

**Gil Teixeira (88194), João Oliveira (93295), Gabriel Ribeiro(93036), João Figueiredo (93250)**  
Turma P1, em 2021-01-20, v1.0

RELATÓRIO – *CONSTRUCTION*

# Desenho e Implementação

## Conteúdos

<b>Desenho e Implementação</b>	<b>1</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Sumário executivo	1
1.2 Controlo de versões	2
<b>2 Arquitetura do sistema</b>	<b>2</b>
2.1 Objetivos gerais	2
2.2 Requisitos com impacto na arquitetura	3
2.3 Decisões tomadas e fundamentação	3
2.4 Arquitetura proposta	3
2.4.1 Arquitetura lógica da solução	3
2.4.2 Interação entre módulos	4
2.4.3 Integrações com sistemas externos	4
<b>3 Ambiente de desenvolvimento e incrementos</b>	<b>5</b>
3.1 Arquitetura de instalação	5
3.2 Tecnologias de desenvolvimento	5
3.3 Incremento desenvolvido	5
<b>4 Histórias e critérios de aceitação</b>	<b>6</b>
4.1 Caracterização das <i>Personas</i> representativas	6
4.2 Histórias para a 1ª iteração da Construção ( <i>Construction #1</i> )	7
4.3 Automação de testes de aceitação	7
<b>5 Referências e recursos</b>	<b>7</b>

## 1 Introdução

### 1.1 Sumário executivo

Este relatório apresenta os resultados da segunda iteração da Elaboration e da fase de Construction, adaptada do método OpenUP, em que se constrói o produto ao longo de várias iterações.

Os principais requisitos com impacto na arquitetura prendem-se com a versatilidade da aplicação de correr na maioria dos sistemas computacionais existentes e ser intuitivo o uso da mesma.

Nesta interação, consideramos prioritário implementar as sub-apps relativas aos restaurantes e estafetas visto que é essencial ao funcionamento da aplicação.

## 1.2 Controlo de versões

Quando?	Responsável	Alterações significativas
18-20/01	João Oliveira	2.4.2, 2.4.3, 4.1, 4.2, 5
20/01	Gil Teixeira	1.1, 2.1, 2.2, 2.3, 3.2
20/01	Gabriel Ribeiro	3.3, 4.3
22/01	João Figueiredo	2.4.1, 3.1

## 2 Arquitetura do sistema

### 2.1 Objetivos gerais

A implementação do sistema é simplista, modular e de alto-nível, já que foi implementado em React Native. A aplicação está disponível em Android e iOS, React Native compila diretamente para código nativo de cada uma das plataformas, e também na [Internet](#). O sistema não necessita de condições de desempenho altas nem no toca o hardware nem na conexão à internet.

e.g.:

- Os clientes e restaurantes devem poder fazer o seguimento das entregas online, praticamente em tempo real, usando uma visualização do mapa da cidade. O seguimento dos estafetas deve recorrer a sensores GPS incorporados no *smartphone*.
- Os clientes registados devem receber notificações (*push notifications*) a anunciar novas oportunidades. Para isso, é importante que haja uma adesão generalizada ao uso das aplicações móveis, na plataforma Android e iOS.

### 2.2 Requisitos com impacto na arquitetura

Requisitos	Descrição
RD-1	Todas as transacções MB demoram menos de 1 minuto
RInt-1	Usar fontes e cores que facilitem a legibilidade da informação.
RS-1	A informação relativa a clientes (pessoal, de histórico de encomendas, e de pagamentos efetuados) deve ser guardada de forma cifrada.
RU-1	O portal da loja deve-se ajustar para ter uma apresentação adequada ao ecrã, designadamente para <i>smartphones</i> , <i>tablets</i> e computadores.
RSeg-1	Usar protocolos de autenticação seguros

### 2.3 Decisões tomadas e fundamentação

Tendo em conta os objetivos para a arquitetura, e os requisitos levantados na Análise, foram tomadas as seguintes decisões:

- Frontend implementado com a biblioteca React-native. Biblioteca criada e mantida pelo Facebook com uma grande comunidade. Possibilita a construção do frontend para aplicações móveis e web ao mesmo tempo.
- Backend feito usando firebase, plataforma da Google que permite fazer a gestão de bases de dados e também dos pagamentos.

