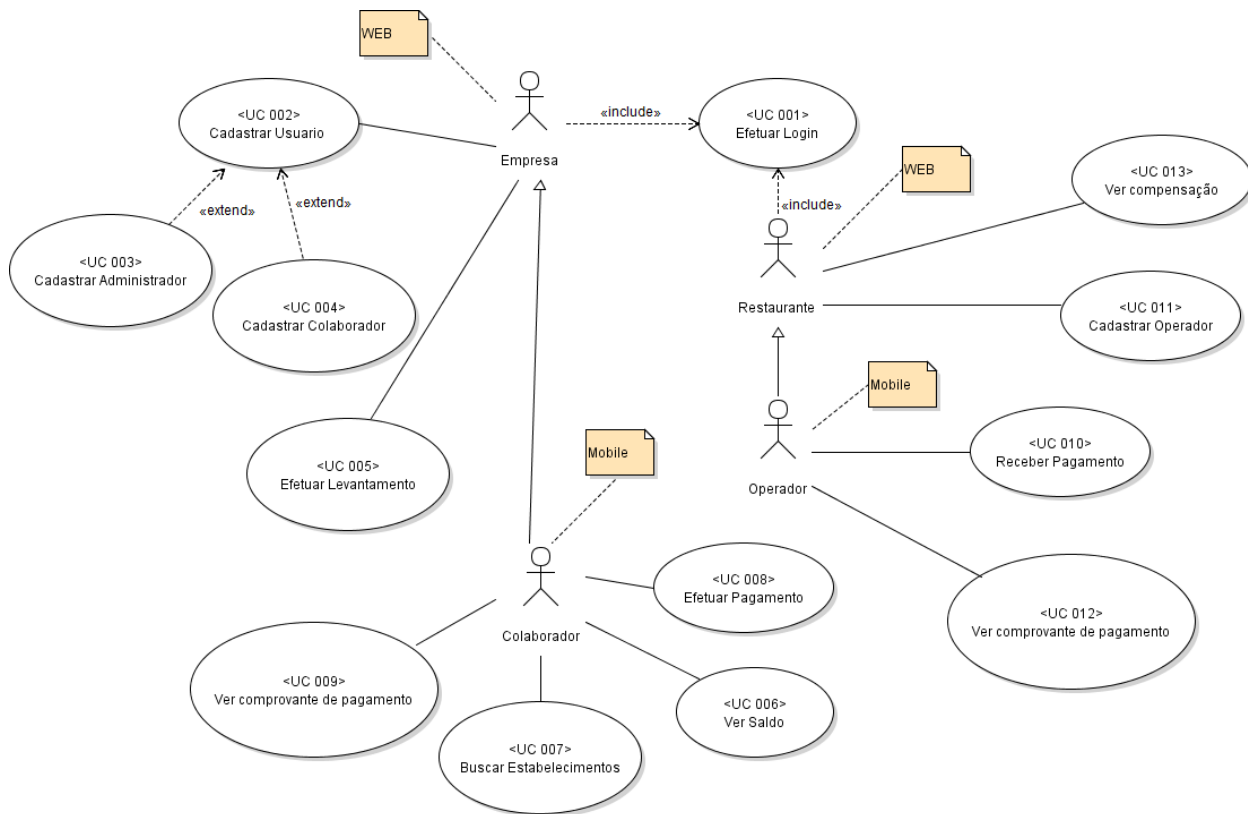
Existem alguns requisitos chave e restrições do sistema que têm um relacionamento significativo com a arquitetura. São eles:

1. O sistema Reat tem como objetivo gerenciar o benefício alimentício de funcionários de uma empresa, disponibilizando uma carteira virtual onde é depositado o saldo no período acordado com o colaborador. Por meio de nossa tecnologia buscamos facilitar a distribuição e legalização desses recursos além de sanar problemas de improbidade, custo com cartões físicos e manutenções, além de apoiar o crescimento do comércio local oferecendo aos usuários finais uma gama de estabelecimentos parceiros para que eles possam realizar suas refeições livremente.
2. Todas as funcionalidades do RH (Empresa) e Restaurante devem estar disponível nos PCs locais e em PCs remotos com conexões dial-up à Internet.
3. O **Reat** deve assegurar a proteção completa dos dados contra acesso não-autorizado. Todos os acessos remotos estão sujeitos a identificação do usuário e controle de senha.

4. O **Reat** será implementado como um sistema cliente/servidor. A parte cliente reside em PCs e a parte servidor deve operar no Servidor **<a definir>**
5. Restrições do projeto:
  - Restrição de Restaurantes: onde os usuários cadastrados (colaboradores) só podem realizar refeições nos restaurantes parceiros da Reat.
  - Expirar Benefício :O saldo em carteiras é renovado mensalmente, assim valores do mês anterior são removidos.
  - Abatimento de Preço para Empresa: Se houver valores expirados do saldo dos colaboradores no ciclo anterior, esse valor é abatido, gerando um desconto no valor que será pago no novo lançamento.

#### 4. Visão de Casos de Uso

Essa seção apresenta os casos de uso que tem a maior importância sobre a arquitetura do sistema, ela ilustra pontos que são realizados de forma mais complexa pelo sistema.





#### 4.1 Realizações de Casos de Uso

##### **UC06 -Efetuar Pagamento.**

O caso de uso *Efetuar Pagamento* é onde o colaborador vai realizar a quitação do serviço utilizado no restaurante.

.

