



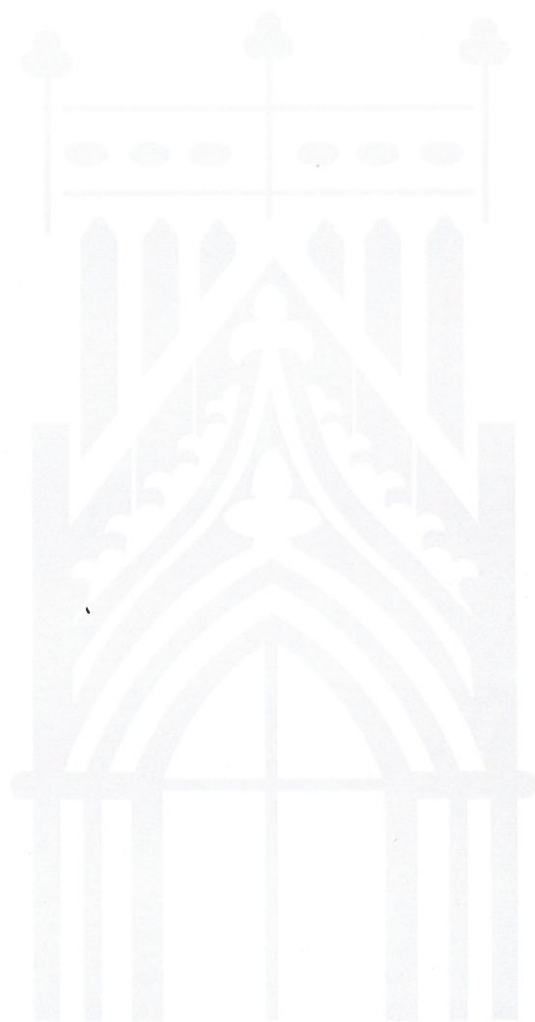
IPG Politécnico
|da|Guarda
Polytechnic
of Guarda

RELATÓRIO DE PROJETO

Licenciatura em Engenharia Informática

Roberto Machado Adelino

janeiro | 2021





Escola Superior de Tecnologia e Gestão

Instituto Politécnico da Guarda

RELATÓRIO DE PROJETO DE INFORMÁTICA

ZERO WASTE

ROBERTO MACHADO ADELINO

RELATÓRIO PARA A OBTENÇÃO DO GRAU DE LICENCIADO
EM ENGENHARIA INFORMÁTICA

Janeiro de 2021

Ficha de identificação

Aluno:

Nome: Roberto Machado Adelino

Nº: 1012539

Licenciatura: Engenharia Informática

Email: roberto_adelino_98@hotmail.com

Estabelecimento de Ensino:

Escola Superior de Tecnologia e Gestão

Instituto Politécnico da Guarda

Av. Dr. Francisco Sá Carneiro 50, 6300-559 Guarda

Tel: 271 220 100 | Fax: 271 222 690

Orientador do Projeto

Nome: Maria Clara Santos Pinto Silveira

Grau académico: Doutoramento em Engenharia Eletrotécnica e de Computadores

Email: mclara@ipg.pt

Resumo

Este relatório descreve o projeto realizado no âmbito da unidade curricular Projeto de Informática, da Licenciatura em Engenharia Informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico da Guarda.

O projeto desenvolvido consiste na criação de uma aplicação *Salesforce*, *ZeroWaste*, para a gestão de uma ONG que presta auxílio a famílias carenciadas e instituições através da distribuição de cabazes. Estes cabazes são constituídos pelos alimentos que sobram em supermercados/restaurantes, ou de doações de terceiros.

Tanto as família como as instituições terão de respeitar regras pré-definidas para terem direito a essa atribuição de alimentos e apenas podem receber doações uma vez por mês.

Estes alimentos serão recebidos e entregues através de voluntários que também se podem inscrever na plataforma. A plataforma também permite a venda de planos de sócio e o recebimento de doações monetárias que serão utilizadas para ajudar a doar alimentos a mais famílias.

Desta forma este documento apresenta todo desenvolvimento da aplicação em si como é o caso de validações, automatismos, programação e tecnologias utilizadas. Para o desenvolvimento do projeto foi utilizada a metodologia de desenvolvimento ágil SCRUM.

A aplicação foi desenvolvida utilizando várias tecnologias como é o caso de *Salesforce*, *Apex*, *CSS*, *HTML*, *Visualforce*, etc.

Palavras-chave: *Salesforce*, *Visualforce Page*, Desperdício Alimentar, Objetivos de Desenvolvimento Sustentável;

Abstract

This report describes the project developed for the curricular unit Project of Informatics, of the Degree in Computer Engineering of the School of Technology and Management of the Polytechnic Institute of Guarda.

The project consists in a *Salesforce* application to manage ZeroWaste, an NGO that aids needy families and institutions through the distribution of food baskets, these are made up with the food left over in supermarkets / restaurants, or donations from third parties.

Both families and institutions will have to respect pre-defined rules to be entitled to this food distribution and can only receive donations once a month.

Food products will be received and delivered through volunteers who can also register on the platform.

The platform also allows the sale of partner plans and money donations that will be used to help donating food to more families.

In this way, this document presents the entire development of the application itself, as is the case with validations, automations, programming and technologies used.

For the development of the project the SCRUM agile development methodology was used.

The web platform was developed using various technologies such as Salesforce, Apex, CSS, HTML, Visualforce, etc.

Keywords

Salesforce, Visual Force Page, Food waste, Sustainable Development Goals.

Índice

Ficha de identificação.....	i
Resumo.....	ii
Abstract.....	iii
Índice de figuras Índice de ilustrações.....	viii
Índice de tabelas.....	xi
1 Introdução.....	1
1.1 Motivação.....	2
1.2 Objetivos.....	3
1.3 Estrutura do documento.....	4
2 Estado da Arte.....	5
2.1 Contexto.....	5
2.2 Plataformas existentes.....	7
2.2.1 <i>GoodAfter</i>	7
2.2.2 <i>Too Good to Go</i>	7
2.2.3 FrutaFeia.....	8
2.2.4 <i>ReFood</i>	8
2.3 Análise Comparativa de Funcionalidades.....	9
3 Metodologia de desenvolvimento de <i>software</i>	10
3.1 Metodologia SCRUM.....	10
3.2 Aprendizagem de <i>Salesforce</i>	11
4 Análise de requisitos.....	13
4.1 Diagrama de casos de uso.....	13
4.2 Descrição de casos de uso.....	15
4.3 Diagrama sequência.....	23
4.4 Tabela de classes/objetos.....	26
4.5 Diagrama de classes.....	27

4.6	Dicionário de dados	29
5	Tecnologias	30
5.1	Tecnologias utilizadas.....	30
5.1.1	<i>Salesforce</i>	30
5.1.2	<i>Visualforce</i>	30
5.1.3	<i>Apex</i>	31
5.1.4	<i>HTML</i>	31
5.1.5	<i>CSS</i>	31
5.1.6	<i>Visual Studio Code</i>	31
5.1.7	<i>Developer Console</i>	31
6	Implementação	32
6.1	<i>Record Types e Page Layouts</i>	32
6.2	Regras de Validação	32
6.3	<i>Process Builders</i>	32
6.4	<i>Flows</i>	33
6.5	<i>Triggers</i>	33
6.6	Classes Apex (Controladores)	33
6.7	<i>VisualForce Pages</i>	34
6.8	<i>Email-alerts/email templates</i>	34
6.9	<i>Reports e Dashboards</i>	34
6.10	<i>AppExchange</i>	34
6.11	Recursos estáticos e tema.....	35
6.12	<i>Home page da organização</i>	36
6.13	Objetos <i>custom</i>	37
6.13.1	<i>Money donations</i>	37
6.13.2	<i>Food Donations</i>	40
6.14	Objetos <i>Standard</i>	43

Zerowaste

6.14.1	<i>Accounts</i>	43
6.14.2	<i>Contacts</i>	44
6.14.3	<i>Leads</i>	48
6.14.4	<i>Products</i>	50
6.14.5	<i>Food requests and Sales (Opportunity)</i>	51
6.14.6	<i>Food Requests to deliver (Order)</i>	63
6.14.7	<i>Campaigns</i>	65
6.14.8	<i>Tasks</i>	67
6.15	Implementação do <i>website</i> com recurso a <i>Visualforce pages</i>	67
6.15.1	Página inicial do website	68
6.15.2	Página de doação de alimentos e de doação de dinheiro.....	70
6.15.3	Página de venda de planos de sócio.....	71
6.15.4	Página de recruta de voluntários.....	71
6.15.5	Página de contacto	72
6.16	Página de agradecimento	72
7	Verificação e validação	73
8	Conclusões	78
9	Bibliografia.....	79
	Anexos.....	82
A 1.	Schema builder da aplicação	83
A 2.	Dicionário de dados	84
A 2.1.	Objeto Contacts	84
A 2.2.	Operações do Objeto Contacts	85
A 2.3.	Objeto Accounts.....	85
A 2.4.	Operações do Objeto Account.....	86
A 2.5.	Objeto Requests and sales	86
A 2.6.	Operações do Objeto Requests and sales	87

Zerowaste

A 2.7.	Objeto leads.....	88
A 2.8.	Operações do objeto Leads	89
A 2.9.	Objeto Requests to deliver	90
A 2.10.	Operações do objeto Requests to deliver.....	91
A 2.11.	Objeto Money Donations	91
A 2.12.	Operações do objeto Money Donations	92
A 2.13.	Objeto Food Donations.....	93
A 2.1.	Operações do Objeto Food Donations	94
A 2.2.	Objeto Campaigns	95
A 2.3.	Operações do objeto Campaigns	96
A 2.4.	Objeto Tasks.....	96
A 2.5.	Operações do Objeto Tasks.....	97
A 2.6.	Objeto Products	98
A 2.7.	Operações do objeto Products	98
A 3.	Código	99
A 3.1.	Código da classe que gera uma fatura.....	99
A 3.2.	Trigger De renovação de Pedido de famílias e instituições	104
A 3.3.	Trigger de renovação de plano de sócio.....	105
A 3.4.	Página inicial do website.....	106
A 3.5.	Página de doação de alimentos.....	120
A 3.6.	Controlador da página de doação de alimentos.....	123
A 3.7.	Página de doação de dinheiro.....	124
A 3.8.	Controlador da página de doação de dinheiro.....	127
A 3.9.	Controlador das páginas de venda de planos de sócio	128
A 3.10.	Controlador da página de recruta de voluntários.....	131
A 3.11.	Página de contacto	132

Índice de figuras | Índice de ilustrações

Figura 1- Perdas alimentares anuais em Portugal na cadeia de abastecimento [6]	2
Figura 2- Objetivos de Desenvolvimento Sustentável	6
Figura 3-Interface do site GoodAfter	7
Figura 4- Interface da aplicação Too Good To Go.....	7
Figura 5- Interface do site FrutaFeia	8
Figura 6- Interface do site ReFood.....	8
Figura 7-Metodologia SCRUM [25].	10
Figura 8-Plano da academia	11
Figura 9-Trailhead	12
Figura 10-Diagrama de Casos de Uso	14
Figura 11-Diagrama de sequência “Doar alimentos” através do website	23
Figura 12- Diagrama de sequência “Entregar alimentos	24
Figura 13- Diagrama de sequência “Confirmar entrega de alimentos”	24
Figura 14- Diagrama de sequência “Distribuir cabazes de alimentos”	25
Figura 15-Diagrama de classes simplificado.....	28
Figura 16-Logotipo da organização.....	35
Figura 17-Tema criado	35
Figura 18-Página inicial no salesforce	36
Figura 19-Filtro no campo dos doadores	37
Figura 20-Regra de validação no campo Amount.....	37
Figura 21-Regra de validação no formato do IBAN	37
Figura 22-Regra de validação no campo da data.....	38
Figura 23-Process builder de notificação de nova doação	38
Figura 24-Process builder que envia um email e respetivo template criado	39
Figura 25-Tarefa criada	39
Figura 26-Report das doações	40
Figura 27-Página de um registo de doação de alimentos.	40
Figura 28- Regra de validação no campo dos cabazes.	41
Figura 29-Filtro no campo dos voluntários	41
Figura 30-Process builder que envia uma notificação.....	41
Figura 31-Process builder que atualiza a quantidade total de cabazes disponíveis.....	42
Figura 32-Gráfico das doações de cabazes.....	42

Figura 33-Dashboard dos cabazes disponíveis para doação.....	43
Figura 34-Record types do objeto account	43
Figura 35-Página de um registo do objeto contacto	44
Figura 36-Record types criados.....	45
Figura 37-Exemplo de um page layout dos contactos	45
Figura 38-Regra de validação da idade	46
Figura 39-Regra no campo do número de pessoas do agregado familiar.....	46
Figura 40- Regra no campo do número de pessoas que a instituição abrange	46
Figura 41-Regra de validação do comprovativo nas famílias	46
Figura 42-Regra de validação do campo do telefone	47
Figura 43-Validação do email	47
Figura 44-Process builder que envia um email ao novo contacto	47
Figura 45-Report do tipo de contactos total	48
Figura 46-Página de um registo do objeto Lead.....	48
Figura 47-Mapeamento das leads	49
Figura 48-Conversão de uma lead	49
Figura 49-Process builder que envia uma notificação para nova lead	50
Figura 50-Todos os produtos da plataforma.....	50
Figura 51-Regra de validação na quantidade	51
Figura 52-Process builder que envia uma notificação.....	51
Figura 53-Página de um registo de pedido de alimentos.....	52
Figura 54-Record types do objeto Food requests and sales (Opportunity)	52
Figura 55-Filtro no campo das Accounts	53
Figura 56-Filtro no campo Partner plan	53
Figura 57-Filtro no campo dos contactos	53
Figura 58-Filtro nos campos dos cabazes.....	53
Figura 59-Regra no campo dos cabazes	54
Figura 60-Flow que gera uma registo no objeto Requests to deliver	55
Figura 61-Process builder semelhante ao flow anterior	56
Figura 62-Process builder que atualiza o preço do plano de sócio.....	57
Figura 63-Process builder que atribui cabazes de acordo com as regras e requisitos	59
Figura 64-Process builder que envia notificação de erro	60
Figura 65-Processo que chama o flow da figura 60	60
Figura 66-Report que mostra o tipo de Plano de sócio vendidos	61