## 2.4 Arquitetura proposta

### 2.4.1 Arquitetura lógica da solução

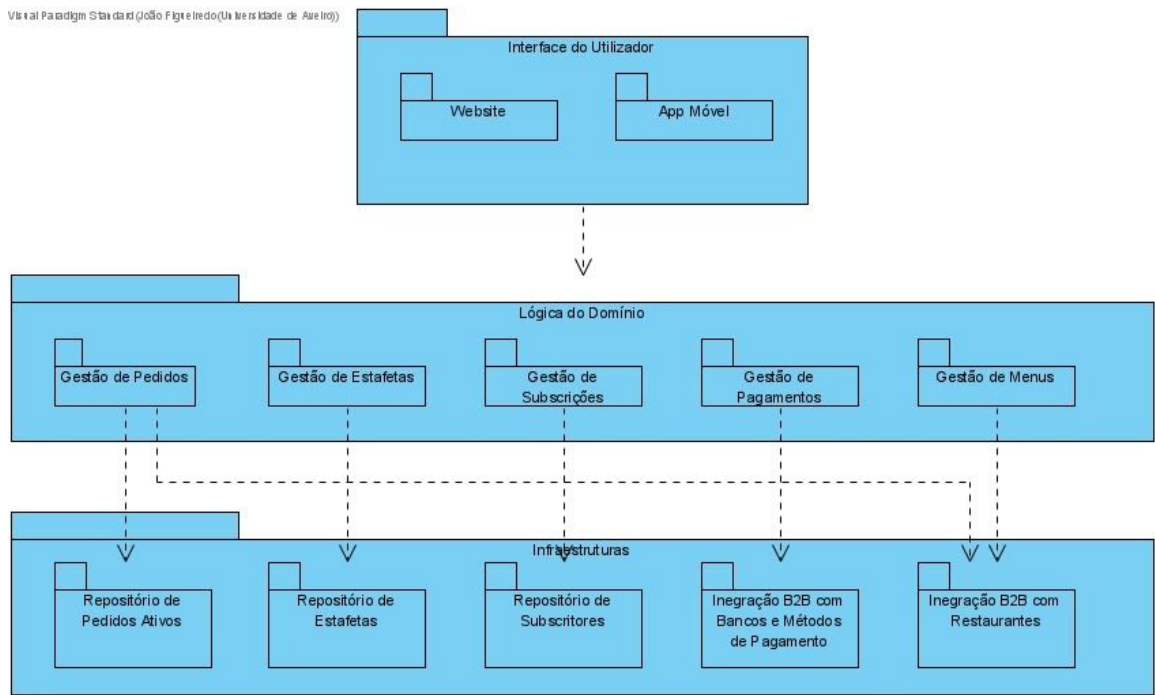


Diagrama 1: Arquitetura Lógica do Sistema

### 2.4.2 Interação entre módulos

Cliente:

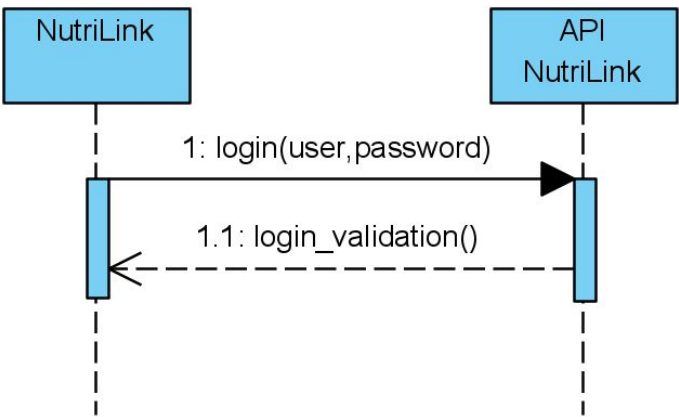


Diagrama 2: Acesso ao sistema

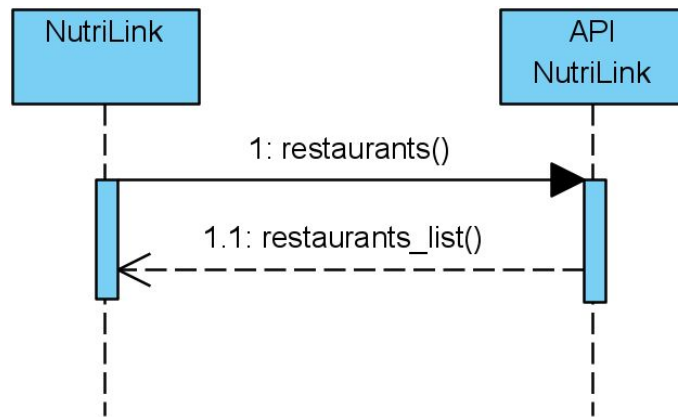


Diagrama 3: Visualização dos restaurantes

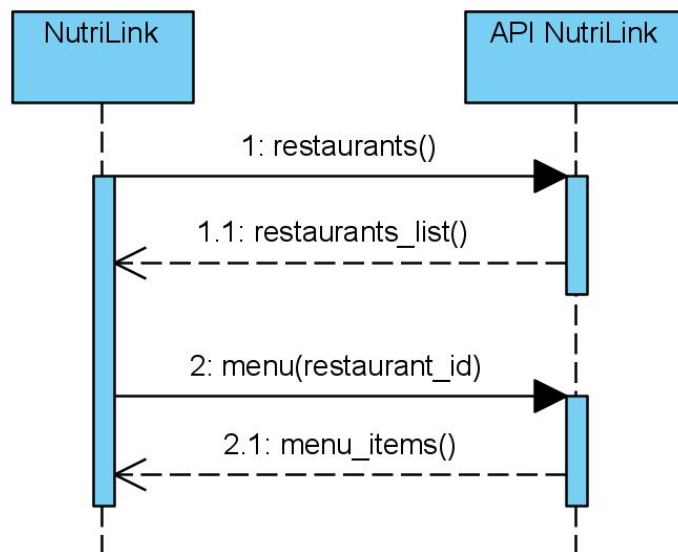


Diagrama 4: Visualização dos pratos disponibilizados por um determinado restaurante