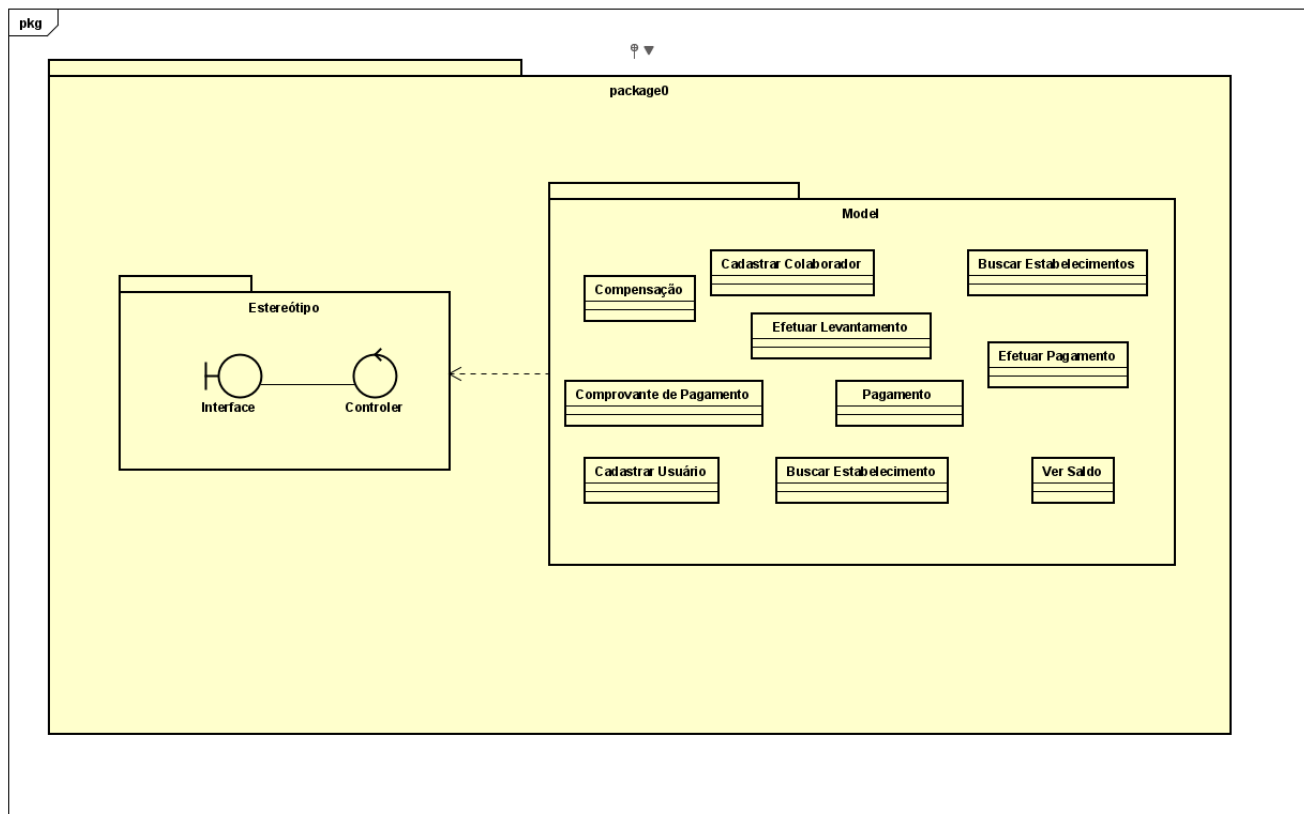
## 5. Visão Lógica

### 5.1 Visão Geral

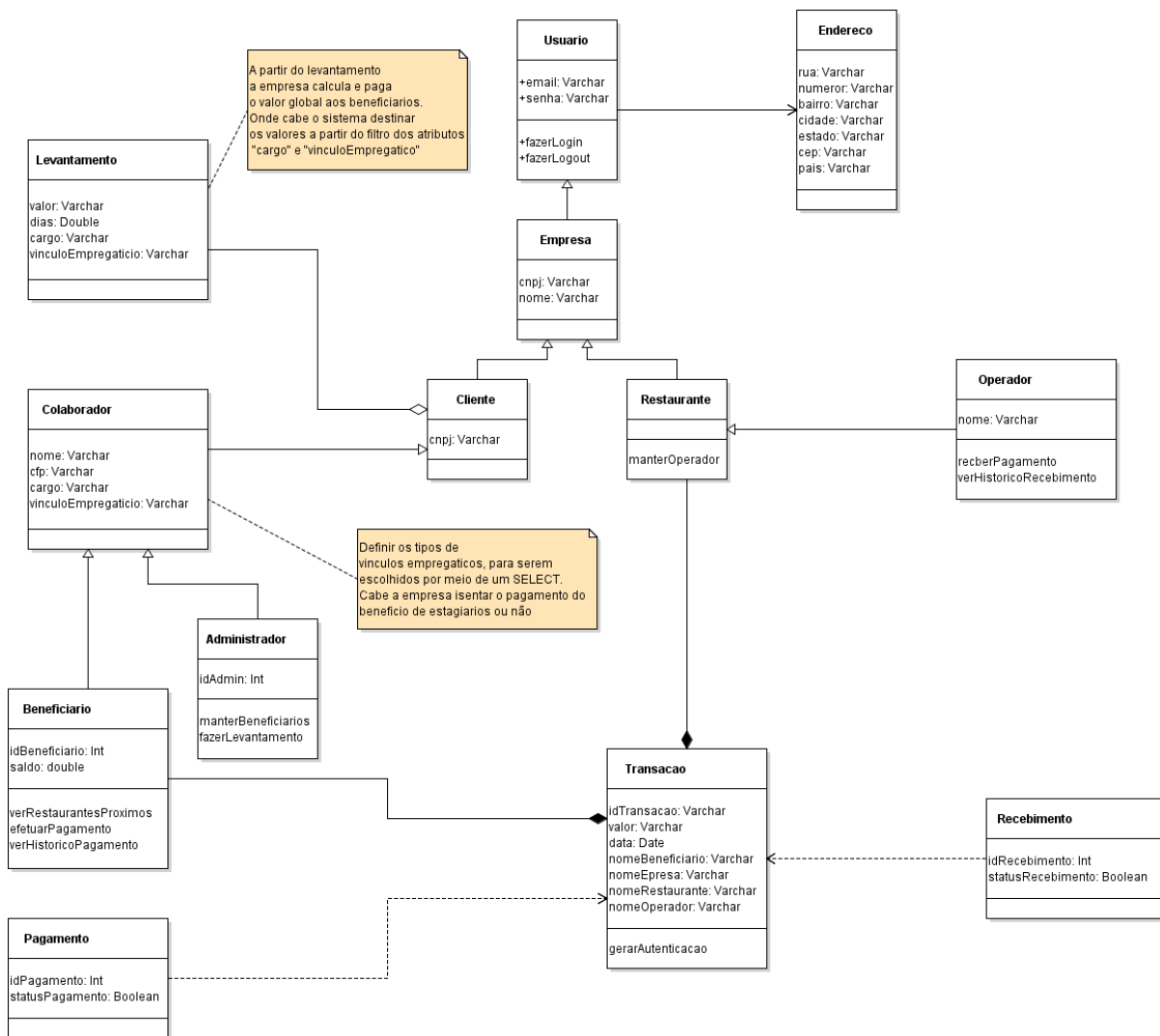
Esta subseção descreve toda a decomposição do modelo de design em termos de camadas e de hierarquia de pacotes. O programa tem os seguintes pacotes.