Figura 67-Botão para gerar PDF	61
Figura 68-Exemplo de fatura gerada	62
Figura 69-Novo pedido gerado automaticamente	63
Figura 70-Renovação de sócio automática	63
Figura 71-Página de uma registo	63
Figura 72-Filtro no campo dos Voluntários	64
Figura 73-Regra de validação no estado da Order	64
Figura 74-Process builder que cria uma task e envia emails	64
Figura 75-Process builder que envia uma notificação	65
Figura 76-Página das campanhas e formulário de criação de nova campanha	66
Figura 77-Página das tarefas	67
Figura 78-Página inicial do site parte 1	68
Figura 79-Página inicial do site parte 2	69
Figura 80-Página de doação de alimento	70
Figura 81-Página de doação de dinheiro	70
Figura 82-Páginas de compra de plano de sócio	71
Figura 83-Página de registo de novo voluntário	71
Figura 84-Página de contacto	72
Figura 85-Página de agradecimento	72
Figura 86-Testes de validação	73
Figura 87-Teste de regras	73
Figura 88-Testes de validação do formato de dados	74
Figura 89-Testes de validação do campo quantidade	74
Figura 90-Validação do formato do IBAN	74
Figura 91-Teste de atualização do estado de uma entrega	75
Figura 92- Tarefa gerada automaticamente	75
Figura 93-Email enviado ao novo contacto registado na plataforma	75
Figura 94-Email enviado ao utilizador que fez uma doação	76
Figura 95-Notificação recebida pelo administrador	76
Figura 96-Notificação de doação de alimentos	76
Figura 97-Validação de dados no website	77
Figura 98-Validação de dados no website	77
Figura 99-Modelo ER gerado pela plataforma	83

Índice de tabelas

Tabela 1- Comparação de funcionalidades.....	9
Tabela 2- Tabela de atores e respetivos casos de uso.....	13
Tabela 3- Descrição do caso de uso “Distribuir cabazes”	16
Tabela 4- Descrição do caso de uso “Comprar plano de sócio”	17
Tabela 5- Descrição do caso de uso “Confirmar entrega”	18
Tabela 6- Descrição do caso de uso “Entregar alimentos”	19
Tabela 7- Descrição do caso de uso “Doar dinheiro”	20
Tabela 8-Descrição do caso de uso “Doar alimentos” através do website.....	21
Tabela 9- Descrição do caso de uso “Doar alimentos”	22
Tabela 10-Objetos/tabelas da aplicação	26
Tabela 11-Regras de atribuição de cabazes a famílias e instituições	58
Tabela 12-Dicionário de dados do objeto Contacts.....	84
Tabela 13 -Conjunto de operações no objeto Contacts	85
Tabela 14- Dicionário de dados do objeto Accounts.....	85
Tabela 15 -Conjunto de operações no objeto Account.....	86
Tabela 16 -Dicionário de dados do objeto Requests and sales (opportunity)	86
Tabela 17- Conjunto de operações no objeto Requests and Sales (opportunity)	87
Tabela 18 Dicionário de dados do objeto Leads.....	88
Tabela 19 -Conjunto de operações no objeto Leads.....	89
Tabela 20- Dicionário de dados do objeto Requests to deliver (orders)	90
Tabela 21-Conjunto de operações no objeto Requests to deliver (orders)	91
Tabela 22- Dicionário de dados do objeto Money Donations	91
Tabela 23 -Conjunto de operações no objeto Money Donations.....	92
Tabela 24 Dicionário de dados do objeto Food Donations	93
Tabela 25 -Conjunto de operações no objeto Food Donations.....	94
Tabela 26- Dicionário de dados do objeto Caompaings.....	95
Tabela 27- Conjunto de operações no objeto Campaigns	96
Tabela 28- Dicionário de dados do objeto Tasks	96
Tabela 29 -Conjunto de operações no objeto Tasks	97
Tabela 30- Dicionário de dados do objeto Products.....	98
Tabela 31- Conjunto de operações no objeto Products	98

Lista de siglas e acrónimos

SIGLA	SIGNIFICADO
API	<i>Application Programming Interface</i>
CLI	<i>Command Line Input</i>
CRM	<i>Customer relationship management</i>
CSS	<i>Cascading Style Sheets</i>
FAO	<i>Food and Agriculture Organization of the United Nations</i>
HTML	<i>Hypertext Markup Language</i>
IDE	<i>Integrated development environment</i>
ODS	<i>Objetivos de Desenvolvimento Sustentável</i>
ONG	Organização não governamental
ONU	<i>Organização das Nações Unidas</i>
UML	<i>Unified Modeling Language</i>

1 Introdução

Este documento descreve o projeto desenvolvido pelo aluno Roberto Machado Adelino, no âmbito da unidade curricular de Projeto de Informática, da Licenciatura em Engenharia Informática no Instituto Politécnico da Guarda. O projeto desenvolvido consiste na criação de uma plataforma *Salesforce* para a gestão de uma ONG que presta auxílio a famílias carenciadas e instituições através da distribuição dos alimentos que sobram em supermercados/restaurantes, ou de doações externas. Tanto as família como as instituições terão de respeitar regras pré-definidas para terem direito a essa atribuição de alimentos, por exemplo apenas podem receber doações uma vez por mês.

Os alimentos serão recebidos e entregues através de voluntários que também se podem inscrever na plataforma. A plataforma também permite doações através de planos de sócio, bem como doações monetárias que serão utilizadas para ajudar a doar alimentos a mais famílias.

É considerado desperdício alimentar as perdas que acontecem em qualquer uma das etapas da cadeia alimentar, da produção ao consumo, e que implicam que os alimentos não cumpram o propósito para o qual foram produzidos, isto é, não sejam consumidos [1].

Segundo a *Food and Agriculture Organization of the United Nations* (FAO), o desperdício global de alimentos por ano é de aproximadamente 1,3 mil milhões de toneladas. Este desperdício gera:

- Um custo de 990 mil milhões de euros [2].
- O aumento das emissões de gases de efeito de estufa
- A diminuição das reservas de água potável
- Um impacto na biodiversidade do planeta.

De acordo com este estudo, esse desperdício corresponde a um terço da produção mundial de alimentos e seria suficiente para alimentar duas vezes [3] as 820 milhões [4] de pessoas que são atingidas pela fome em todo o mundo [5].

Estima-se que em Portugal sejam desperdiçados anualmente mais de 1 milhão de toneladas de alimentos [6] que dariam para alimentar as 360 mil pessoas que passam fome no nosso país [7], uma grande parte desse desperdício provém da distribuição (Figura 1).



Figura 1- Perdas alimentares anuais em Portugal na cadeia de abastecimento [6]

1.1 Motivação

A motivação para desenvolver este projeto surge do impacto que o assunto do desperdício alimentar tem no planeta, uma vez que o elevado número de alimentos desperdiçados pode servir para combater a fome e também combater as alterações climáticas. Desta forma os recursos naturais utilizados para a produção desses alimentos não serão utilizados em vão.

Neste contexto existiu a necessidade de criar uma plataforma que facilitasse aos utilizadores da ONG o registo da informação dos produtos alimentares disponíveis para posteriormente ajudar na distribuição para as pessoas carenciadas.

1.2 Objetivos

Este projeto tem como objetivo principal o desenvolvimento de uma plataforma *Salesforce* para auxiliar na distribuição de alimentos dos supermercados e restaurantes para famílias carentes e instituições. A plataforma deverá cumprir os seguintes objetivos:

- Plataforma online;
- *User Friendly*, ou seja, a plataforma é de boa usabilidade;
- Produz gráficos que ajudam na análise e compreensão de dados;
- Registrar informação das instituições e famílias;
- Registrar informações de doadores, sócios e voluntários;
- Registrar cabazes e alimentos disponíveis;
- Permitir doações através de planos de sócio, por exemplo doações semanais;
- Aceitar doações (dinheiro e alimentos);
- Atribuir cabazes às famílias e instituições;
- Efetuar pedido de alimentos;
- Atribuir pedidos a voluntários, para estes fazerem a entrega;
- Registrar se os cabazes chegaram ao destino;
- Desenvolver um *website* que permita receber doações, e recrutar voluntários.

No caso do *website* é necessário que este seja adaptável a dispositivos móveis, que seja um *website* responsivo, de modo a facilitar a sua utilização.

1.3 Estrutura do documento

O presente relatório é constituído por oito capítulos, que descrevem, todas as fases do desenvolvimento de *software*. É feita uma breve descrição de cada capítulo.

Capítulo 1:

O primeiro capítulo apresenta uma breve introdução do projeto, da motivação da sua realização e dos objetivos.

Capítulo 2:

Neste capítulo é apresentada a realização de um estudo e a análise das funcionalidades de aplicações/plataformas já existentes e é feita uma comparação das mesmas em relação à aplicação a desenvolver.

Capítulo 3:

Neste capítulo é apresentada a metodologia SCRUM que foi utilizada no desenvolvimento.

Capítulo 4:

No capítulo 4 é descrita a análise de requisitos

Capítulo 5:

No quinto capítulo são descritas as tecnologias utilizadas no desenvolvimento do projeto.

Capítulo 6:

No capítulo 6 é descrita a fase de implementação.

Capítulo 7:

Neste capítulo é descrito o conjunto de testes que foram feitos ao longo do desenvolvimento.

Capítulo 8:

No último capítulo são apresentadas conclusões do trabalho efetuado.

2 Estado da Arte

Após a definição dos objetivos do projeto, foi feito um estudo e análise das funcionalidades de aplicações já existentes no mercado. Esta pesquisa permitiu ter uma visão do que é necessário neste tipo de plataformas, assim como estudar funcionalidades novas que são relevantes para a solução proposta.

Esta pesquisa foi baseada nos seguintes parâmetros:

1. Funcionalidades existentes;
2. Semelhança com os objetivos apresentados.

2.1 Contexto

O projeto desenvolvido consiste na criação de uma aplicação *Salesforce* para a gestão de uma ONG que presta auxílio a famílias carentes através da distribuição dos alimentos que sobram em supermercados que já não podem ser comercializados, mas ainda se encontram próprios para consumo, os excedentes de refeições de restaurantes, ou de doações externas, evitando assim algum desperdício alimentar. A plataforma também permite a venda de planos de sócio e o recebimento de doações monetárias que serão utilizadas para ajudar a doar alimentos a mais famílias.

A plataforma está assim relacionada com os Objetivos de Desenvolvimento Sustentável (ODS). Os ODS [8] são um conjunto de 17 metas globais (Figura 2), estabelecidas pelos países da ONU (Organização das Nações Unidas). Os ODS são parte da Agenda 2030 [9] para o Desenvolvimento Sustentável.



Figura 2- Objetivos de Desenvolvimento Sustentável

A aplicação vai assim integrar os ODS:

1-Eradicar a pobreza: ao reduzirmos o desperdício e a doar alimentos estamos a contribuir para a erradicação da pobreza e da fome.

2-Eradicar a fome e promover a agricultura sustentável: porque uma boa gestão dos alimentos vai permitir acabar com a fome, alcançar a segurança alimentar, melhorar a nutrição e promover uma agricultura sustentável.

3-Saúde e bem-estar: com uma melhor alimentação é assegurada uma vida saudável e um bem-estar para todos.

12-Consumo e produção responsáveis: pois todos os alimentos cumprirão o seu propósito

13-Ação contra as alterações climáticas: através do aproveitamento de todos os alimentos, todos os recursos naturais utilizados na produção não serão desperdiçados.

2.2 Plataformas existentes

Na pesquisa foram encontradas várias plataformas focadas em diminuir o desperdício alimentar, das quais as seguintes quatro têm a maior presença no mercado.

2.2.1 *GoodAfter*

A GoodAfter [10] é um supermercado online dedicado à venda de produtos que se encontram perto do fim do prazo de consumo, ou mesmo com esse prazo ultrapassado. Na Figura 3 é apresentada a *interface* do site.