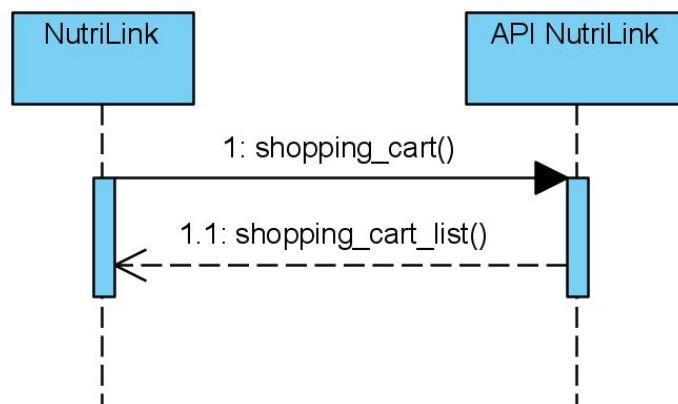


Diagrama 5: Visualização do carrinho

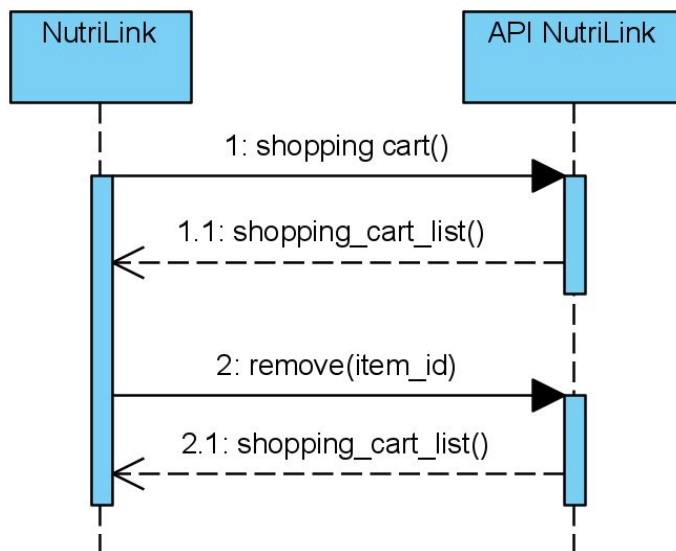


Diagrama 6: Remoção de prato do carrinho

**Dono/representante de um restaurante:**

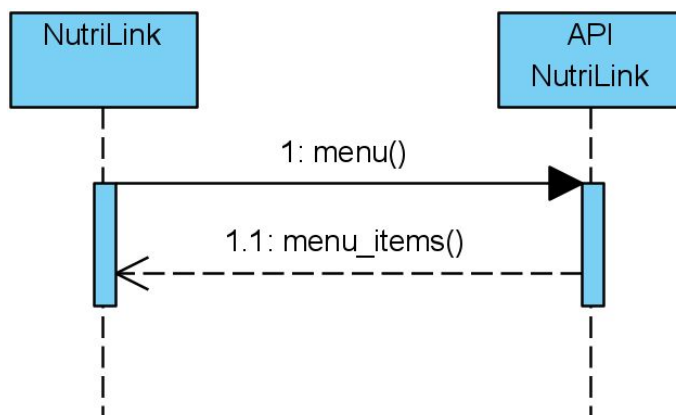


Diagrama 7: Visualização dos pratos disponibilizados pelo restaurante

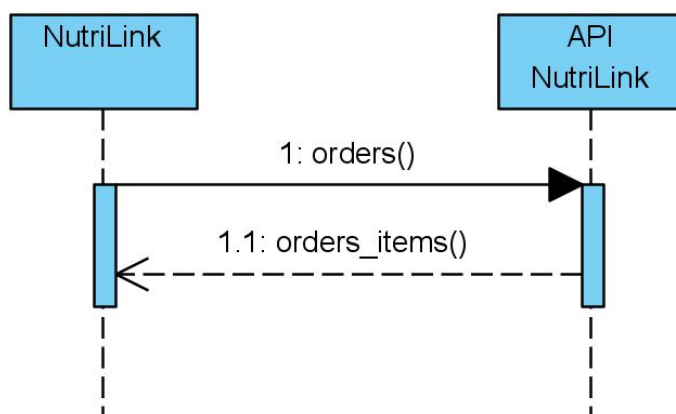


Diagrama 8: Visualização dos pedidos direcionados ao restaurante

**Estafeta:**

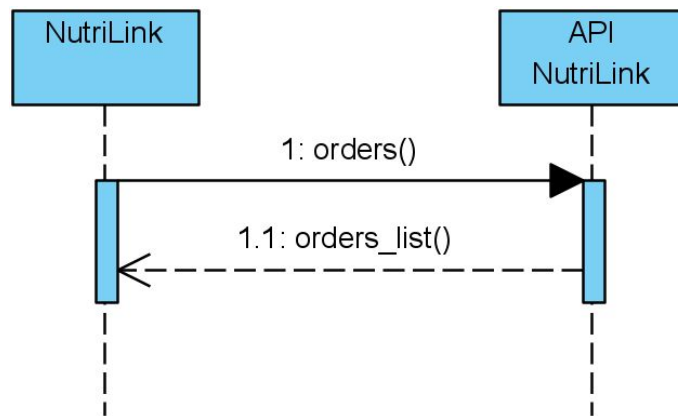


Diagrama 9: Visualização pedidos realizados pelos clientes

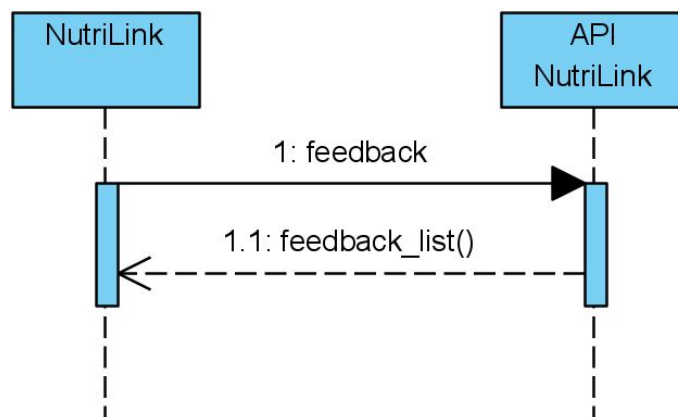


Diagrama 10: Visualização do feedback da entregas deixado pelos clientes

### 2.4.3 Integrações com sistemas externos

Relativamente à integração com sistemas externos, não necessitamos de o implementar na aplicação móvel, visto que, neste caso, não se justificou.

## 3 Ambiente de desenvolvimento e incrementos

### 3.1 Arquitetura de instalação

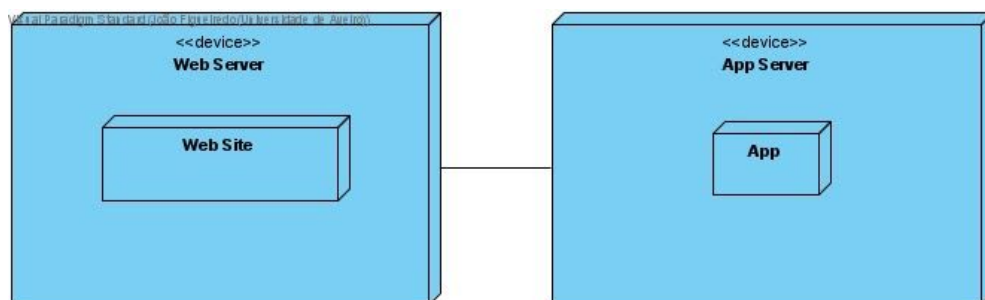


Diagrama 11: Arquitetura de Instalação

## 3.2 Tecnologias de desenvolvimento

A aplicação está a ser desenvolvida em Javascript com recurso à biblioteca React Native. A ligação à database vai ser feita com recurso a fetches à api do firebase, tecnologia escolhida também para dar handle dos pagamentos. A aplicação fará uso de sensores de GPS, sendo que apenas o cliente estará dispensado do uso deste sensor devendo indicar a morada de entrega à priori. O projeto encontra-se disponível no [Github](#).

## 3.3 Incremento desenvolvido

No nosso incremento implementamos as interfaces de utilização de cada possível utilizador, isto é, a interface dirigida ao estafeta, a interface dirigida ao cliente e, por fim, a interface dirigida ao restaurante/representante do restaurante.

Começámos por representar o resultado desejado num protótipo utilizando o site [marvelapp.com](#) que nos permitiu ter uma ideia do que queríamos para as interfaces na app.