- **Controller:** Contém as classes que recebem as requisições feitas na interface do usuário.
- **Model:** Contém as classes que recebem as requisições do controller e realizam requisições.
- **View:** Pacote que contém todas as views que irão interagir diretamente com os usuários.

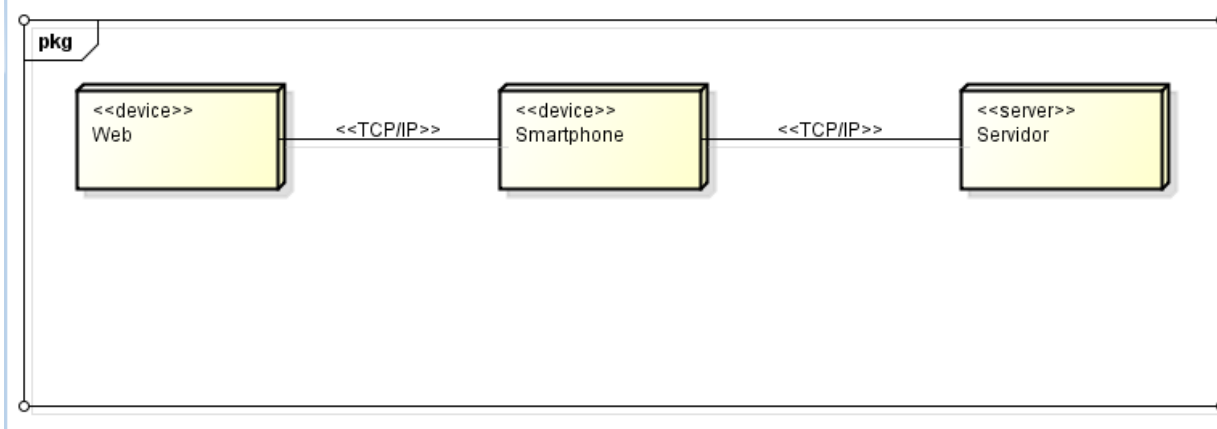
## 5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura



## 6. Visão de Processos

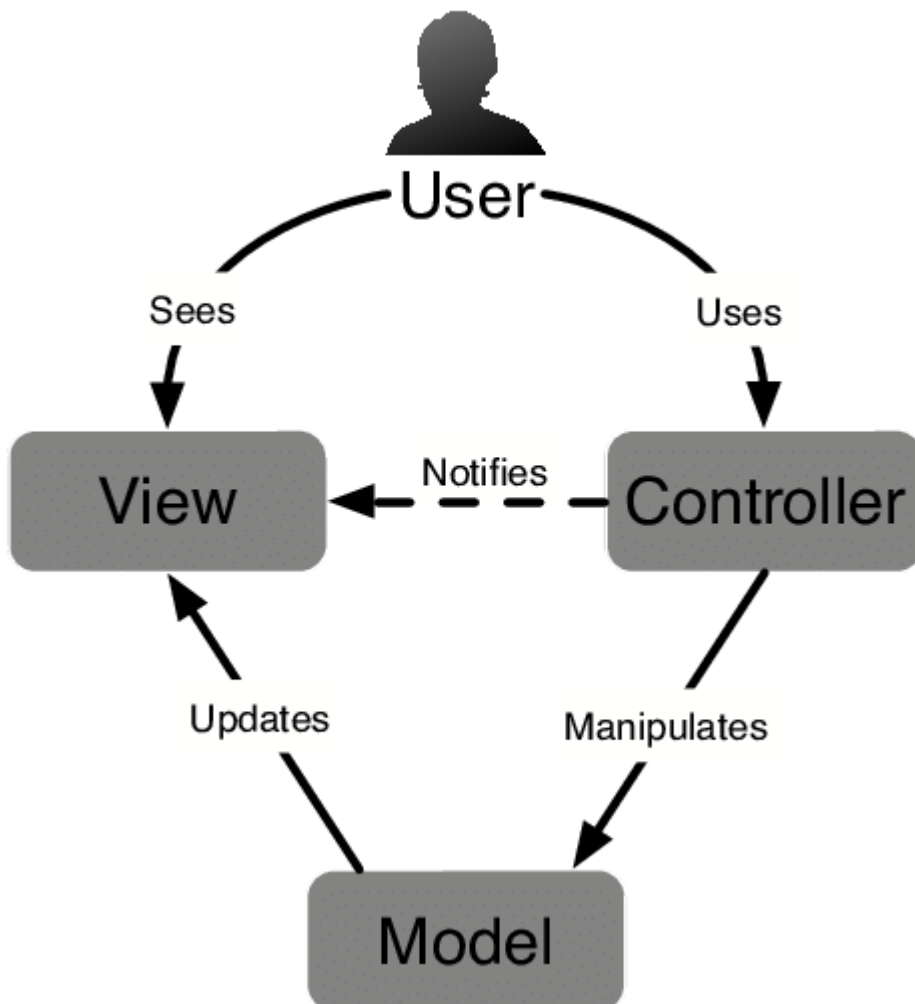


## 7. Visão de Implantação



## 8. Visão da Implementação

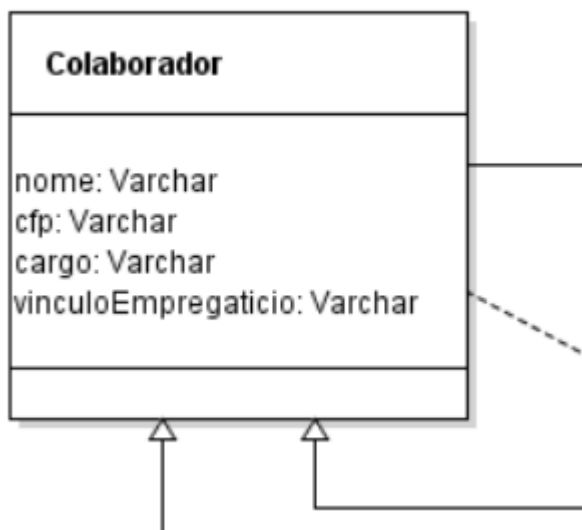
### 8.1 Visão Geral



### 8.2 Camadas

- VIEW
- Login.js
- Senha.js
- MODEL
- CONTROLLER

## 9. Visão de Dados (opcional)



# mongoDB

Foi escolhido como banco de dados o mongoDB que usa o modelo noSQL para guardar os registros dos usuários referentes ao seu login no sistema.

| Dicionário de dados |           |           |
|---------------------|-----------|-----------|
| Tabela              | Atributos | Descrição |

|    |             |  |  |
|----|-------------|--|--|
| 01 | Colaborador | Nome VARCHAR(45)<br>Cpf : VARCHAR(45)<br>Cargo: VARCHAR(45)<br>VinculoEmpregaticio:<br>VARCHAR(45) | Possui os registros de login dos usuários. |
|----|-------------|--|--|

### **Tamanho e Desempenho**

O aplicativo será construído inicialmente para smartphones com sistema operacional Android visando seu melhor desempenho nos mais variados tipos de aparelhos, adequando-se a versões do android que contemplem da versão 4.4 e seus sucessores. Em memória, ele ocupará um espaço de 80 MB. O sistema suportará uma média de 100 usuários logados simultaneamente sem que haja comprometimento em seu desempenho. A arquitetura escolhida suporta o dimensionamento dos requisitos de tempo, conforme descrito no Documento de Requisitos: Ao selecionar um botão o resultado exato deve aparecer em no mínimo 10 segundos (RNF01).

### **Qualidade**

O aplicativo Reat é um aplicativo de vale alimentação inicialmente com poucas funcionalidades, onde com o seu crescimento no mercado, terá novas funcionalidades implementadas, onde terá uma maior facilidade nesse procedimento, pois no seu desenvolvimento será utilizada a arquitetura MVC, onde permitirá esse acréscimo de funcionalidades com agilidade além da sua manutenção. O sistema será web e Android .

# DOCUMENTO DE TESTE DE SOFTWARE

## Introdução

### 1.1 Finalidade

A finalidade do Plano de Teste de Iteração é reunir todas as informações necessárias ao planejamento e ao controle do esforço de teste referente a uma iteração específica. Ele descreve a abordagem dada ao teste do software e é o plano de nível superior gerado e usado pelos gerentes para coordenar o esforço de teste.

Este *Plano de Teste* referente ao Reat suporta os seguintes objetivos:

- Elucidar o testador quanto as ferramentas necessárias para a realização dos testes cabíveis;
- Identificar erros e assegurar suas correções tanto no sistema quanto na sua documentação;
- Identificar os requisitos para o testador realizar os devidos testes com a finalidade de melhorias;
- Mostrar inúmeras formas aos testadores para a utilização dos testes;
- Identificar os resultados dos testes
- Explicar os recursos necessários para a execução dos testes;
- Adquirir estimativa dos esforços de teste;
- Verificar possíveis riscos por meios dos testes;

### 1.2 Escopo

O projeto Reat visa, uma nova forma de gerar o benefício de vale alimentação dos colaboradores dentro da empresa, ou seja a empresa fará um contrato com um ou mais estabelecimento, sendo assim evitando possíveis fraudes no sistema, e contudo oferecendo aos usuários do aplicativo uma carteira virtual, que será depositado uma quantia determinada pela regra de negócio da

empresa para com seus colaboradores, onde o mesmo possa utilizar de seu benefício a fim de escolher o local que será feito a sua refeição. O Reat será online onde vai ter a necessidade de se fazer testes de unidade, integração do sistema. Os testes unitários e de integração vão lidar com a qualidade funcional, base de dados, interface gráfica e controle de acesso, já o teste do sistema irá verificar o funcionamento, desempenho, e a usabilidade do sistema.