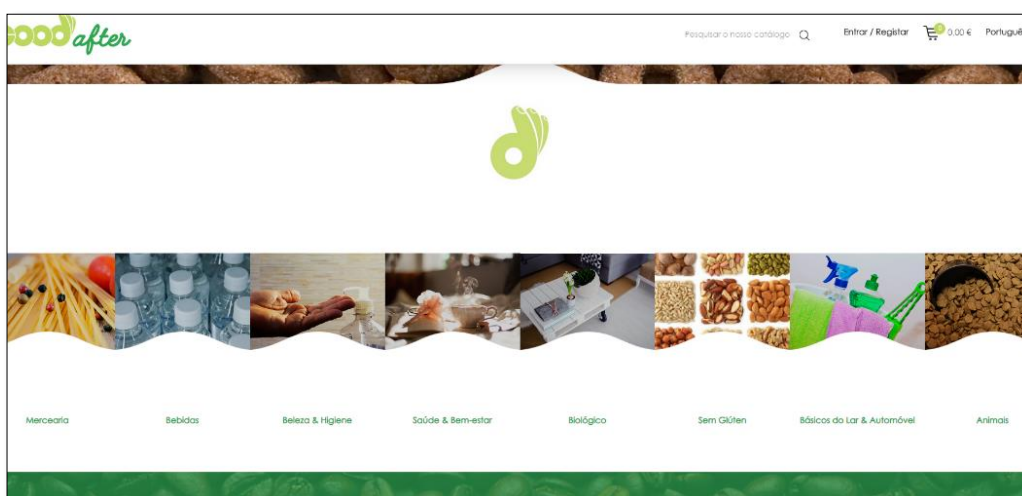


Figura 3-Interface do site GoodAfter

2.2.2 *Too Good To Go*

A Too Good To Go [11] combate o desperdício alimentar através de uma app gratuita [12] onde é possível encomendar excedentes alimentares de restaurantes, hotéis e supermercados evitando que estes acabem no lixo. Tudo a preços reduzidos. Atualmente só está disponível na zona litoral do país. A Figura 4 representa a *interface* da aplicação.

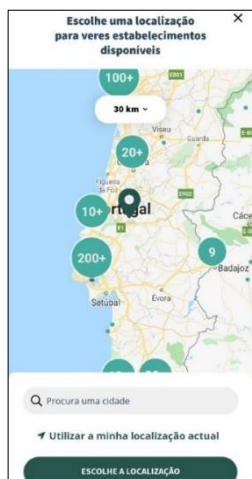


Figura 4- Interface da aplicação Too Good To Go

2.2.3 FrutaFeia

A cooperativa Fruta Feia [13] resulta de uma ideia para aproveitar cerca de um terço da fruta e vegetais que os supermercados desperdiçam por considerarem que não têm a aparência perfeita ou o calibre necessário que os consumidores procuram. Na Figura 5 é apresentada a *interface* do *site*.

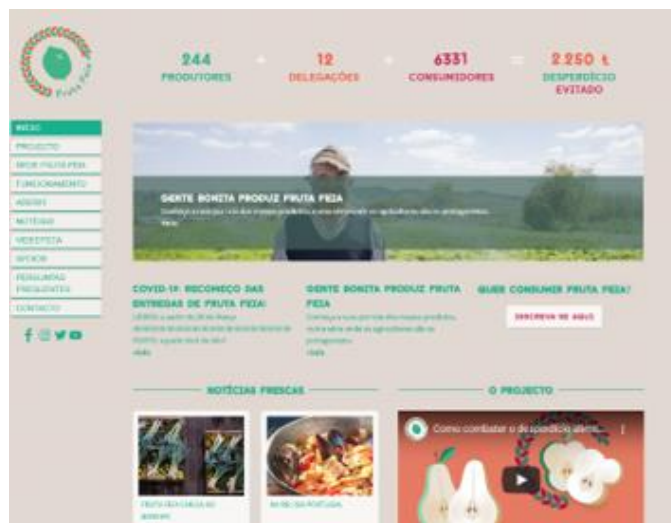


Figura 5- Interface do site FrutaFeia

2.2.4 ReFood

A ReFood [14] é uma organização sem fins lucrativos, conduzida por cidadãos voluntários, que recolhem a comida que sobra nos restaurantes, supermercados, padarias, cafés, pastelarias, hotéis, etc. e distribuem por aqueles que necessitam. Desta forma, pretendem reduzir o desperdício alimentar, atenuando a fome e diminuindo a quantidade de resíduos que, de outra forma, acabariam nos aterros sanitários, agravando o problema da gestão dos resíduos nas cidades. Na Figura 6 é apresentada a *interface* do *website*.



Figura 6- Interface do site ReFood.

2.3 Análise Comparativa de Funcionalidades

A Tabela 1 mostra uma comparação das funcionalidades das aplicações apresentadas, relevantes à execução deste projeto.

Tabela 1- Comparação de funcionalidades

Funcionalidades	Good After	Too Good To Go	Fruta feia	Refood	<u>ZeroWaste</u>
Registo de Supermercados/ Restaurantes	✗	✓	✗	✓	✓
Registo de instituições	✗	✗	✗	✓	✓
Registo de voluntários	✗	✗	✗	✓	✓
Registo de pessoas/famílias	✓	✓	✓	✓	✓
Registo de alimentos disponíveis.	✓	✓	✗	✗	✓
Fazer pedido de alimentos gratuitamente	✗	✗	✗	✗	✓
Venda de alimentos	✓	✓	✓	✗	✗
Entrega alimentos	✓	✗	✗	✓	✓
Fazer Doações de alimentos	✗	✗	✗	✓	✓
Receber doações monetárias/ de alimentos	✗	✗	✗	✗	✓
Venda de planos de sócio	✗	✗	✗	✗	✓
Registrar estado das entregas de cabazes	✗	✗	✗	✗	✓

A *GoodAfter* [10], a *Too Good To Go* [11] e a *Fruta Feia* [13] focam-se na venda de bens alimentares. A *ReFood* [14] é a plataforma que mais se assemelha aos objetivos pretendidos da *ZeroWaste*, no entanto as suas atividades não são geridas numa plataforma *online* o que torna a sua gestão mais complicada. Todas as aplicações apresentadas têm semelhanças com os objetivos da *ZeroWaste*, no entanto, não têm funcionalidades essenciais como é o caso da possibilidade de famílias ou instituições receberem doações de alimentos ou fazerem pedidos para receberem essas doações.

3 Metodologia de desenvolvimento de *software*

No desenvolvimento de qualquer projeto de *software* é necessário seguir certos modelos e regras, que permitem agilizar o processo. Para que esse processo seja mais claro existem diferentes tipos de metodologias de desenvolvimento de *software*, tais como, a metodologia estruturada, metodologia orientada a objetos e metodologias de desenvolvimento ágil.

Uma metodologia de *software* é um procedimento que tem um objetivo importante na estruturação, planificação e controlo do processo de desenvolvimento de *software*. É a utilização de uma metodologia que vai permitir que o desenvolvimento seja realizado de forma focalizada de acordo com as exigências do cliente.

A metodologia Ágil traz mais flexibilidade durante o processo de planeamento e desenvolvimento de *software* relativamente às outras metodologias nas quais os seus processos rígidos e pouco iterativos dificultam os meios que necessitam de constante atualização tanto na parte da introdução de novas tecnologias como por exigências do cliente

Optou-se assim por usar a metodologia ágil, *Scrum*, por se adequar a projetos complexos, com requisitos que mudam com alguma frequência.

3.1 Metodologia SCRUM

Scrum é das *frameworks* mais utilizadas da metodologia ágil uma vez que é simples e facilidade de adaptar e utilizar. Scrum é caracterizada pelas etapas de desenvolvimento, chamadas *Sprints*. O funcionamento deste modelo pode ser observado na Figura 7.

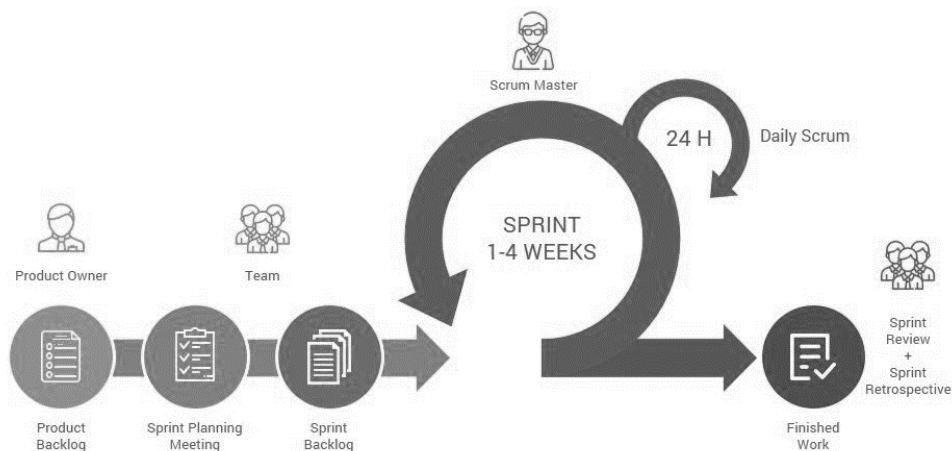


Figura 7-Metodologia SCRUM [25].

Este modelo é baseado vários processos executados até ao fim do projeto, sendo estes [15]:

- *Product Backlog* - lista prioritária dos requisitos a serem implementados no projeto, estabelecidos pelo *Product Owner* neste caso é o aluno;
- *Sprint Backlog* - lista que provém do *Product Backlog* e contém os requisitos a serem desenvolvidos na próxima iteração;
- *Sprints* – realização de diversas tarefas para completar um requisito definido no *Sprint Backlog*;
- *Daily Scrum* - reuniões diárias onde se apura o trabalho realizado desde a última *Daily Scrum* e se tenta prever que trabalho será feito até à próxima reunião, auxiliando a perceber a probabilidade de a equipa alcançar o objetivo do *Sprint*.
- *Sprint Review* - reunião onde está presente *Product Owner*(aluno), *Scrum Master* e *Team* (professor orientador), onde é mostrado o trabalho alcançado durante o *Sprint*.
- *Retrospective*- ocorre ao final de um *Sprint* e serve para identificar o que funcionou bem, o que pode ser melhorado e que ações serão tomadas para melhorar.

3.2 Aprendizagem de Salesforce

O aluno participou na academia *Salesforce* que se realizou de 6 de outubro a 13 de novembro, em que os colaboradores da Loba [16] passaram os seus conhecimentos e experiências sobre a tecnologia *Salesforce*. O programa formativo está ilustrado na Figura 8.



ESTE É O PROGRAMA FORMATIVO COMPOSTO POR 6 MÓDULOS QUE PENSAMOS PARA TI

Início da formação: 1 de outubro das 17h às 19h

- 1ª semana (6 a 9 de outubro): **Salesforce Overview**
- 2ª semana (12 a 16 de outubro): **The Process Automation**
- 3ª semana (19 a 23 de outubro): **Data and Reports**
- 4ª semana (26 a 30 de outubro): **Administration Setup**
- 5ª semana (2 a 6 novembro): **Code with Apex Class**
- 6ª semana (9 a 13 novembro): **Lightning Design System**

Figura 8-Plano da academia

Para além disso, utilizou-se também a plataforma de aprendizagem *Trailhead* [17] (Figura 9). Esta plataforma é a maneira mais eficiente de aprender *Salesforce* que pode ser comparada a um jogo em que se sobe de nível à medida que se aprende mais e se fazem mais módulos. A plataforma vai atribuindo pontos e emblemas e tem como objetivo o aumento das capacidades e habilidades dentro do *Salesforce*.

Dentro da plataforma existem “trilhas” que são caminhos de aprendizagem guiados que traçam o curso de quem aprende. Além das *trilhas* existem os emblemas que são tutoriais breves. Estes emblemas ensinam tópicos isolados que uma pessoa queira aprender de imediato.

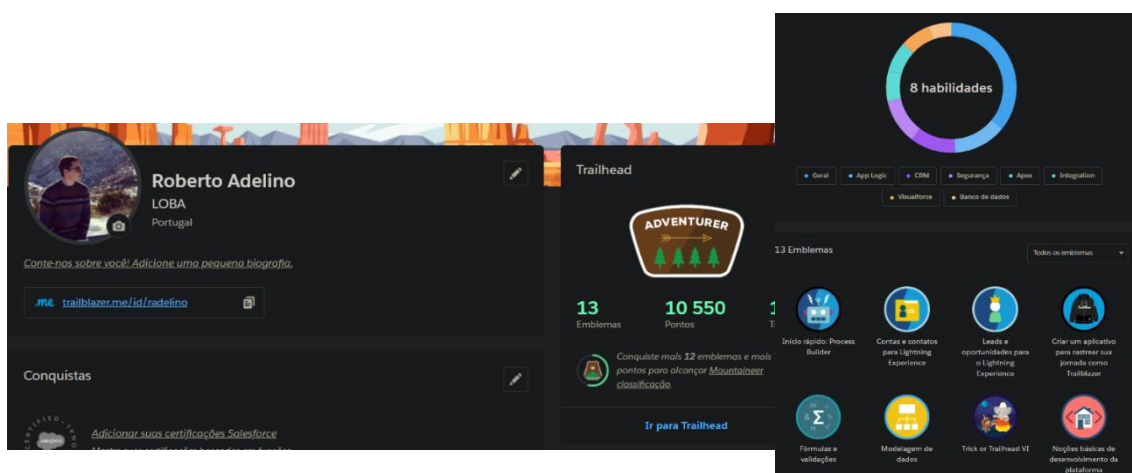


Figura 9-Trailhead

4 Análise de requisitos

Os requisitos de um sistema são a descrição estruturada do procedimento que é previsto antes deste ser desenhado, implementado e testado. Nesta etapa é feita a especificação do sistema e do que é que o sistema deve fazer, ou seja, os requisitos.

Neste capítulo vai ser utilizada a linguagem *UML (Unified Modeling Language)* para modelar a plataforma online. A *UML* é uma linguagem de modelação gráfica para especificação, visualização, construção e documentação de sistemas de *software* [18].