Link do protótipo : <https://marvelapp.com/prototype/55394ed/screen/75382713>

Depois transpusemos esta ideia numa app móvel utilizável com suporte em Android que pode ser encontrada em <https://micro-site-ams.herokuapp.com/> clicando em “Try Live” debaixo da secção projetos.

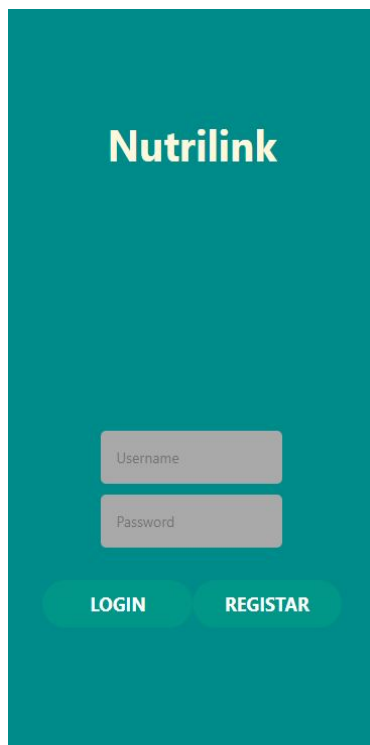
Aqui também pode ser encontrado o código desenvolvido (<https://github.com/bearkillerPT/appAMS>) e o download da app para ser utilizada em Android clicando no botão “Download”.

Depois de estar na app na secção username consoante a interface que queremos ver colocamos como username o utilizador, i.e., se quisesse entrar como cliente colocaria o username “cliente” e uma password aleatória.

Depois de estar logado ainda podemos dar logout clicando no botão correspondente.

No nosso incremento não é possível para as interfaces comunicarem entre si, isto é, se entrar como cliente e fizer um pedido a um restaurante se depois entrar como restaurante nada mudará.

Seguem os diagramas implementados e alguns screenshots ( para ter conhecimento total convém abrir a app e explorar a mesma).