Durante todo o processo, poderá haver mudanças de acordo com o ciclo de vida do software, como inclusões dos requisitos ou até mesmo exclusões

### 1.3 Público-alvo

Este documento de plano de teste é destinado aos seguintes envolvidos: o Gerente de Testes, Analista de Sistema, Testador e Gerente de Projeto, com o intuito de orientá-los nos testes a serem desenvolvidos. Onde cada um tem sua responsabilidade e sua parte ao todo, focando nos objetivos do projeto Reat.

### 1.4 Terminologia e Acrônimos do Documento

As definições utilizadas neste documento do Reat serão abordadas posteriormente no glossário.

| <b>Termo</b> | <b>Descrição</b>                           |
|--------------|--|
| RF           | Requisito Funcional                        |
| RNF          | Requisito Não Funcional                    |
| SGBD         | Sistema de Gerenciamento de Banco de Dados |



### 1.5 *Referências*

- Documento de Visão: Documento de Visão.\*
- Documento de Requisitos: DOCUMENTO DE REQUISITOS.\*
- Documento de Arquitetura: Documento de Arquitetura.\*

### 1.6 *Estrutura do Documento*

- As seções seguintes estão organizadas como descritas abaixo:

Parte 1 – Introdução:

Parte 2: Missão de Avaliação e Motivação dos testes.

Parte 3: Itens-alvo dos Testes.

Parte 4: Resumo dos Testes Planejados.

Parte 5: Abordagem dos Testes:

Parte 6: Catálogos Iniciais de Ideias de Teste e outras Fontes de Referência.

Parte 7: Critérios de Entrada e de Saída.

Parte 8: Produtos Liberados.

Parte 9: Fluxo de Trabalho de Teste.

Parte 10: Necessidades Ambientais.

Parte 11: Responsabilidades, Perfil da Equipe e Necessidades de Treinamento.

Parte 12: Procedimento e Processos de Gerenciamento.

## **2. Missão de Avaliação e Motivação dos Testes**

O Teste de software estabelece de um procedimento realizado pelo testador, do qual tem como objetivo detectar possíveis falhas que venham apresentar durante

todo o processo no sistema, sendo assim passando a ser corrigidos o mais rápido possível, para obter uma maior qualidade no produto em desenvolvimento podendo ocorrer maior satisfação do cliente com o mesmo, analisar ou verificar se todos os requisitos estão de acordos, e se o que foi pedido está de acordo com o resultado final esperado.

## 2.1 *Informações Detalhadas*

O principal problema está nos gastos com cartão físico e fraudes dos colaboradores da empresa, afetando principalmente a administração da empresa. Essas empresas acabam perdendo informações sobre a alimentação no ambiente, fazendo com o que colaborador possa fazer mais de uma refeição ao longo do dia, o que gera o levantamento incoerente dos dados naquele dia.

Uma boa solução seria criar um sistema que faça o cadastro de refeições feitas e um relatório de clientes mensais e semanais, causando assim uma facilidade para as empresas terem um relatório mensal exato das refeições que são servidas no restaurante, aumentando o controle administrativo.

## 2.2 *Missão de Avaliação*

O teste de software tem como objetivo:

- Localizar o maior número de erros possíveis, e em seguida corrigi-los com habilidade, diminuindo ao máximo possível os custos que foram nos reparos anteriores.
- Conferir se o sistema está seguindo corretamente o que a arquitetura impõe a serem seguidas: verificando se toda as documentações do sistema estão de acordo com a necessidade do cliente, propondo maior qualidade nos processos e no produto final.

## 2.3 *Motivadores dos Testes*

O teste de software é muito importante, para se ter um bom sucesso desse projeto, pois ele contém uma grande responsabilidade pela qualidade do produto e garante que todo o projeto está de acordo com a documentação descrita. O testador verifica e executa todas as suas tarefas com o intuito de corrigir possíveis falhas que possa haver no projeto, evitando retrabalhos para o futuro. Com a conclusão do projeto ele deve estar de forma correta e dentro do prazo e como foi planejado, trazendo uma boa satisfação ao cliente.

## **Itens-alvo dos Testes**

A listagem abaixo identifica os itens – software, hardware e elementos de suporte do produto – que foram identificados como alvo dos testes. Essa lista representa os itens que serão testados.

### ***3.1 Teste de Configuração***

- Fazer uma análise e depois verificar se todas as funcionalidades do sistema está funcionando corretamente em pelo menos um ou dois navegadores.
- Fazer uma análise e depois verificar se o sistema está executando as configurações mínimas de hardware sem que há perda de performance;

### ***3.2 Teste de Unidade***

- Verificar cada componente do sistema para ver se o seu funcionamento está funcionando de forma correta.

### ***3.3 Teste de Carga***

- Verificar e Avaliar o desempenho do sistema com 1 usuário conectado no mesmo;
- Verificar e Avaliar o desempenho do sistema com 10 usuários conectado no mesmo;
- Verificar e Avaliar o desempenho do sistema com 50 usuários conectado no mesmo;

### ***3.4 Teste de Usabilidade e compatibilidade***

- Fazer o teste se a interface do ambiente se adapta a diferentes resoluções de tela e diferentes tipos de navegadores em diferentes sistemas operacionais;

### ***3.5 Teste de Banco de Dados***

- Analisar se o banco de dados não possui informações inseguros ou incoerentes;

- Testado para verificar integridade do banco de dados;

### **3.6 *Teste de Segurança***

- Verificar se tem a possibilidade de captura de dados de uma transação, ou que seja disponíveis em tela, e ter acesso não permitido direto ao banco de dados.