4.1 Diagrama de casos de uso

O diagrama de casos de uso auxilia no levantamento dos requisitos funcionais do sistema e a entender as interações que os atores vão ter com o sistema. Na Figura 10, observa-se o diagrama de casos de uso. Todos os casos de uso a ser desenvolvidos no projeto encontram-se dentro da fronteira. Na tabela 2 observam-se os atores e respetivos casos de uso da aplicação *Salesforce*.

Tabela 2- Tabela de atores e respetivos casos de uso

Atores	Casos de uso
Administrador	Contactar utilizadores da plataforma
	Definir regras para a atribuição de alimentos
	Gerir as operações da ONG
	Registar utilizadores
Voluntários	Registar-se na plataforma
	Entregar alimentos
	Confirmar entregas de alimentos
	Recolher alimentos de doações
Restaurantes/ Supermercados	Registar-se na plataforma
	Doar alimentos
Doadores	Doar dinheiro
	Doar alimentos
Famílias/ Instituições	Registar-se na plataforma
	Receber cabazes de alimentos
Parceiros	Registar-se na plataforma
	Comprar planos de sócio

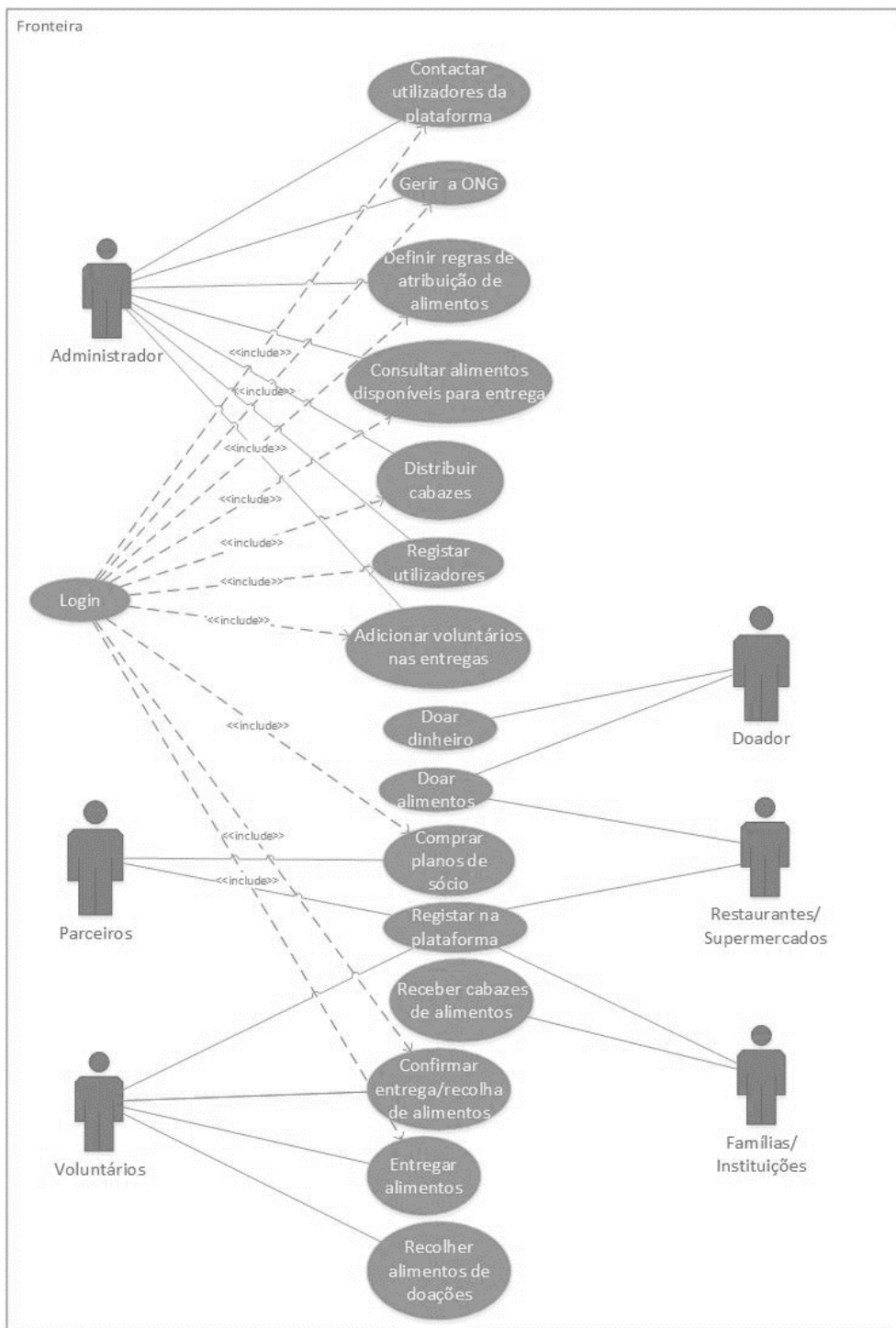


Figura 10-Diagrama de Casos de Uso

4.2 Descrição de casos de uso

Após o levantamento dos casos de uso percebemos a forma como os atores vão interagir com o sistema, é então necessário documentar os casos de uso mais importantes. Cada caso de uso é explicado de acordo com o fluxo de eventos no ponto de vista dos atores. Será detalhado o que o sistema tem de fornecer ao ator quando o caso de uso é executado.

A descrição de um caso de uso contém o fluxo normal de eventos, em que tudo corre de acordo com o esperado e o fluxo de eventos anormal, em que as coisas não correm da forma pretendida.

O *template* para a descrição de casos de uso é o seguinte:

- **Nome:** nome do caso de uso a descrever;
- **Descrição:** descrição do caso de uso em questão.
- **Pré-Condição:** condição inicial necessária para o caso de uso decorrer com sucesso;
- **Caminho Principal:** descrição do fluxo normal de eventos;
- **Caminhos Alternativos:** descrição do fluxo anormal de eventos;
- **Tamanho:** S, M, L, XL – classificação da complexidade do desenvolvimento do Caso de Uso;
- **Suplementos ou adornos:** descrição de testes a realizar, requisitos não funcionais.

As secções seguintes descrevem alguns dos casos de uso com o *template* apresentado anteriormente.

O título demonstra, claramente, o objetivo do caso de uso da Tabela 3. Trata-se da interface que permite fazer o pedido dos alimentos de acordo com a quantidade disponível e as regras estabelecidas.

Tabela 3- Descrição do caso de uso “Distribuir cabazes”

Actor Primário	Administrador
Nome	Distribuir cabazes
Descrição	O ator acede ao sistema para distribuir cabazes para famílias e instituições.
Pré-Condição	Login efetuado com sucesso
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona a interface Pedidos e vendas. 2. O ator clica em “New”. 3. O sistema pergunta que tipo de registo é. 4. O ator escolhe o tipo de registo “Pedido de alimentos”. 5. O Sistema devolve o formulário 6. O ator preenche o formulário e clica em guardar. 7. O sistema atribui automaticamente o número de cabazes de acordo com os atributos da família/instituição e das regras definidas. 8. O sistema devolve uma mensagem de confirmação.
Caminhos Alternativos	<ol style="list-style-type: none"> 1.a -Sistema Indisponível. 2.a -O ator cancela. 4.a-O ator escolhe outro tipo de registo. 4.b- O ator cancela. 6.a- O sistema devolve uma mensagem de erro por causa das validações de dados introduzidos. 7.a- O sistema devolve uma mensagem de erro quando não há cabazes suficientes disponíveis.
Tamanho	XL
Suplementos ou adornos	<p>Testar se funciona quando não há alimentos disponíveis</p> <p>Testar as várias regras de atribuição.</p>

Para complementar a descrição de caso de uso “Distribuir cabazes” (Tabela 3), o respetivo diagrama de sequência da Figura 14 apresenta o cenário principal do caso de uso.

A Tabela 4, mostra o caso de uso “Comprar plano de sócio”, em que os atores acedem ao site para se tornarem sócios.

Tabela 4- Descrição do caso de uso “Comprar plano de sócio”

Actor Primário	Parceiros
Nome	Comprar plano de sócio
Descrição	O ator pretende comprar um plano de sócio.
Pré-Condição	
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona a interface “<i>Tornar-se sócio</i>”. 2. O sistema devolve o formulário de registo. 3. O ator preenche o formulário e clica em guardar. 4. O sistema devolve o formulário com os planos de sócio. 5. O ator seleciona o tipo de plano pretendido e clica em guardar. 6. O Sistema mostra a uma mensagem de sucesso. 7. O ator recebe um email que indica que se tornou sócio.
Caminhos Alternativos	<ol style="list-style-type: none"> 1.a – Sistema Indisponível. 2.a – O ator sai da página. 3.a- O sistema mostra uma mensagem de erro de validações de dados. 5.a- O ator não seleciona nenhum plano. 6.a- O sistema mostra uma mensagem de erro. 7.a- O ator não recebe um email.
Tamanho	XL
Suplementos ou adornos	Testar se funciona com diferentes tipos de plano de sócio.

No caso de uso descrito na Tabela 5, o ator vai confirmar a entrega de um pedido de alimentos.

Tabela 5- Descrição do caso de uso “Confirmar entrega”

Actor Primário	Voluntários
Nome	Confirmar entrega de alimentos.
Descrição	O ator acede ao sistema para confirmar a entrega de alimentos.
Pré-Condição	Login efetuado com sucesso
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona o objeto “<i>Pedidos para entregar</i>” 2. O sistema mostra os registos 3. O ator seleciona o registo pretendido 4. O ator clica no botão “<i>Entregue</i>”
Caminhos Alternativos	<ol style="list-style-type: none"> 1.a – Sistema Indisponível 2.a- O ator fecha a página 3.a – O ator seleciona outro registo. 4.a – O ator fecha a página 4.b – o ator clica noutro botão
Tamanho	L
Suplementos ou adornos	Testar se atualiza o estado da entrega

Para complementar a descrição de caso de uso “Confirmar entrega de alimentos” (Tabela 5), o respetivo diagrama de sequência encontra-se na Figura 13 que apresenta o cenário principal do caso de uso.

A tabela 6 descreve o caso de uso da entrega de alimentos por parte do respetivo ator que são os voluntários.

Tabela 6- Descrição do caso de uso “Entregar alimentos”

Actor Primário	Voluntários
Nome	Entregar alimentos
Descrição	O ator vai fazer uma entrega de um pedido de alimentos.
Pré-Condição	Login efetuado com sucesso
Caminho Principal	<ol style="list-style-type: none"> 1. O ator recebe um email a indicar a morada, a quantidade e o nome da família/instituição para fazer a entrega. 2. O ator clica no objeto “<i>Pedidos para entregar</i>”. 3. O sistema mostra os registos. 4. O ator seleciona o registo pretendido. 5. O ator Clica no botão “Em distribuição”.
Caminhos Alternativos	<ol style="list-style-type: none"> 1.a – Sistema Indisponível. 1.b – O ator sai da página. 2.a - Sistema indisponível. 2.b – O ator sai da página. 3.a- O ator seleciona outro registo. 3.b- O ator sai da página. 4.a- O ator não clica no botão.
Tamanho	L
Suplementos ou adornos	Testar se é possível mudar o estado da entrega sem ter um voluntário associado

Para complementar a descrição de caso de uso “Entregar alimentos” (Tabela 6), o respetivo diagrama de sequência na Figura 12 apresenta o cenário principal do caso de uso.

No caso de uso descrito na Tabela 7 é exposto o processo de doação de dinheiro.

Tabela 7- Descrição do caso de uso “Doar dinheiro”

Actor Primário	Doador
Nome	Doar dinheiro
Descrição	O ator faz uma doação monetária.
Pré-Condição	Login efetuado com sucesso
Caminho Principal	<ol style="list-style-type: none"> 1. O ator clica na interface “Doar dinheiro” do <i>website</i>. 2. O sistema devolve o formulário. 3. O ator preenche o formulário e clica no botão “Doar”. 4. O ator recebe um email com a indicação de que a doação foi completada.
Caminhos Alternativos	<ol style="list-style-type: none"> 1.a-O ator clica noutra interface. 1.b-O ator sai do <i>website</i> 2.a-O ator sai do <i>website</i> 3.a-O ator sai do <i>website</i> 3.b- O ator introduz dados incorretos e o sistema devolve uma mensagem de erro 4.a-O ator não recebe um email.
Tamanho	L
Suplementos ou adornos	Testar doações com quantidades negativas

A tabela 8 descreve o caso de uso de quando se faz uma doação de alimentos através do website da ONG.

Tabela 8-Descrição do caso de uso “Doar alimentos” através do website.

Actor Primário Doador	
Nome	Doar alimentos
Descrição	O ator faz uma doação de alimentos
Pré-Condição	
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona a interface “<i>Doações de alimentos</i>” 2. O sistema devolve o formulário para preencher. 3. O ator fornece os dados. 4. O ator clica em guardar. 5. O sistema mostra uma mensagem de sucesso, envia uma notificação para o administrador e cria uma tarefa para um voluntário recolher os alimentos.
Caminhos Alternativos	<ol style="list-style-type: none"> 1.a – Sistema Indisponível 2.a- O ator sai da página 3.a – O ator sai do formulário de preenchimento 4.a – Se houver campos vazios ou incorretos, o sistema mostra uma mensagem de erro 5.a O sistema mostra uma mensagem de erro
Tamanho	L
Suplementos ou adornos	<p>Testar as validações dos dados do formulário.</p> <p>Testar se o sistema insere o alimento sem dados.</p> <p>Testar se envia notificação e cria uma tarefa.</p>

Para complementar a descrição de caso de uso “Doar alimentos” (Tabela 8), o respetivo diagrama de sequência na Figura 11 apresenta o caso de uso.

