Anexo E: Matriz de CRUZ

	Cliente	Admin
Selecionar Produto para adicionar ao carrinho	CRU	
Adicionar/Remover items do carrinho	RU	
Mapa com preços e distâncias	RU	
Seleção da loja	RU	
Introduzir Produtos Novos		CR
Alterar Produtos		RU

Project Charter for Software Development

1. Descrição do Projeto

A ideia surgiu da necessidade de encontrar um método para poupar dinheiro cada vez que vamos às compras.
Existem plataformas que fazem o mesmo mas para outros tipos de serviços, como por exemplo, comparar preços de quartos no mesmo hotel – Trivago ou procurar e comparar o preço de componentes eletrónicos da informática e preços de eletrodomésticos – KuantoKusta. Com a criação do nosso website pretendemos oferecer aos clientes uma forma rápida e diferente de comparar os preços de vários supermercados, e de seguida, ajudá-los a chegar à loja escolhida com os produtos prontos para levantar.

2. Critério de sucesso

O nosso projeto irá ter colaboradores que irão estar sempre disponiveis e espalhados por todos os supermercados. Quando estes receberem uma notificação de compra eles irão de seguida comprar os produtos que poderão ou levar até ao cliente ou esperar que o cliente levante a encomenda. O cliente irá pagar uma taxa dependo do serviço que escolheu sendo a taxa superior a da entrega até casa devido à deslocação do colaborador até ao cliente.

3. Project scope

O nosso projeto tem como objetivo principal oferecer o melhor preço para as suas compras de comida, sejam estas online ou em loja física. Ao criar um carrinho de compras o nosso projeto irá filtrar todos os supermercados e por fim mostrar a loja com a melhor oferta. Este projeto está inserido num contexto do género portal "Trivago", onde o utilizador irá ter à sua escolha a esmagadora maioria das redes de supermercados existentes no país, onde poderá fazer o seu carrinho de compras, e no final, nós damos-lhe os respetivos valores da sua compra nas diferentes cadeias de supermercados perto de si.

4. Constrangimentos

Dimension	**Constraint**	**Driver**
Schedule	Project Duration	School Project
Scope	Full fledged supermarket filtering system	School Project
Quality	W3C standars	School Project
Resources	Html, Css, javascript, node.js	School Project

5. SWOT analysis

Strengths	**Weaknesses**	**Threats**	**Opportunities**
Utility	Not Professional	Existance of similar websites	Have reoccuring customers
User-Friendly	Low Reputation	Erro nas Api's	

Responsive			
Design			

6. Requisitos

Os requisitos que irão ter de ser completados para garantir que o projeto fique funcional são:

-A criação da página principal com uma barra de navegação com as categorias principais seguido de todos os produtos vendidos nas subcategorias.

-Iremos também criar uma página de Login e Register para que o utilizador possa criar uma conta ou entrar numa conta já existente.

-Será necessário haver um carrinho para ir guardando os produtos desejados.

-Utilizaremos o Api do google mapa para que o utilizador possa escolher a localização da loja desejada

-Será necessário implementar um checkout onde o cliente irá pagar pelos produtos escolhidos

-Será necessário também uma base de dados com a informação dos preços para que o cliente possa saber quanto irá pagar por cada produto.

7. Business risks

Risk	**Probability**	**Impact**
Google Maps APi Failure	Low	Medium-low
Database failure	Low	Severe
Software presents security risks	Low	severe

8. Milestones

Milestones	Data	Responsibility
Criação da página principal	27/09/2019	Sebastian
Implementação do Nav	06/10/2019	Sebastian
Implementação do Nav com sub categorias	13/10/2019	Sebastian
Implementar o Register	31/11/2019	Catarina
Implementar o Login	02/11/2019	Catarina
Carrinho	05/11/2019	Sebastian
Implementação dos produtos	10/11/2019	Sebastian
Implementação do Mapa	17/11/2019	Catarina
Implementação do Checkout	23/11/2019	Catarina
Base de Dados Funcional	30/11/2019	Sebastian