## **3. Resumo dos Testes Planejados**

Todos os testes mostrados aqui serão planejados para execução, os que não serão executados não precisam ser aplicados.

### **3.1 *Resumo das Inclusões dos Testes***

Principais testes planejados para a interação atual:

- Teste de caixa branca;
- Teste de caixa preta;
- Teste de carga;
- Teste de configuração;
- Teste de segurança;
- Teste de unidade;
- Teste de usabilidade;

### **4.2 *Resumo dos Outros Candidatos a Possível Inclusão***

Em caso de necessidade há possibilidade para utilização de ferramenta para teste automatizado end-to-end.

## **Abordagem dos Testes**

A abordagem de testes será realizada manual e/ou automatizada, e uma profunda análise a ser realizada. Em outras palavras ela define se o que interessa e somente respostas geradas pelo item em teste e o seu comportamento.

**Teste de configuração:**

Verifica a operação do sistema em diferentes configurações de software e hardware para que haja um bom desempenho em todas as funcionalidades de software e hardware, evitando mudanças em questão, sem afetar o desempenho do software a ser desenvolvido. Essa técnica verifica se o alvo de teste funciona corretamente nas configurações exigidas pelo software e hardware. Faz parte desse teste o sistema operacional de rede, os servidores de redes, banco de dados e outros recursos.

**Teste de unidade:**

Tem por objetivo a especificação de explorar a menor unidade do projeto procurando por defeitos ou falhas de lógicas implementadas em cada módulo individualmente, são testados os métodos dos objetos e/ou pequenos trechos de códigos. Isso validando todas as assinaturas de entradas e saídas sendo elas válidas ou inválidas garantindo que não haverá falhas no sistema.

**Teste de usabilidade e compatibilidade:**

Verifica se o sistema possui uma boa usabilidade para todos os públicos, se adaptando a telas de diversas resoluções, rodando em diversos navegadores em diversos sistemas operacionais, sendo compatível assim com o máximo de dispositivos possível.

**Teste de carga:**

Verificar o comportamento do sistema após o mesmo ser submetido a uma certa quantidade de dados, verificar se os dados inseridos de forma contínua e automatizada foram realmente inseridos no banco de dados.

**Teste de segurança:**

O objetivo é garantir a segurança do funcionamento do sistema averiguando o seu comportamento mediante às diversas solicitações de acesso ilegais nos três ambientes, visando a existência de vulnerabilidades testando os mecanismos de proteção incluso no sistema. Perante a esse teste, irá ver a possibilidade de dúvidas sobre prováveis falhas de vulnerabilidades do software a serem eliminadas.

### **Catálogos Iniciais de Idéias de Teste e Outras Fontes de Referência**

Os testes seguirão as estratégias e técnicas de acordo com os demonstrados abaixo.

## **5.2 Técnicas e Tipos de Teste**

### **5.2.1 Teste de Função**

|                          |  |
|--------------------------|--|
| Objetivo da Técnica:     | O principal objetivo do teste tem como apoio os requisitos funcionais do sistema para que possa verificar-se a consistência de dados da interface testando-as todas as entradas e analisar como o sistema está agindo, de forma incorreta serão testadas as entradas para analisar se não haverá nenhuma falha de entrada de dados           |
| Técnica:                 | Teste de caixa preta será usado como técnica para validar todas as entradas da interface certificando o andamento correto do sistema, com essa técnica deverá revelar os problemas de entradas, tais como: funções incorretas; erros de interface; erros de comportamento ou desempenho; erros de iniciação e término.                       |
| Estratégias:             | As técnicas executadas têm como garantir o desempenho do sistema diante dos processos validando o funcionamento correto, no entanto, os dados de entradas são fornecidos, o teste será executado e o resultado obtido é comparado a um resultado aguardado. Todos os componentes testados pode ser um método, uma função, um componente etc. |
| Ferramentas Necessárias: | Teste funcional ou teste de caixa preta e a ferramenta Apache JMeter/Selenium IDE.   |
| Critérios de Êxito:      | Ao cumprir-se esse teste, o testador deve ter como início o guia de casos de teste ou as especificações de casos de uso, neles estarão todos expostos detalhadamente todo o comportamento do sistema, onde deverá buscar entender erros que o usuário poderá cometer ou que fogem da especificação.  |
| Considerações Especiais: | Assegurar que todo o desempenho ou funcionalidade do sistema para que o usuário final se sinta confortável e que o manuseio do sistema será sem falhas de interfaces.  |

### 5.2.2 Teste da Interface do Usuário

|                          |   |
|--------------------------|---|
| Objetivo da Técnica:     | O objetivo é analisar a usabilidade, acessibilidade da interface em que o sistema disponibiliza para o usuário, oferecendo elementos visuais de total facilidade de manuseio e aprendizado, proporcionando melhor manuseio do sistema e dos recursos que ele oferece. |
| Técnica:                 | Serão utilizadas aplicações web voltadas a teste de interface bem como teste de compatibilidade e expansividade.  |
| Estratégias:             | Avaliando a usabilidade do sistema durante todo o processo do desenvolvimento a fim de fazer alterações necessárias para otimizar a usabilidade do sistema antes da conclusão .   |
| Ferramentas Necessárias: | Para a verificação dos teste indicado será utilizado as ferramentas Selenium e caixa preta ou o teste funcional.  |
| Critérios de Êxito:      | Para ter uma maior interação com o sistema tem que haver ambientes aceitáveis pelo usuário final e que proporcione maior usabilidade e acessibilidade do mesmo.   |
| Considerações Especiais: | Depois da conclusão o sistema tem por obrigação passe por todos os testes onde seja validado todo seu desempenho, usabilidade e acessibilidade para que o usuário se sinta confortável.   |