A tabela seguinte demonstra como é feita uma doação de alimentos por parte de um Restaurante ou Supermercado

Tabela 9- Descrição do caso de uso “Doar alimentos”

Actor Primário Supermercado/Restaurante	
Nome	Doar alimentos
Descrição	O ator faz uma doação de alimentos
Pré-Condição	Estar registado na plataforma através do Administrador
Caminho Principal	<ol style="list-style-type: none"> 1. O ator contacta o administrador para fazer uma doação 2. O administrador seleciona o objeto “<i>Doações de alimentos</i>” 3. O sistema abre o formulário para preencher. 4. O ator fornece os dados ao administrador 5. O administrador preenche o formulário. 6. O administrador clica em guardar. 7. O sistema envia uma notificação para o administrador, envia um email ao ator com os dados da doação e cria uma tarefa para um voluntário recolher os alimentos.
Caminhos Alternativos	<ol style="list-style-type: none"> 1.a – O ator não contacta o administrador 2.a- O administrador sai da página 3.a – O ator sai da página 4.a- O ator fornece dados incorretos 6.a – Se houver campos vazios ou incorretos, o sistema mostra uma mensagem de erro
Tamanho	L
Suplementos ou adornos	<p>Testar as validações dos dados do formulário.</p> <p>Testar se envia notificação e cria uma tarefa.</p>

4.3 Diagrama sequência

Os diagramas de sequência apresentam as interações de objetos dispostas numa sequência temporal. Pode-se utilizar o fluxo de eventos para determinar quais os objetos e interações que é preciso para completar a funcionalidade especificada pelo fluxo de eventos. Os diagramas de sequência são bons para mostrar aquilo que se está a passar, para chegar a requisitos e para trabalhar com os clientes.

A Figura 11, mostra o diagrama de sequência para o caso de uso “Doar alimentos” do ator Doador.

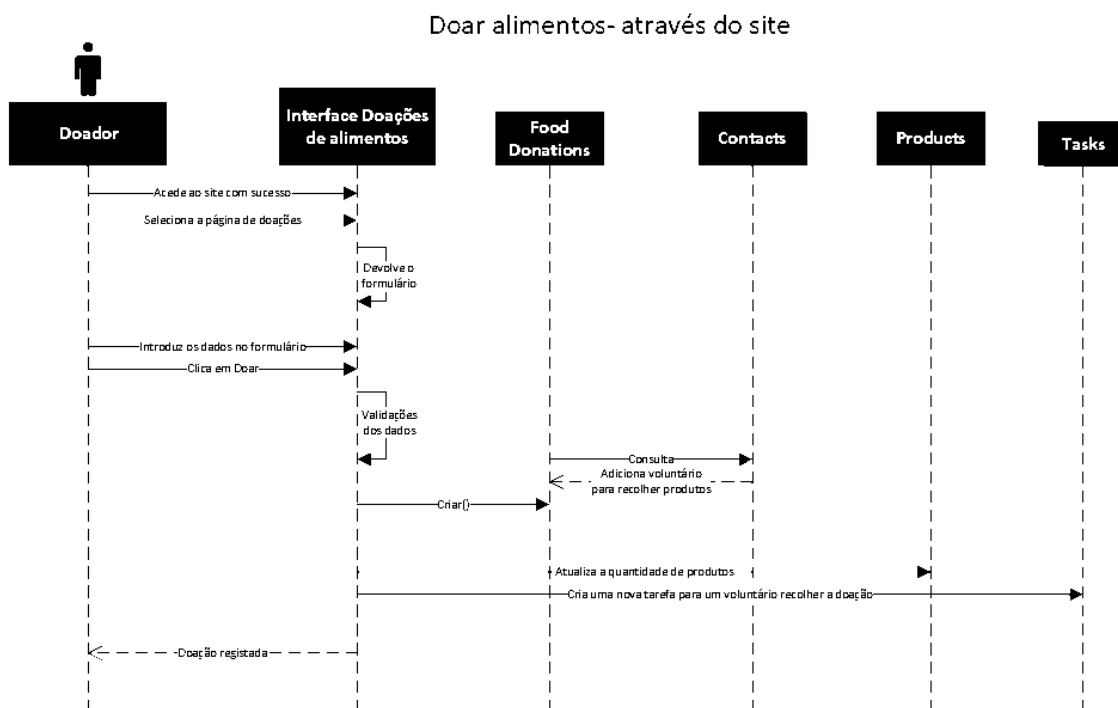


Figura 11-Diagrama de sequência “Doar alimentos” através do website

A Figura 12, mostra o diagrama de sequência para o caso de uso” entregar alimentos”.

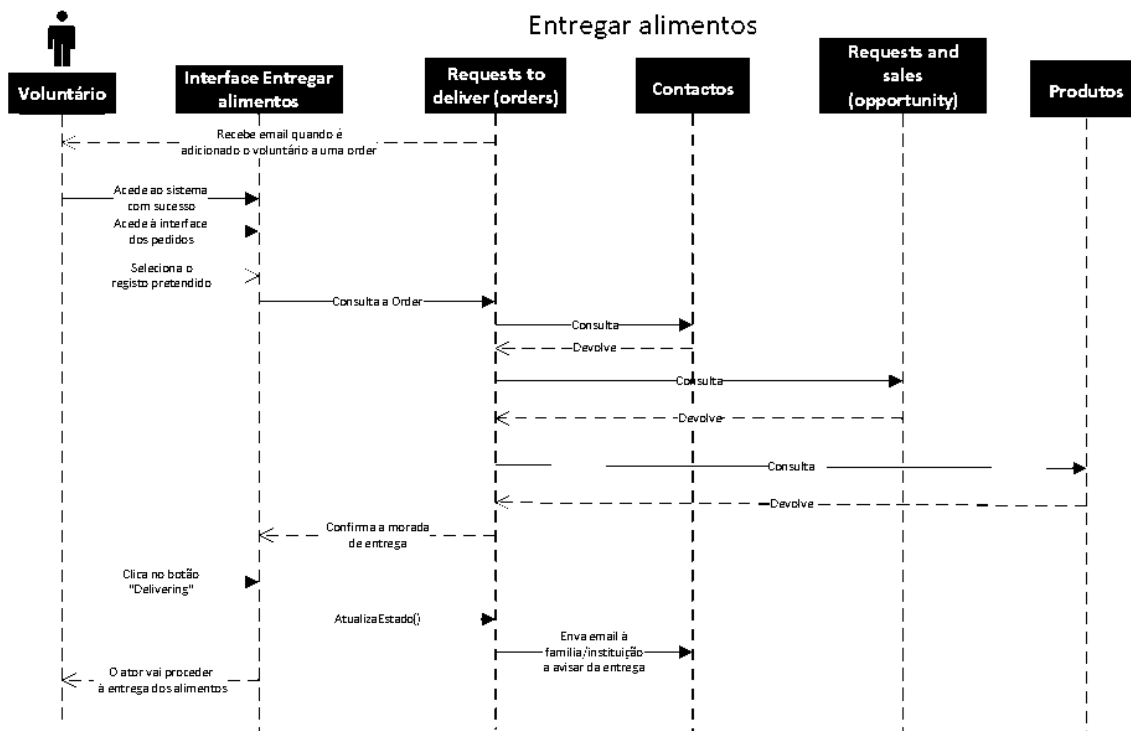


Figura 12- Diagrama de sequência “Entregar alimentos

Figura 13, mostra o diagrama de sequência para o caso de uso” confirmar entrega de alimentos”.

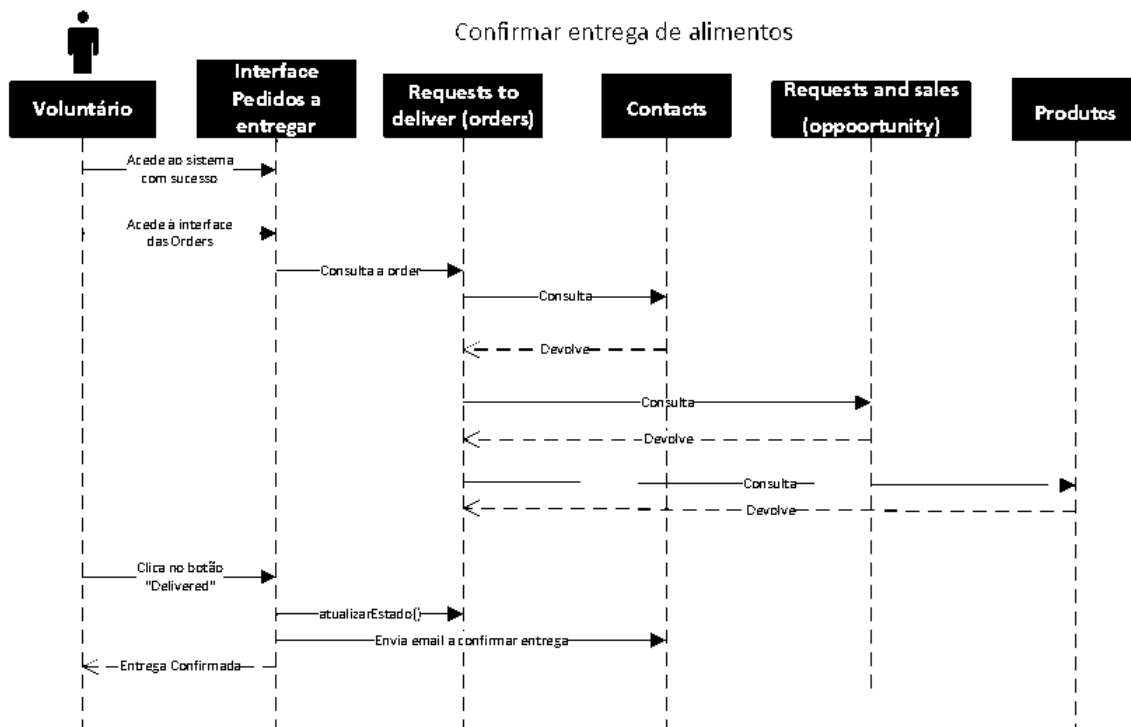


Figura 13- Diagrama de sequência “Confirmar entrega de alimentos”

A figura 14 demonstra como é o processo de distribuição automática dos alimentos pelas famílias e instituições.

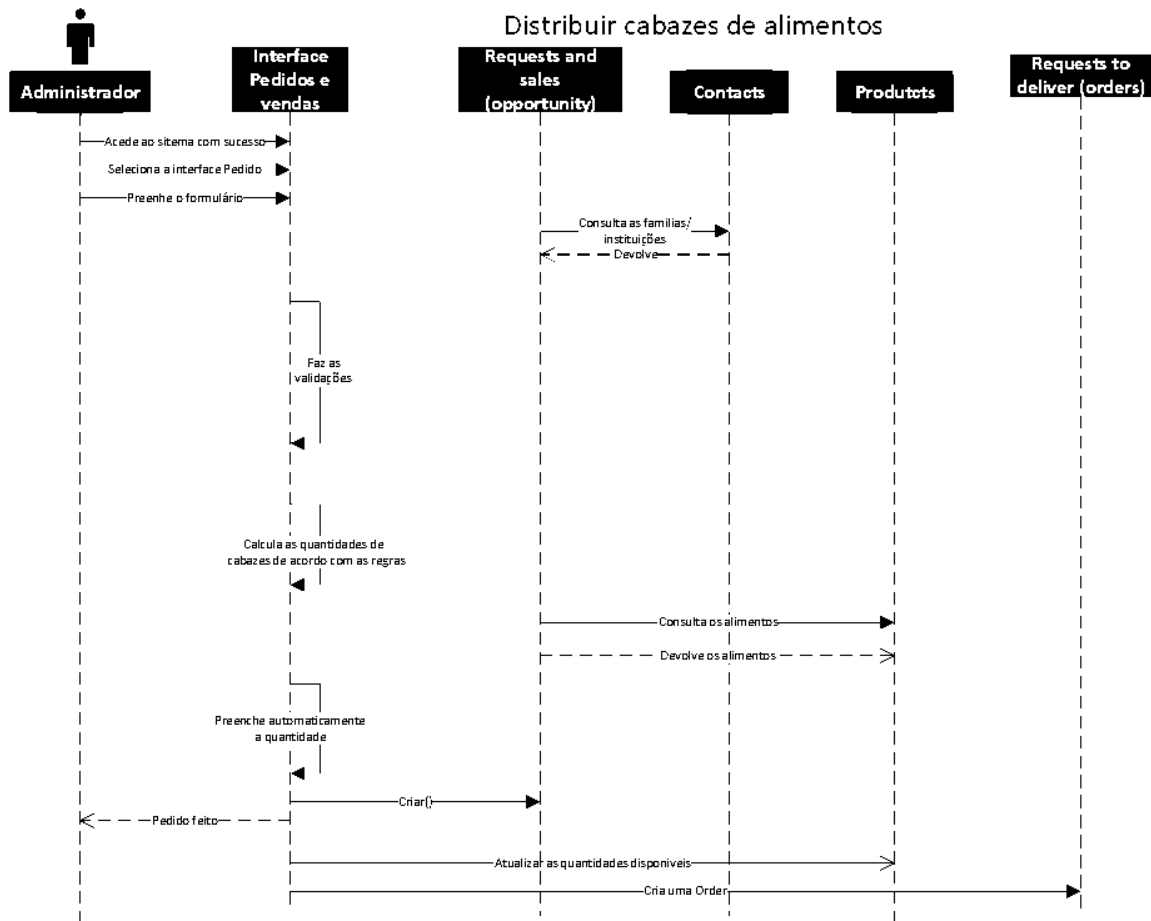


Figura 14- Diagrama de sequência “Distribuir cabazes de alimentos”

4.4 Tabela de classes/objetos

Para analisar os requisitos a serem utilizados no projeto, teve de haver um estudo das entidades a utilizar.