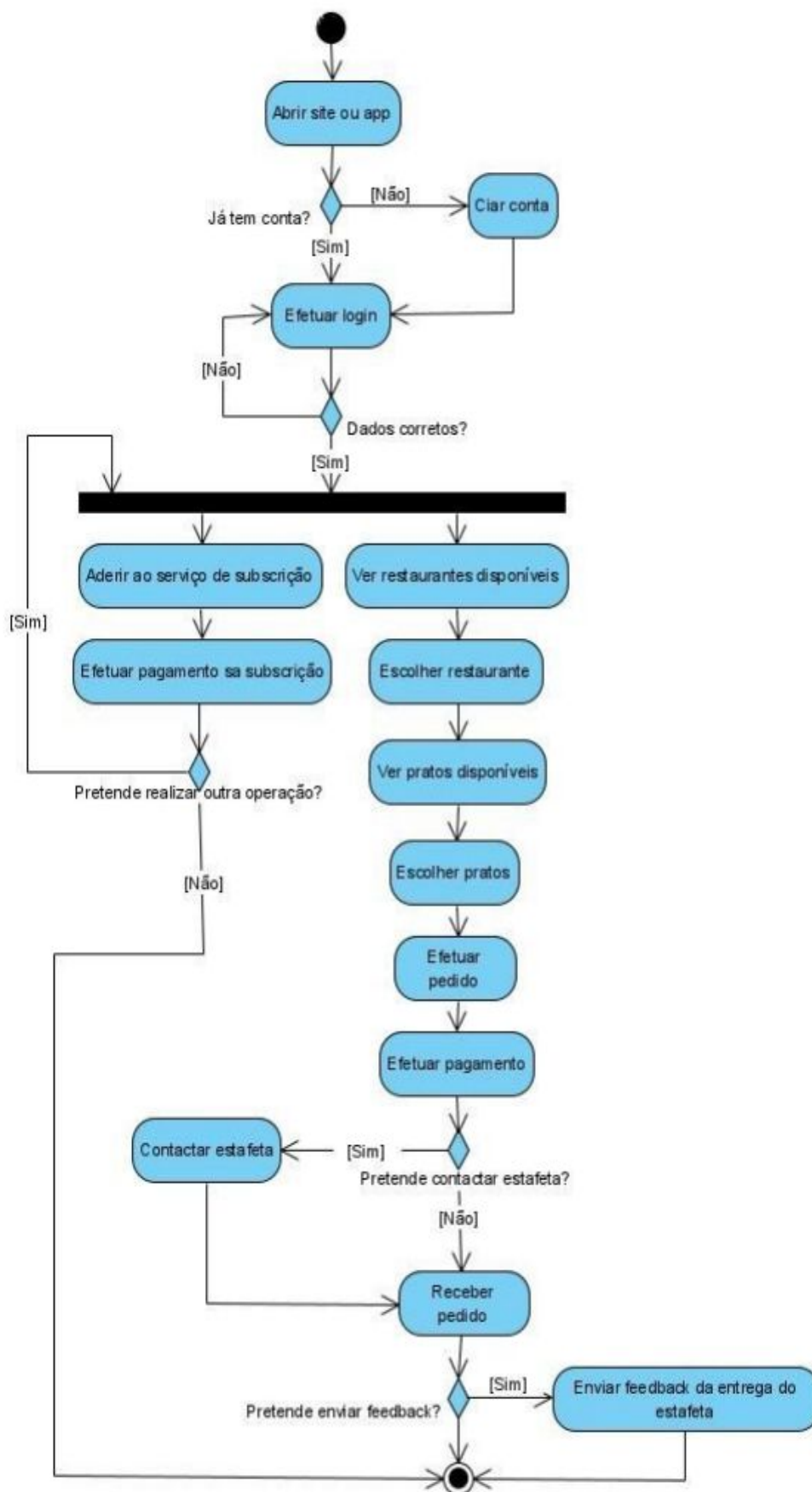
Screenshot 1: Interface inicial



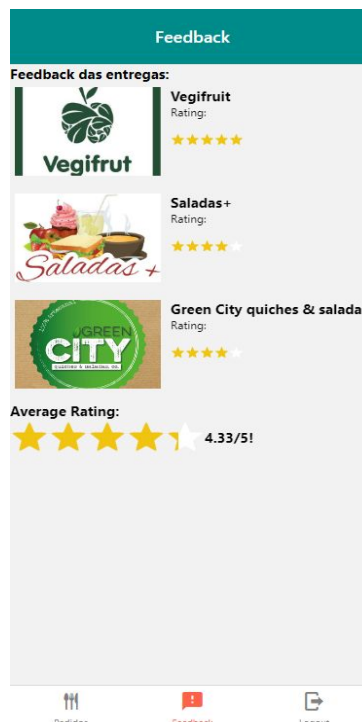
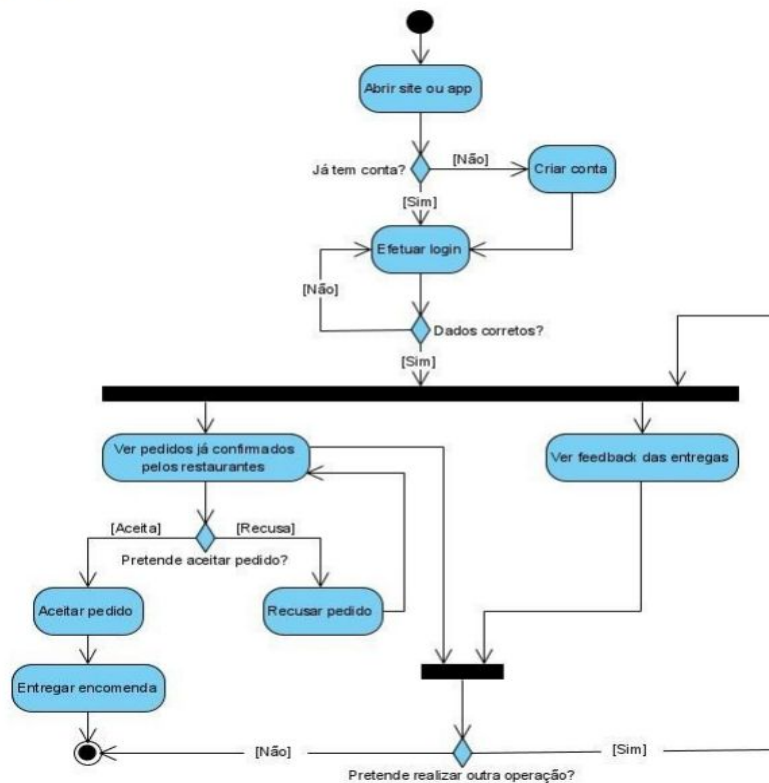
Screenshot 2 : Exemplo de carrinho de cliente



Na interface do cliente temos o seguinte diagrama implementado:

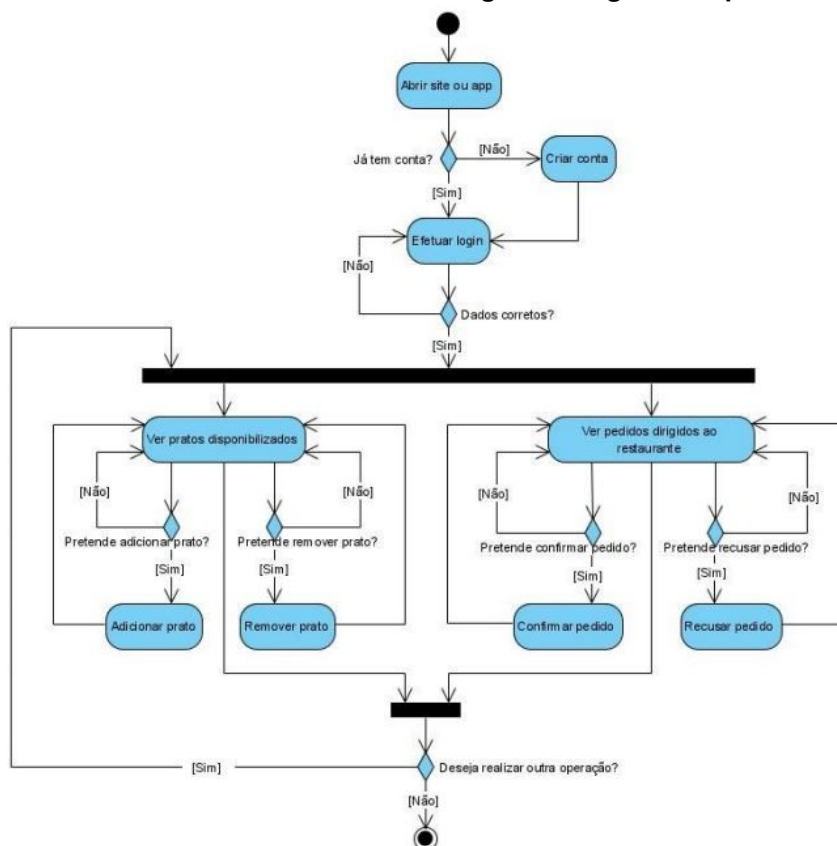


Na interface dos estafetas temos o seguinte diagrama implementado:



Screenshot 3: Feedback na interface do estafeta

Na interface do restaurante temos o seguinte diagrama implementado:



Screenshot 4: Pratos disponibilizados na interface do estafeta

## 4 Histórias e critérios de aceitação

### 4.1 Caraterização das *Personas* representativas

#### *Persona 1: Artur*

O Artur tem 37 anos, vive em Vila Nova de Gaia, é barbeiro, tem 3 filhos e um cão, o que faz com que não tenha muito tempo disponível, ele gostava de ter uma alimentação mais saudável pois não se sente bem com a sua saúde e preocupa-se com a futura saúde dos seus filhos.

Com isto, optou por utilizar a aplicação NutriLink o que lhe facilitou nos seus cuidados alimentares bem como o dos seus filhos, melhorando assim a sua saúde e consequentemente o seu bem estar.

#### *Persona 2: Rui*

O Rui tem 20 anos, vive com o pai em Matosinhos, é estudante universitário, gosta de ir ao ginásio e tem consultas com um nutricionista que lhe aconselha certos pratos saudáveis para, por vezes, incluir na sua alimentação, pratos esses que nem ele nem o seu pai sabem cozinhar.

Então, para facilitar a sua tarefa na alimentação utiliza a aplicação NutriLink para ver os melhores restaurantes bem como os respetivos pratos saudáveis disponíveis.

#### *Persona 3: Alexandra*

A Alexandra tem 45 anos, vive em Aveiro, é enfermeira e trabalha por turnos. Devido a esta situação pandémica tem pouco tempo para dedicar aos seus cozinhados saudáveis que tanto gosta.

Então usa a aplicação NutriLink para escolher os pratos que deseja comer, o que lhe poupa tempo e dá acesso a cozinhados saudáveis com a qualidade que deseja.

#### *Persona 4: Manuel*

O Manuel tem uma paixão pela cozinha e tem problemas graves de saúde o que lhe condiciona a alimentação, então decidiu abrir um restaurante de comida saudável, mas devido ao confinamento optaram por serviços take-away. Neste sentido, para o ajudar aconselharam a aplicação NutriLink, por um lado, pelo conceito de comida saudável e, por outro, pela facilidade de gestão do restaurante na aplicação.

#### *Persona 5: Joana*

A Joana tem uma mota e gosta muito de passear nela, com as despesas que a vida universitária lhe trouxe pensou arranjar um emprego então juntou o útil ao agradável e optou por ser estafeta da aplicação NutriLink.