Store swap

Este pedido vai buscar o nome e a localização das lojas

- **URL**

  '/api/produto/store'

- **Method:**

  GET

- **URL Params**

None

- **Data Params**

  None

- **Success Response:**

  Content: {storeNome: Pingo Doce, storeLatitude: (longitude), storeLatitude: (latitude)}

- **Sample Call:**

```
$.ajax({
        url: '/api/produto/store',
        method: 'get',
        success: function (result, status) {
            store = result;
            str = ''
            card = document.getElementById("stores")
            for (i in store) {
                str += '<li onclick="swap(\'' + store[i].nomeSM + '\',' + store[i].Latitude + ',' + store[i].Longitude + ')">' + store[i].nomeSM + '</li>'

            }

            card.innerHTML = str + card.innerHTML

        },
        error: function () {
            console.log('Error');
        }
    })
```

Store swap

Este pedido envia ao servidor os produtos do carrinho de compras com a data, total e id.

- **URL**

  '/api/produto/compra'

- **Method:**

  POST

- **URL Params**

None

- **Data Params**

  Id,

  Produto,

  Preço,

  date

- **Success Response:**

- **Sample Call:**

```
$.ajax({
        url: "/api/produto/compra",
        method: "post",
        data: {
            id: id,
            produto: produtos,
            preco: total,
            date: String(today),

        },
        success: function (res, status) {
            console.log('Success')
        },
        error: function () {
            console.log("Error on post")
        }
    });
```

Store swap

Este pedido vai buscar o produto individual

- **URL**

  '/api/produto'

- **Method:**

  GET

- **URL Params**

  None

- **Data Params**

  None

- **Success Response:**

  Content: {produtoImage: (Image URL), produtoNome: Basmati, produtoAvgPrice: 2,99, produtoDescrição: é bom e faz bem}

- **Sample Call:**

```
$.ajax({
        url: '/api/produto',
        method: 'get',
        success: function (result, status) {
            var str = '';
            produto = result;
            main = document.getElementById("produto")
            for (i in produto) {
                if (produto[i].nome == item) {
                    str = '<div class = "card1"><img id ="productImage" src=' + produto[i].imagem + 'style ="width:"100" height:"100"">' +'<h1 id = "nome" data-value = "' + produto[i].nome + '" class ="shop-item-title">' + produto[i].nome + '</h1><p class="shop-item-price" data-value = "' + produto[i].AvgPrice + '" id ="price">Avg Price: ' + produto[i].AvgPrice + '€</p>' +'<p>' + produto[i].descricao + '</p><p><button type="button" class="btn btn-primary shop-item-button" id="button3">Add to Cart</button></p></div>'
                } }
            main.innerHTML = str
            ready();}

```

Store swap

Este pedido vai buscar os produtos da categoria selecionada

- **URL**

  '/api/produto'

- **Method:**

  GET

- **URL Params**

  None

- **Data Params**

  None

- **Success Response:**

  Content: {produtoImage: (Image URL), produtoNome: Basmati, produtoCategoria: arroz}

- **Sample Call:**

```
$.ajax({
        url: '/api/produto',
        method: 'get',
        success: function (result, status) {
            $(".sub-ctg").click(function (e) {
                e.preventDefault();
                var id = this.id;
                teste = document.getElementById(id).getAttribute("data-value");
                produto = result;
                str = ''
                card = document.getElementById("card")

                for (i in produto) {
                    if (produto[i].categoria == teste) {
                        $('#abc').remove();
                        $('#yeetus').remove();
                        console.log(produto[i].categoria + '==' + teste)
                        str += '<div class="card"><img src ="' + produto[i].imagem
 + '" id ="abc" style="width:100%">' + '<p id = "yeetus" onclick = "loadProduto(\'
' + produto[i].nome + '\')"><a>' + produto[i].nome + '</a></p></div>'

                    }
                }

                card.innerHTML = str + card.innerHTML

            });

        },
        error: function () {
            console.log('Error');
        }
    })
}
```