As entidades, também conhecidas por tabelas, em *Salesforce* denominam-se por objetos. No *Salesforce* existem objetos *standard*, ou seja, objetos pré-existentes que podem ser utilizados e modificados de acordo com o projeto a realizar. Neste projeto foram utilizados objetos *Standard* e objetos *Custom*. Esses objetos são expostos na Tabela 10.

Tabela 10-Objetos/tabelas da aplicação

	Objetos	Descrição
Standard	<i>Accounts</i>	Tipos de contas, cada contacto vai fazer parte de uma conta.
	<i>Leads</i>	Potenciais clientes como por exemplo parceiros, famílias, doadores, restaurantes, etc. As leads quando são aceites, serão ser convertidas a <i>contacts</i> .
	<i>Contacts</i>	Utilizadores que já estão registados na plataforma, ou seja, são leads que foram convertidas
	<i>Requests and Sales (Opportunity)</i>	Fluxo de vendas e de doações por parte da Organização, por exemplo a venda de planos de sócio ou de doação de cabazes a famílias. Originalmente o objeto tem o nome de “ <i>opportunity</i> ”, mas este nome foi alterado para fazer mais sentido para o projeto desenvolvido.
	<i>Requests to deliver (Orders)</i>	Registo das entregas de cabazes a famílias e instituições. Originalmente este objeto tem o nome de “ <i>Orders</i> ”
	<i>Campaigns</i>	Campanhas para a angariação de fundos ou clientes
	<i>Tasks</i>	Gestão de tarefas, como por exemplo, denominar um voluntário para fazer uma entrega ou uma recolha de alimentos
	<i>Products</i>	Registo dos produtos disponíveis
Custom	<i>Money Donations</i>	Doações de fundos monetários à ONG.
	<i>Food Donations</i>	Doações externas de alimentos, por exemplo por parte dos restaurantes ou supermercados

4.5 Diagrama de classes

Nesta secção, apresenta-se o diagrama de classes da plataforma - modo simplificado, representado pela Figura 15. O diagrama de classes mostra como as diferentes classes se relacionam entre si. Como se pode visualizar na Figura 5 há classes que têm vários relacionamentos como é o caso da classe “*Contacts*” que tem relacionamentos “um para muitos” com as classes: “*Food Donations*”, “*Money Donations*”, “*Requests to deliver*”, “*Requests and sales*”; tem relacionamentos “muitos para um” com a classe “*Accounts*” e há classes que têm apenas 1 relacionamento como é o caso da classe “*Campaigns*” que apenas tem um relacionamento ‘um para muitos’ com a classe das “*Leads*”.

Cada classe é constituída pelo nome, os respetivos atributos e por fim as operações.

O diagrama completo da aplicação foi gerado automaticamente pelo *Salesforce* e é chamado *SchemaBuilder* onde é possível observar os vários tipos de relacionamentos entre os objetos. Pode ser consultado no Anexo A 1

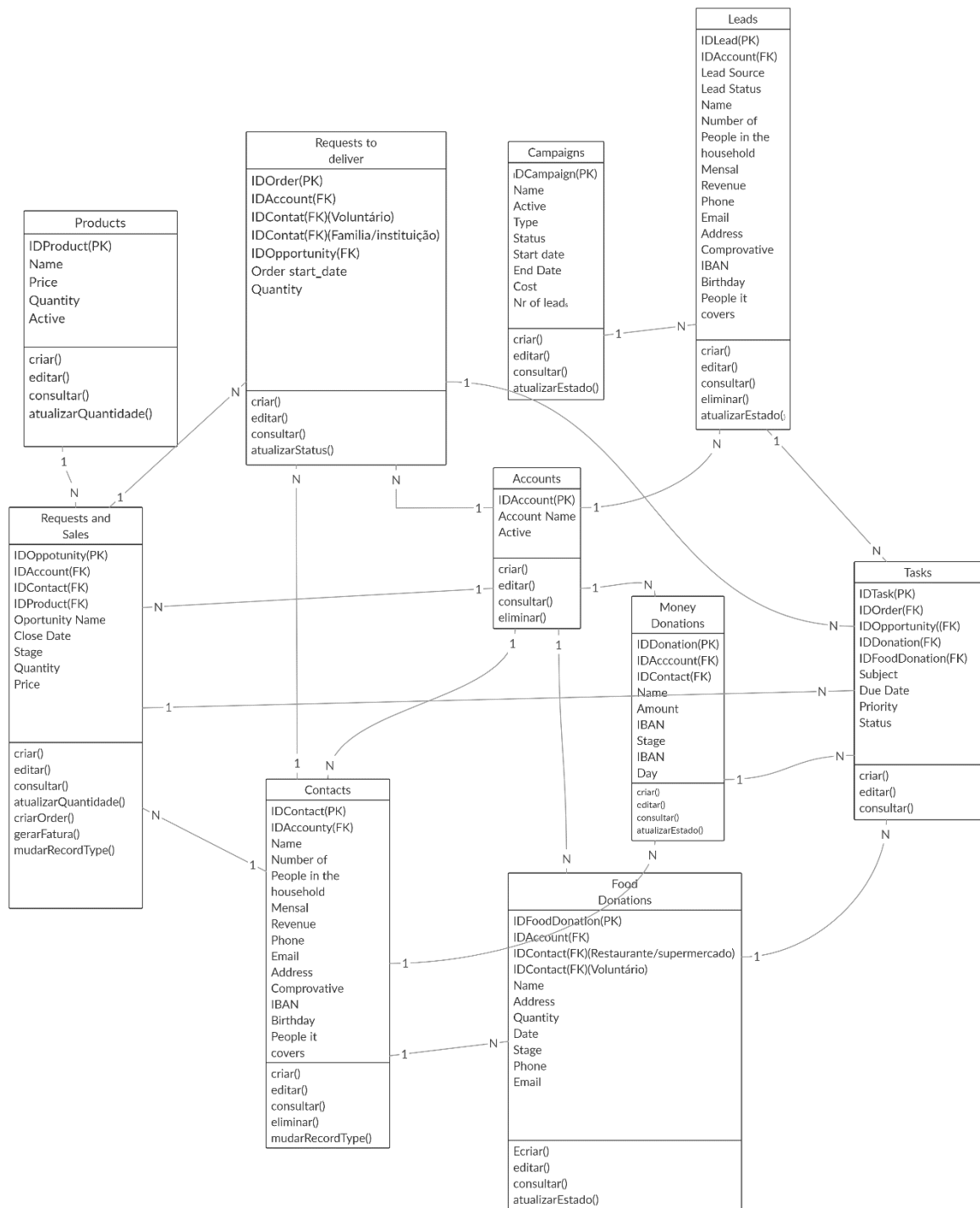


Figura 15-Diagrama de classes simplificado

4.6 Dicionário de dados

O dicionário de dados descreve todos os campos do diagrama de classes, dizendo qual o tipo de dados de cada campo sendo estes:

- O nome
- A descrição
- O tipo de dados
- O tamanho
- O formato
- Restrições

Também serão descritos os métodos disponíveis em cada uma das classes.

Todos estes aspetos estão representados no Anexo A 2.

5 Tecnologias

5.1 Tecnologias utilizadas

Com base na análise de requisitos (Capítulo 4), procedeu-se ao desenvolvimento da aplicação em *Salesforce*. Neste capítulo, são apresentadas as várias tecnologias e ferramentas utilizadas no desenvolvimento deste projeto. Durante a implementação, foi necessário fazer certos ajustes aos requisitos definidos para que as funcionalidades ficassem utilizáveis.

5.1.1 *Salesforce*

A plataforma *Salesforce* é a número 1 em plataformas de CRM (*Customer Relationship Management*). Destaca-se ainda das restantes plataformas por ser um serviço baseado na *Cloud*, porém o serviço é mais rápido, as empresas nunca ficam com o serviço em baixo e possibilita o acesso à plataforma através qualquer equipamento desde que tenha acesso à internet. O *Salesforce* também ajuda a automatizar, mas com menos código. Mas com o mesmo nível de complexidade. Uma vez que é na *cloud*, o *Salesforce* tem vários serviços [19]sendo alguns deles:

- Vendas - vender rápido e de maneira inteligente;
- Serviços - gerir o suporte a clientes de todos os canais. (mensagens, emails, redes sociais);
- Marketing - Enviar mensagens personalizadas em qualquer canal (mensagens, emails, redes sociais);
- *Commerce* - Tornar única a experiência do comprador;
- Quip - Criar, editar, discutir e organizar o trabalho em equipa, tudo num só sítio;
- *Pataform* - Criar, ligar e integrar aplicações.

5.1.2 *Visualforce*

O *Visualforce* [20] é a estrutura que permite aos programadores criar interfaces para o utilizador de forma personalizada e sofisticada. Podem ser alojadas nativamente na plataforma *Lightning*. Esta estrutura do *Visualforce* inclui uma linguagem de marcação baseada em *tags*, muito semelhante ao HTML. Além disso, tem também um conjunto de “controladores *standard*” do lado do servidor que tornam as operações básicas da base de dados, como consultas e inserções, simples de executar.

Na linguagem de marcação, cada *tag* corresponde a um componente de interface de utilizador, por exemplo uma seção de uma página, uma lista relacionada ou um campo. O comportamento dos componentes do Visualforce pode ser controlado pela mesma lógica usada nas páginas padrão do *Salesforce* ou os programadores podem associar a própria lógica a uma classe de controlador gravada no Apex.

5.1.3 Apex

O *Apex* [21], é uma linguagem de programação orientada a objetos, usada pela plataforma do *Salesforce*. Permite aos programadores executar instruções de controlo de transações e fluxos em servidores *Salesforce*, sendo possível fazer chamadas à API. A sintaxe do Apex é muito parecida com o a do Java. O código Apex pode ser iniciado por pedidos ao serviço Web e por *triggers* em objetos.

5.1.4 HTML

O HTML, acrónimo de *Hypertext Markup Language*, é a linguagem de marcação padrão para páginas web. O HTML foi a linguagem usada para a criação dos elementos fundamentais de uma página web recorrendo a *tags*, que posteriormente são interpretadas por um web *browser* e apresentadas ao utilizador.

5.1.5 CSS

O CSS (*Cascading Style Sheets*) é uma linguagem de folhas de estilo usada para especificar de que forma alguns elementos da interface do utilizador são apresentados, em termos visuais. Estes documentos são escritos utilizando uma linguagem de marcação, o HTML, sendo neste caso usada nas páginas Visualforce.

5.1.6 Visual Studio Code

O *Visual Studio Code* [22] fornece um editor de código, ferramentas de automação de compilação, um depurador e o preenchimento de código inteligente, ou seja, um IDE (ambiente de desenvolvimento integrado). O Controlo das aplicações a desenvolver é feito através da CLI do *Salesforce*, conectando o mesmo ao IDE, neste caso ao *Visual Studio Code*, podendo assim sincronizar meta dados entre organizações e fornecendo um sistema de controlo de versões.

5.1.7 Developer Console

O *Developer Console* é um IDE que faz o mesmo trabalho que o Visual Studio Code, sendo este uma interface do *Salesforce*, que atua no *browser* na plataforma *Salesforce*, o que faz com que não haja necessidade de instalar o *Visual Studio Code*.

6 Implementação

Com base no estudo das aplicações existentes no mercado e tendo em conta a análise de requisitos segue-se o desenvolvimento da aplicação *Salesforce ZeroWaste*.

Numa aplicação *Salesforce* existem vários critérios essenciais usados, sendo estes, *record types*, *page layouts*, regras de validação, páginas VisualForce externas, entre outros. Nesta parte do relatório são descritos todos estes parâmetros que foram utilizados na realização deste projeto.

6.1 *Record Types e Page Layouts*

Para o desenvolvimento da aplicação são necessários *record types* (tipos de registos). Os *record types* permitem que haja diferentes processos de negócios, *picklists* e layouts de página para diferentes registos num objeto. No caso deste projeto, existe a obrigatoriedade de ter diferentes tipos de contactos, sendo eles instituições, famílias, voluntários, restaurantes, supermercados, sócios e doadores e também diferentes tipos de registos no objeto das vendas e pedidos sendo ou de pedidos de alimentos ou de venda de planos de sócio, uma vez que será possível fazer a venda de planos de sócio e também fazer a distribuição dos cabazes pelos necessitados sem que estes tenham de pagar. A aplicação vai então ter *layouts* de página diferentes para cada tipo de registo no objeto contacto e no objeto pedidos e vendas.

6.2 Regras de Validação

As regras de validação verificam se os dados que o utilizador insere num registo estão de acordo com os padrões especificados. Uma regra de validação pode conter uma fórmula ou expressão que avalia os dados em um ou mais campos e devolve um valor de “Verdadeiro” ou “Falso”. As regras de validação mostram uma mensagem de erro ao utilizador quando retornam “Verdadeiro”, este valor representa que os dados introduzidos são inválidos. Foram implementadas diversas validações nos campos dos objetos.