## 4.2 Histórias para a 1ª iteração da Construção (*Construction #1*)

### **O Artur aceder ao sistema**

Sendo o Artur, um cliente

Quer aceder à aplicação

De modo a ter acesso às funcionalidades de um cliente

#### **Cenário 1: Adere com sucesso**

Dado que estou página de login

E insiro as credenciais

E as credenciais são válidas

Quando seleciono o botão de login

Então é apresentada a página inicial do cliente

#### **Cenário 2: Adere sem sucesso**

Dado que estou página de login

E insiro as credenciais

E as credenciais são inválidas

Quando seleciono o botão de login

Então é apresentada a página de login

### **O Rui vê os restaurantes disponíveis**

Sendo o Rui, um cliente

Quer ver os restaurantes disponíveis na aplicação

De modo a ter acesso aos restaurantes que confeccionam a comida que deseja

#### **Cenário 1: Vê com sucesso**

Dado que estou página inicial do cliente

Quando seleciono a aba dos restaurantes

Então é apresentada a página com os restaurantes presentes na aplicação

### **O Rui vê os pratos disponibilizados por um restaurante**

Sendo o Rui, um cliente

Quer ver os pratos disponibilizados por um determinado restaurante

De modo a ter acesso aos pratos saudáveis que deseja

#### **Cenário 1: Vê com sucesso**

Dado que estou página inicial do cliente

E vou para a aba dos restaurantes

Quando seleciono um restaurante dos disponibilizados

Então é apresentada a página com os pratos disponibilizados pelo respetivo restaurante

#### **Cenário 2: Vê com sucesso**

Dado que estou página inicial do cliente

Quando seleciono um restaurante dos favoritos que aparecem

Então é apresentada a página com os pratos disponibilizados pelo respetivo restaurante

### **A Alexandra remove um prato do seu carrinho**

Sendo a Alexandra, uma cliente  
Quer remover um prato do seu carrinho  
De modo a eliminar um prato indesejado do seu pedido

**Cenário 1: Prato removido com sucesso**

Dado que estou na página inicial do cliente  
    E escolho o restaurante que desejo  
    E escolho os pratos  
    E vou à aba do carrinho  
Quando seleciono o prato indesejado  
Então é apresentada a página do carrinho sem o prato indesejado

**A Alexandra encomenda um prato**

Sendo a Alexandra, uma cliente  
Quer encomendar um prato  
De modo a ter acesso ao prato que deseja

**Cenário 1: Pedido efetuado com sucesso**

Dado que estou na página inicial do cliente  
    E escolho o restaurante que desejo  
    E escolho o prato desejado  
    E vou à aba do carrinho  
    E efetuo o pagamento  
Quando seleciono o botão de pagamento final  
Então é apresentada a página de espera  
    E o restaurante aceita  
    E um estafeta aceita

**Cenário 2: Pedido efetuado sem sucesso**

Dado que estou na página inicial do cliente  
    E escolho o restaurante que desejo  
    E escolho o prato desejado  
    E vou à aba do carrinho  
    E efetuo o pagamento  
Quando seleciono o botão de pagamento final  
Então é apresentada a página de espera  
    E o restaurante recusa

**O Manuel vê os pratos disponibilizados pelo seu restaurante**

Sendo o Manuel, um dono/representante de um restaurante  
Quer ver os pratos disponibilizados pelo seu restaurante  
De modo a ver a ementa do seu restaurante disponível na aplicação

**Cenário 1: Prato visualizados com sucesso**

Dado que estou página inicial do restaurante  
Então é apresentada a página inicial do restaurante com os pratos disponibilizados pelo mesmo

**O Manuel vê os pedidos direcionados ao seu restaurante**

Sendo o Manuel, um dono/representante de um restaurante  
Quer ver os pedidos direcionados ao seu restaurante  
De modo a confeccionar esses mesmos pedidos se possível

**Cenário 1:** Pedidos visualizados com sucesso

Dado que estou na página inicial do restaurante

Quando seleciono a aba dos pedidos

Então é apresentada a página dos pratos direcionados ao restaurante

**A Joana vê os pedidos realizados pelos clientes**

Sendo a Joana, uma estafeta

Quer ver os pedidos realizado pelos clientes

De modo a decidir qual aceitar

**Cenário 1:** Pedidos visualizados com sucesso

Dado que estou na página inicial do estafeta

Então é apresentada a página inicial do estafeta com os pedidos realizados pelos clientes

**A Joana visualiza o feedback das suas entregas deixado pelos clientes**

Sendo a Joana, uma estafeta

Quer ver o feedback deixado pelos clientes acerca das suas entregas

De modo a avaliar-se a si própria com objetivo a melhorar

**Cenário 1:** Visualização efetuada com sucesso

Dado que estou página inicial do estafeta

Quando seleciono a aba do feedback

Então é apresentada a página do feedback deixado pelos clientes acerca dos pedidos entregues

## 4.3 Automação de testes de aceitação

Visto não termos feitos testes automáticos, pode-se referir ao capítulo 3.3 para verificar o funcionamento da aplicação.

## 5 Referências e recursos

Documentos:

- <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
- <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-package-diagram/>
- <https://surfedigital.io/blog/o-que-e-persona/>