### 5.2.3 Teste de Carga

|                      |  |
|----------------------|--|
| Objetivo da Técnica: | Assegurar que o sistema funcione com certa quantidade de dados sendo inserida de forma simultânea e contínua.  |
| Técnica:             | Será feito os testes de carga e de performance para ver se o sistema pode suportar um alto índice que será utilizado por vários usuários ao mesmo tempo. |
| Estratégias:         | Verificar a capacidade do sistema durante todo o processo de desenvolvimento, podendo assim ter um aumento na escala do sistema antes da conclusão       |

|                          |   |
|--------------------------|---|
| Ferramentas Necessárias: | A realização dos testes será utilizada como ferramenta o Apache JMeter  |
| Crítérios de Êxito:      | 50 usuários conectados ao mesmo tempo no sistema sem que apresente perda de performance.  |
| Considerações Especiais: | Garantir que durante todo o processo tenha um bom desempenho no processamento utilizado por no mínimo 10 usuários e/ou 50 usuários, que não venha aparecer a corrupção de dados ou perda. |

#### 5.2.4 Teste de Stress

|                          |  |
|--------------------------|--|
| Objetivo da Técnica:     | O objetivo é testar quantidades de transações por minutos que estão sendo solicitados através de vários usuários entre 10 ou 20 ao mesmo tempo, com o intuito de ter resultados suportados do software e do hardware verificando seu processo diante as solicitações.                    |
| Técnica:                 | O teste tem a capacidade de mostrar os problemas no sistema de trabalho muito pesadas devido suas transações, analisa o alto índice de atividades simultâneos dos usuários.  |
| Estratégias:             | Constitui-se em testar se o sistema até o seu limite, testando se o sistema que venha exibir recursos de quantidades, frequências anormais, no entanto, teste limites de memória, volume grande de dados de entrada até que chegue ao limite do processamento do software e do hardware. |
| Ferramentas Necessárias: | Será utilizado a ferramenta JMeter para a realização do teste e verificar o desempenho de todo o sistema.  |
| Crítérios de Êxito:      | Será realizado este teste para que o sistema seja de total eficiência e que haja um bom desempenho e na usabilidade.   |
| Considerações Especiais: | A principal importância é que o usuário se sinta satisfeito na utilização com o sistema, sem que não haja erros ou falhas e que tenha um bom desempenho.   |

#### 5.2.5 Teste de Configuração

|                      |  |
|----------------------|--|
| Objetivo da Técnica: | Garantir que o sistema funcione conforme pretendido seguindo o hardware mínimo especificado. |
|----------------------|--|



|                          |   |
|--------------------------|---|
| Técnica:                 | Testar funcionalidades do sistema usando máquina com requisitos mínimos e teste de desempenho do sistema.   |
| Estratégias:             | Abrir / fechar diversos aplicativos do PC, como parte do este ou antes do início do teste e executar transações selecionadas para simular atividades do usuário para dentro e fora de diversos aplicativos de PC. |
| Ferramentas Necessárias: | Navegadores para acesso sistema WEB: Mozilla Firefox e Google Chrome e/ou utilização de Emulador BlueStacks para simular Android.   |
| Critérios de Êxito:      | Sistema deve estar funcionando corretamente utilizando configuração de hardware mínimo.   |
| Considerações Especiais: | Assegurar que sistema funcione 100% utilizando configuração de hardware mínimo.   |

## **Critérios de Entrada e de Saída**

### ***Plano de Teste***

#### ***Critérios de Entrada de Plano de Teste***

O plano de teste inicializará ao término da implementação dos requisitos especificados no documento de requisitos.

#### ***Critérios de Saída de Plano de Teste***

A conclusão dos testes se dará quando todos os requisitos forem testados, e minimizados a quantidade erros (bugs). Em casos de erros no processo, os testes serão suspensos até que os problemas sejam concluídos e assim possam continuar os testes.

#### ***Critérios de Suspensão e de Reinício***

No caso de ocorrência de muitos erros (bugs), haverá a suspensão dos testes, pois demandaria muito esforço desnecessário. O reinício ocorrerá quando a raiz do problema que causou a suspensão tiver sido corrigida.

## *Ciclos de Teste*

### ***Critérios de Entrada de Ciclo de Teste***

Os ciclos de teste seguirão a ordem especificada do plano de testes.

### ***Critérios de Saída de Ciclo de Teste***

Os testes deverão passar pelos ciclos, sendo que todos os defeitos e exceções terão que serem corrigidos.

### ***Término Anormal do Ciclo de Teste***

Ocorrerá se houver falhas no planejamento do plano de teste.

## **Produtos Liberados**

Serão produtos liberados todos os resultados dos testes em planilhas, documentos e/ou imagens que comprovem o teste que serão disponibilizados via Google Drive a cada iteração.

## **Fluxo de Trabalho de Teste**

Todos os resultados dos testes devem ser registrados em um documento que deverá conter um histórico para futuras análises. Será utilizada a ferramenta Google Drive, assim podendo gerenciar as versões desses documentos, para que todos os envolvidos possam acompanhar todos os processos.