6.3 *Process Builders*

Há certas tarefas que costumam ser realizadas pelos utilizadores podendo ir de tarefas simples a tarefas complexas. No *Salesforce* é possível, configurar processos para fazer tudo de modo automático em vez do trabalho ser feito de forma manual e repetitiva. Os *Process Builders* ajudam na automatização de processos e fornecem uma interface gráfica quando criados. Esta ferramenta suporta três tipos de automação de processos que determinam quando começa o processo. Esses tipos podem ser:

- Quando um registo é criado ou atualizado.
- Quando ocorre um evento de plataforma.
- Quando outro elemento o invoca.

Cada processo vai conter critérios que vão determinar quando uma ou mais ações devem ser executadas, e contêm grupos de ações, que consistem em ações imediatas ou agendadas, sendo estas apenas em processos de modificação de registos. No projeto, sempre que possível, criaram-se processos deste género, de forma a facilitar o trabalho.

6.4 Flows

Os *Flows* são uma aplicação dentro do *Salesforce* para automatizar processos de negócios complexos. Simplificando, eles coletam dados e fazem algo com eles. O *Flow Builder* é a interface declarativa que usamos para construir fluxos individuais. O *Flow Builder* pode ser usado para construir lógica semelhante a código sem a necessidade de conhecer uma linguagem de programação.

Os fluxos no *Salesforce* podem ser implementados de duas maneiras:

- Fluxos de tela
- Fluxos lançados automaticamente

6.5 Triggers

Embora os *process builders* sejam uma ferramenta poderosa, nem sempre possibilitam a criação de todos os requisitos necessários. Os *triggers* são criados com recurso à linguagem Apex e podem ser configurados para realizar ações personalizadas antes ou depois da alteração dos registos, tal como *inserts*, *deletes* ou *updates*.

6.6 Classes Apex (Controladores)

Tal como em Java, no Apex é possível a criação de classes. Uma classe é um conjunto de instruções. Neste projeto as classes foram utilizadas para especificar o que acontece quando um utilizador interage com os componentes associados a uma página Visualforce, por exemplo, quando um utilizador clica num botão ou link. Os controladores também fornecem acesso aos dados que devem ser exibidos numa página e podem modificar o comportamento do componente.

6.7 VisualForce Pages

O *Visualforce* permite a criação de páginas e interfaces para o utilizador interagir. As *Visualforce pages* têm dois elementos principais que são a marcação do *Visualforce* e o controlador do *Visualforce*.

Neste projeto foi criado um website através das *visualforce pages*.

6.8 Email-alerts/email templates

Os alertas de email são emails criados automaticamente por processos, estes emails serão enviados a destinatários definidos no alerta. Os emails podem ser criados pelo programador que define que tipo de ações causarão o envio destes emails. Podem estar associados a processos, fluxos, regras de fluxo de trabalho e processos de aprovação.

6.9 Reports e Dashboards

Um *Report* é uma lista de registos que vai de encontro aos critérios definidos pelo utilizador. É exibido no *Salesforce* em linhas e colunas, e pode ser filtrado, agrupado ou aplicado em gráficos. Todos os *Reports* são armazenados numa pasta. As pastas podem ser públicas, ocultas ou compartilhadas, e ser definidas como somente leitura ou leitura / gravação.

Um *dashboard* é uma exposição visual de tendências e métricas chave para registos na organização. O relacionamento entre um *report* e um *dashboard* é 1:1, ou seja, para cada *dashboard* há um único *report* relacionado. Contudo pode-se usar o mesmo *report* em múltiplos *dashboards*. Múltiplos *dashboards* e *reports* podem ser mostrados juntos num único layout de página, criando uma forte exibição visual.

6.10 AppExchange

A *AppExchange* é a loja do *Salesforce*. É um ecossistema de mais de 5.000 recursos prontos para instalar como é o caso de: componentes, *Lightning Data*, *Bolt Solutions* e *Flows*, que permitem aumentar as funcionalidades de uma aplicação.

6.11 Recursos estáticos e tema

Os recursos estáticos são recursos que podem ser imagens, ficheiros CSS, ficheiros de *javascript*, *.zip*, *.jar*, etc. Neste projeto foi utilizada uma imagem para o ícone da aplicação na plataforma de *Salesforce*.



Figura 16-Logotipo da organização

Foi também criado um tema com o logotipo da ONG e uma imagem simples de modo a personalizar ainda mais a organização e a tornar a aplicação visualmente apelativa.

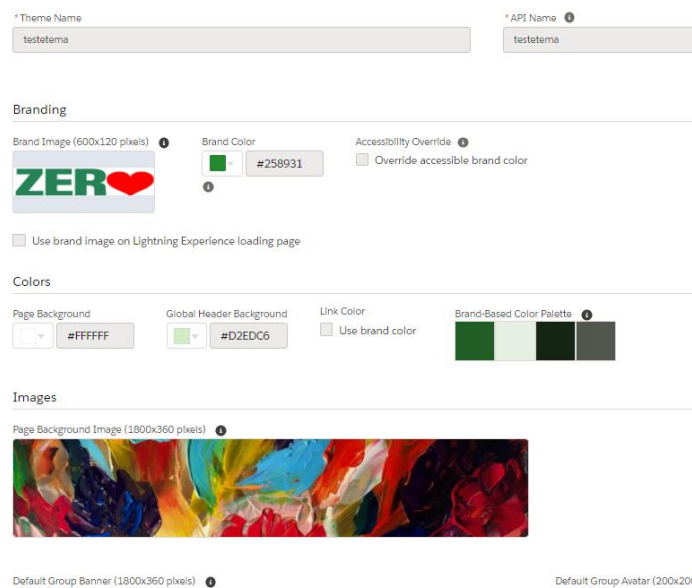


Figura 17-Tema criado

6.12 Home page da organização

A página inicial da aplicação *Salesforce* foi customizada (Figura 18) tornando-a visualmente apelativa e fácil de entender. No topo foi colocada uma *preview* do website, foram também criados vários gráficos que informam de dados importantes da ONG e que permitem uma gestão mais rápida e fácil de todas as operações da ONG.

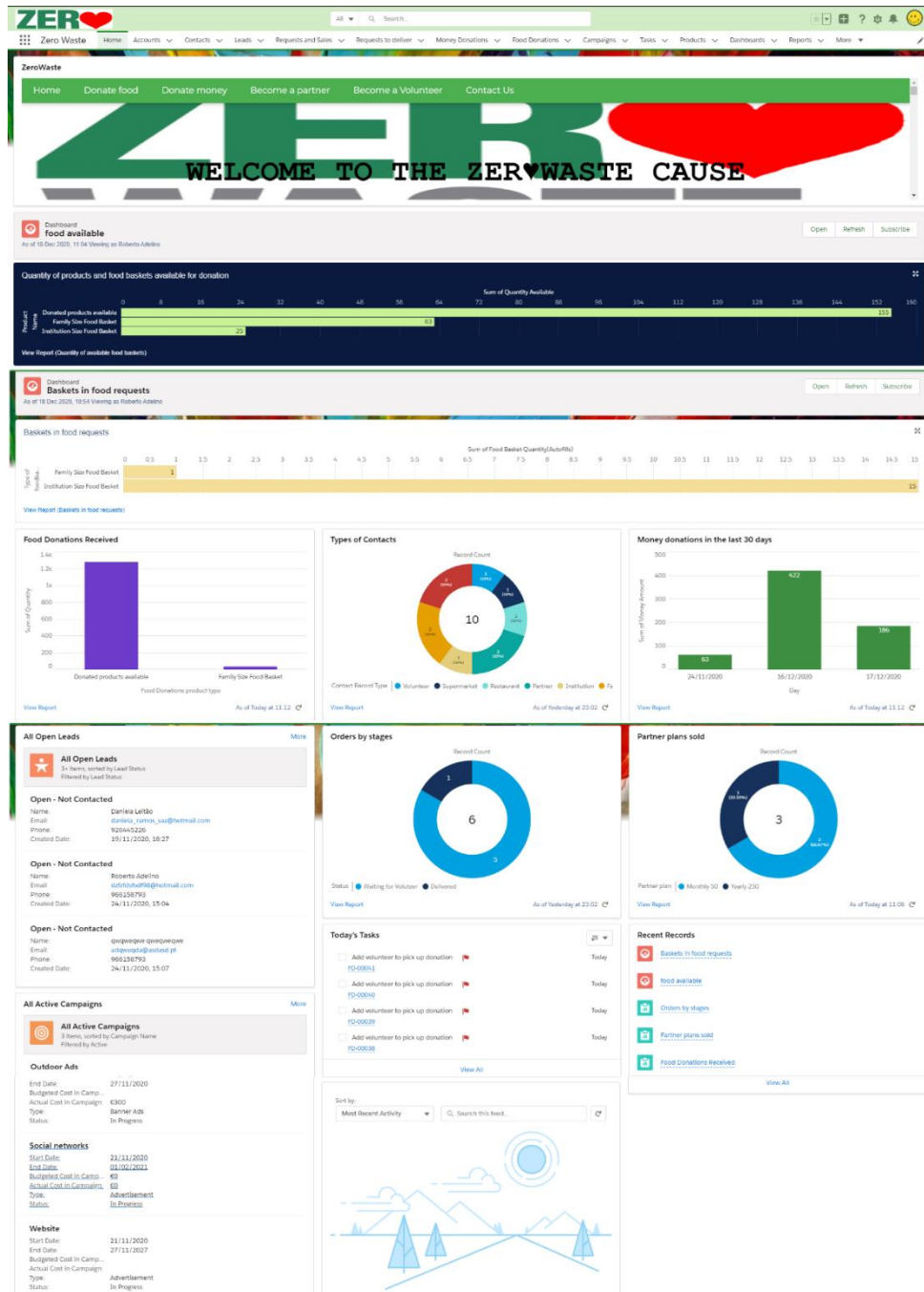


Figura 18-Página inicial no *salesforce*

6.13 Objetos *custom*

O *Salesforce* dá suporte a vários tipos de objetos diferentes. Eles podem ser objetos *standard*, *custom*, externos e *BigObjects*.

Os objetos *standard* são objetos pré criados e fornecidos pelo *Salesforce*, objetos *Custom* ou personalizados, são objetos criados para armazenar informações específicas da plataforma desenvolvida e que ainda não existem, Objetos externos são semelhantes a objetos *custom*, exceto pelo fato de serem mapeados para dados armazenados fora do *Salesforce*, *BigObjects* são objetos que arquivam e gerem grandes volumes de dados dentro do *Salesforce* sem afetar o desempenho, tudo com uma escala de processamento de milhares de milhões de registros.

6.13.1 *Money donations*

Este objeto serve para receber doações de dinheiro de pessoas que podem ou não estar registadas nos contactos. As doações podem ocorrer através do website ou através da aplicação *Salesforce*.

6.13.1.1 *Validation rules*

As doações podem ser de pessoas que não estão registadas na plataforma, mas quando são de pessoas registadas apenas faz sentido serem de contactos com o *record type* ‘*Donor*’ e para tal foi criado um filtro/regra no campo :



Figura 19-Filtro no campo dos doadores

Para o campo de ‘*Money Amount*’ foi criada uma *validation rule* (Figura 20) que não permite que seja inserida uma quantidade negativa e no layout da página o campo foi definido como *read-only*, para não permitir que esse valor seja posteriormente alterado por outra pessoa.

Validation Rule Detail	
Rule Name	moneyAmountBiggerThan0
Error Condition Formula	Money_Amount__c < 0
Error Message	You must enter a positive amount!

Figura 20-Regra de validação no campo *Amount*

Foi também definida uma regra (Figura 21) para o formato do IBAN:

Error Condition Formula	NOT(REGEX(IBAN__c, "[A-Z]{2}[0-9]{23}"))
Error Message	Please enter a valid IBAN number. (AZ0000000000000000000000).

Figura 21-Regra de validação no formato do IBAN

Foi definida também, uma regra para a data (Figura 22) para que não possa ser diferente do dia em que o registro é criado:

Rule Name	data
Error Condition Formula	Day__c <=> Today()
Error Message	Should be todays date

Figura 22-Regra de validação no campo da data

6.13.1.2 Process builders

A figura 23 mostra o *process builder* que envia uma notificação sempre que ocorre uma nova doação, a notificação será mostrada no topo da página ao administrador da organização.

The image shows two parts of the Salesforce interface. On the left is the Process Builder flowchart for 'new money donation notification'. It starts with a 'START' event, followed by a 'Money Donation' object. A decision diamond labeled 'new donation' has a 'TRUE' path leading to 'IMMEDIATE ACTIONS' which includes a 'Notification' action. A 'TIME LAPSE' action follows. The 'FALSE' path exits the process. On the right is the configuration for the 'Send Custom Notification' action. The 'Action Name' is 'Notification'. The 'Notification Type' is 'New money donation'. The 'Notification Recipient' is 'User' and the 'User' is 'Current User'. The 'Notification Title' is 'New money donation on {[MoneyDonations__c].Day__c}.'. The 'Notification Body' is '{[MoneyDonations__c].Name__c} / {[MoneyDonations__c].Donator__c.FirstName} just made a {[MoneyDonations__c].Money_Amount__c} donation.'. The 'Target Object' is 'MoneyDonations__c Object that started the process'.

Figura 23-Process builder de notificação de nova doação

Foi criado um segundo *process builder* (Figura 24) que envia um email que foi previamente criado nos templates, ao doador quando o estado da doação passa a ‘*Transaction completed*’ que serve como aviso de que o pagamento foi concluído com sucesso.