## **Necessidades Ambientais**

| Recursos do Sistema              |            |             |
|----------------------------------|------------|-------------|
| Recurso                          | Quantidade | Nome e Tipo |
| Pc's de Desenvolvimento de Teste | 01         | Desktop     |

|                            |    |         |
|----------------------------|----|---------|
| Rede ou Sub-Rede           | 01 | DHCP    |
| Servidor do Banco de Dados | 01 | MongoDB |

### *Hardware Básico do Sistema*

Os conjuntos de tabelas a seguir apresentam os recursos do sistema necessários ao esforço de teste descrito neste *Plano de Teste*.

| <b>Recursos do Sistema</b>      |                   |  |
|---------------------------------|-------------------|--|
| <b>Recurso</b>                  | <b>Quantidade</b> | <b>Nome e Tipo</b>                             |
| Servidor de Banco de Dados      | 01                | MongoDB  |
| Rede ou Sub-rede                | 01                | DHCP   |
| Repositório de Teste            | 01                | Google Drive                                   |
| Rede ou Sub-rede                | 01                | DHCP   |
| PCs de Desenvolvimento de Teste | 01                | AMD 6300 3,50GHZ , HD 1TB, SSD 120GB, 8 GB RAM |

### *Elementos de Software Básicos do Ambiente de Teste*

São necessários os seguintes elementos de software básicos no ambiente de teste deste *Plano de Teste*.

| <b>Nome do Elemento de Software</b> | <b>Versão</b> | <b>Tipo e Outras Observações</b> |
|-------------------------------------|---------------|----------------------------------|
| Windows 10                          | 1909          | Sistema Operacional              |
| Google Chrome                       | 86            | Navegador da Internet            |

### *Ferramentas de Produtividade e de Suporte*

Serão utilizadas as seguintes ferramentas para suportar o processo de teste deste *Plano de Teste*.

| <b>Categoria ou Tipo de Ferramenta</b> | <b>Nome da Marca da Ferramenta</b> | <b>Fornecedor ou Desenvolvida Internamente</b> | <b>Versão</b> |
|--|------------------------------------|--|---------------|
| Gerenciamento de Teste                 | Testlink                           | TestLink Development Team                      | 1.9.20        |
| Ferramenta ASQ para                    | GTmetrix                           | CarbonGO                                       | 0.1.6         |

|                     |  |  |  |
|---------------------|--|--|--|
| teste de desempenho |  |  |  |
|---------------------|--|--|--|

### *Configurações do Ambiente de Teste*

Devem ser fornecidas e suportadas as seguintes Configurações de Ambiente de Teste para este projeto.

| <b>Nome da Configuração</b>   | <b>Descrição</b>                                       | <b>Implementada na Configuração Física</b>                              |
|-------------------------------|--|---|
| Configuração do usuário comum | Terminal de uso com App ou Painel Web com Chrome       | Computador, com máquinas virtuais devidamente implantadas               |
| Mínima configuração suportada | Configuração mínima para garantir um bom funcionamento | HD de 8 gb e mínimo de 512mb de RAM, Google Chrome versão 86 ou próxima |

### **Responsabilidades, Perfil da Equipe e Necessidades de Treinamento**

#### *Pessoas e Papéis*

Esta tabela mostra as suposições referentes ao perfil da equipe do esforço de teste.

| <b>Recursos Humanos</b> |   |  |
|-------------------------|---|--|
| <b>Papel</b>            | <b>Recursos Mínimos Recomendáveis ou Responsáveis</b> | <b>Responsabilidades ou Comentários Específicos</b>  |
| Testador                | Ivan Luís   | <ul style="list-style-type: none"> <li>● adquirir recursos apropriados</li> <li>● avaliar a eficiência do esforço de teste</li> <li>● determinar os resultados dos testes</li> <li>● definir os elementos dos testes</li> <li>● estruturar a implementação dos testes</li> <li>● implementar os testes e os conjuntos de testes</li> <li>● executar os conjuntos de testes</li> <li>● registrar os resultados</li> <li>● analisar as falhas dos testes e possibilitar a recuperação</li> </ul> |

|                     |                                |  |
|---------------------|--------------------------------|--|
|                     |                                | posterior documentar incidentes  |
| Analista de Sistema | Brenda Maria<br>Italo Vinicius | <ul style="list-style-type: none"> <li>● documentar solicitações de mudança</li> <li>● avaliar a qualidade do produto</li> <li>● revisão e/ou implementação caso faça-se necessário</li> </ul>                                     |
| Gerente de Projeto  | Lucas Vinícius                 | <ul style="list-style-type: none"> <li>● verificar as técnicas de teste</li> <li>● identificar ideias de teste</li> <li>● combinar missão</li> <li>● identificar motivadores</li> <li>● defender os interesses do teste</li> </ul> |

## Procedimentos e Processos de Gerenciamento

***Em caso de falha na execução do Plano de Teste, deverão ser documentadas as falhas e/ou modificar partes e fornecer a revisão adequada do documento com a finalidade de continuar com os testes.***

### *Avaliação dos Produtos Liberados deste Plano de Teste*

A avaliação dos produtos liberados será por todos as partes interessadas de visualização e uso, para tanto a avaliação será documentada e caso faça-se necessário outro teste será realizado.

*Relato de Problemas, Seleção de Pessoas para Resolvê-los e Busca de Soluções*

O relato de problemas pode ser solicitado por qualquer partes do projeto REAT e, portanto, o testador será responsável por buscar a solução da maneira mais adequada possível, a depender do tipo de problema.

**R-EAT**

Trabalho de Conclusão de Curso apresentado à coordenação do Centro Universitário Leão Sampaio como requisito para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Aprovado em: \_\_\_ / \_\_\_ / 2020.

**BANCA EXAMINADORA:**

---

Prof. Esp. – Carlos Barreto de Almeida

---

Prof. Me. – Herlon Ribeiro Parente Cortez

---

Prof. Esp. – Renato da Silva Marques

---

Prof. Me. – Wellington Feitoza Gonçalves

JUAZEIRO DO NORTE - CE  
2020