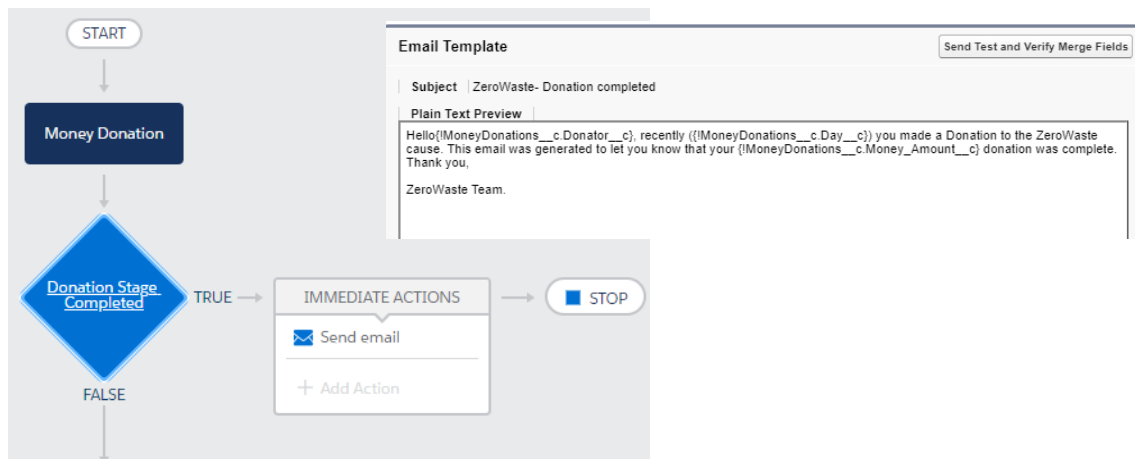


Figura 24-Process builder que envia um email e respetivo template criado

Também se criou um *process builder* que cria uma *task* sempre que é feita uma doação. Esta *task* servirá para se proceder ao processamento da doação (Figura 25).

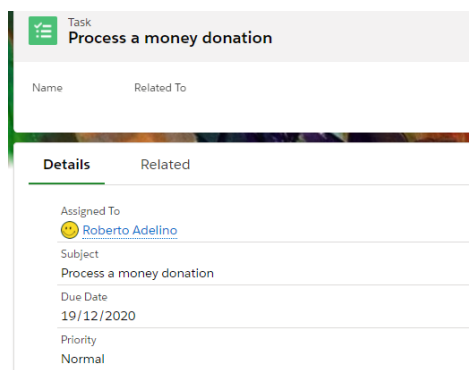


Figura 25-Tarefa criada

6.13.1.3 Reports e dashboards

Foi criado um *report* (Figura 26) na página inicial que mostra todas a doações feitas nos últimos 30 dias

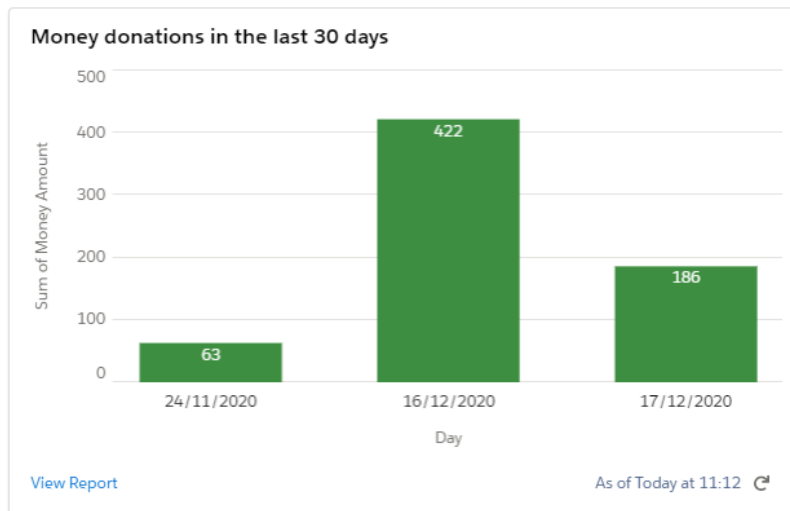


Figura 26-Report das doações

6.13.2 Food Donations

Este objeto serve para receber doações de cabazes de pessoas que podem ou não estar registadas nos contactos através do website ou então de restaurantes e de supermercados, tendo estes que estar registados na plataforma para o processo ocorrer de forma mais direta.

Existe um *path* no topo da página que permite que ao se clicar em cada botão, o estado da doação seja atualizado, ativando assim os automatismos para cada estado, desta maneira será mais fácil controlar quando as doações estão completas.

The screenshot displays a detailed view of a food donation record. The 'Details' section includes the following information:

- Food Donation Number:** FD-00030
- Donator:** PINGO DOCE GUARDA
- Date:** 23/11/2020
- Food Basket Type:** Family Size Food Basket
- Quantity:** 20
- Stage:** Details provided
- Address (City and Street):** Bairro S. Domingos Quinta do Carmalho, 6300-804 São Miguel da Guarda
- Volunteer to pick up:** Roberto Adelinho
- Phone:** 966158793
- Name:** Roberto Adelinho
- Owner:** Roberto Adelinho
- Created By:** Roberto Adelinho, 23/11/2020, 11:30
- Last Modified By:** Roberto Adelinho, 23/11/2020, 11:31

The 'Food Donation History (1)' section shows one entry with the following details:

- Date:** 23/11/2020, 11:30
- From:** Created
- User:** Roberto Adelinho
- Original Value:**
- New Value:**

The 'Activity' section includes an email field with a 'Compose' button and filters for 'All time', 'All activities', and 'All types'. The 'Upcoming & Overdue' section indicates 'No next steps' and 'No past activity'.

Figura 27-Página de um registo de doação de alimentos.

6.13.2.1 Validation rules

Existe uma validação no campo dos cabazes que não permite que o campo fique vazio (Figura 28). Existem também regras semelhantes à do objeto “*Money donations*” tais como, não ser possível inserir quantidades negativas e a data ter de ser a do próprio dia.

Validation Rule Detail	
Rule Name	foodbasketObrigatorio
Error Condition Formula	ISBLANK(Food_Basket_Type__c)
Error Message	Enter a food basket type.

Figura 28- Regra de validação no campo dos cabazes.

Por fim existe um filtro no campo do voluntário para só ser possível colocar contactos com o *record type* “*Volunteer*” como mostra a Figura 29.

Filter Criteria	Volunteer to pick up: Contact Record Type EQUALS Volunteer
Filter Type	Required. The user-entered value must match filter criteria.
Error Message	Enter a volunteer account

Figura 29-Filtro no campo dos voluntários

6.13.2.2 Process builders

O *process builder* apresentado na seguinte figura 30 mostra que vai ser enviada uma notificação ao administrador do sistema sempre que é feita uma nova doação de cabazes.

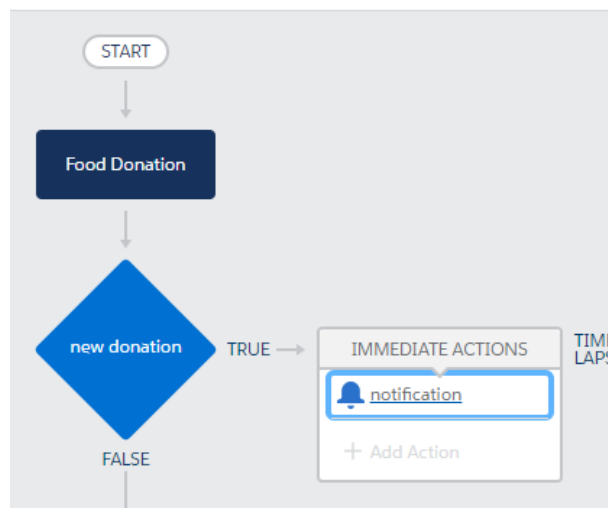


Figura 30-Process builder que envia uma notificação.

O *process builder* da figura 31 atualiza a quantidade total de produtos disponíveis no stock da ONG quando o estado da doação de alimentos de um restaurante ou supermercado passa a “*Donation received*”.

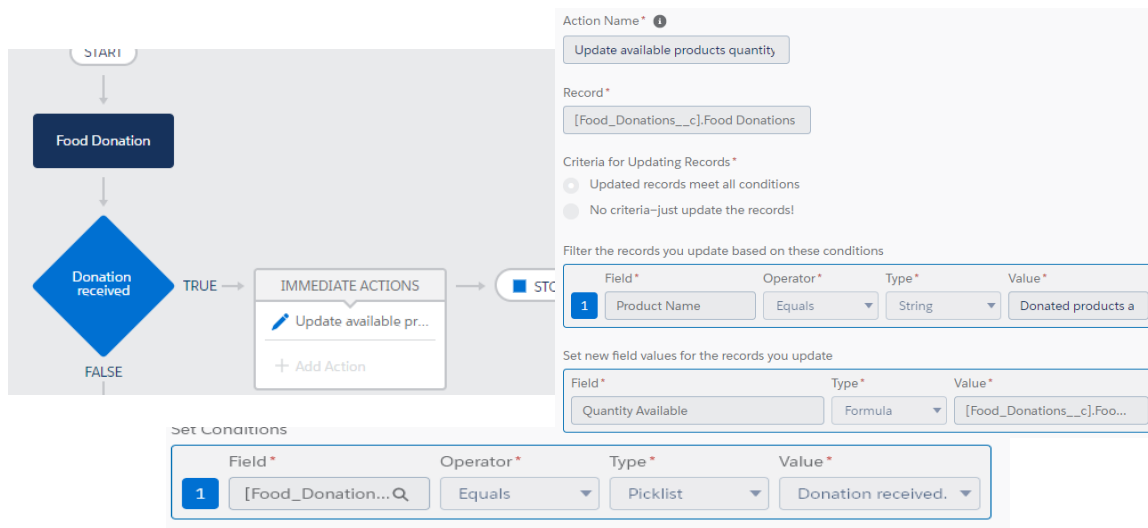


Figura 31-Process builder que atualiza a quantidade total de cabazes disponíveis

Também se criou um *process builder* que cria uma *task* sempre que é feita uma doação de alimentos. Esta *task* servirá para o administrador adicionar um voluntário ao registo da doação para proceder à recolha de alimentos para a organização.

6.13.2.3 Reports e dashboards

A figura seguinte mostra o gráfico presente na página inicial da aplicação que mostra a quantidade total de doações de alimentos recebidas, que também é atualizado automaticamente.



Figura 32-Gráfico das doações de cabazes

O *dashboard* da figura 33 está presente na página inicial e mostra a quantidade de cabazes e produtos disponíveis para serem doados a famílias e instituições, este é atualizado sempre que uma nova doação é recebida.

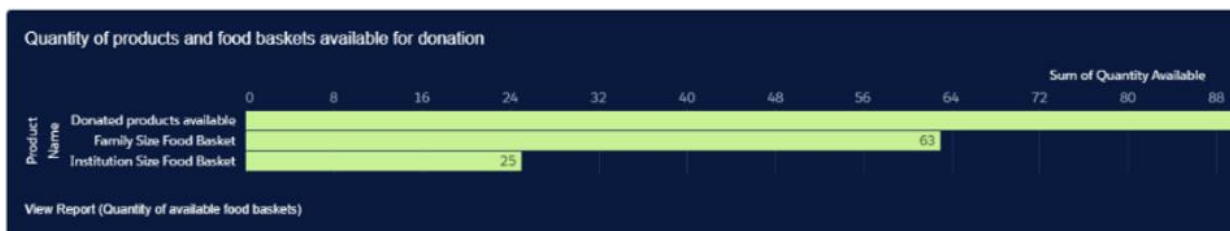


Figura 33-Dashboard dos cabazes disponíveis para doação

6.14 Objetos *Standard*

Os objetos *standard* ou padrão são os objetos que vêm incluídos no *Salesforce*. Objetos comerciais comuns, como Conta, Contato, *Lead* e Oportunidade, são todos objetos padrão, mas que podem ser modificados de modo a corresponder aos nossos objetivos.

6.14.1 *Accounts*

No *Salesforce*, cada contacto tem de estar associado a um tipo de conta, então neste objeto apenas vamos ter 5 registos.

	Account Name ↑
1	Food Donations
2	Food Requests
3	Money Donations
4	Partners
5	Volunteer

Figura 34-Record types do objeto account

6.14.1.1 *Validation rules*

Nas *Accounts* não foi necessário fazer nenhuma regra de validação específica porque não será necessário criar novas contas, uma vez que todos os contactos vão fazer parte das 5 contas demonstradas anteriormente.

Também não foi necessário criar *process builders* para este objeto.

6.14.2 Contacts

Este objeto é dos mais importantes, uma vez que vai ter os dados de todos os utilizadores da plataforma.

6.14.2.1 Record Types

Foram criados 7 *record types* para permitir uma melhor gestão de todos os processos que estão envolvidos na plataforma:

- *Family*- Contactos das famílias que recebem cabazes;
- *Donor*- Contactos dos doadores de dinheiro;
- *Institution*- Contactos das instituições que recebem cabazes;
- *Partner*- Contactos dos sócios;
- *Restaurant*- Contactos dos restaurantes que fazem doações de cabazes;
- *Supermarket*- Contactos dos supermercados que fazem doações de cabazes;
- *Volunteer*- Contacto dos voluntários que fazem entregas e vão buscar os cabazes das doações.

A figura 35 mostra a página de um registo criado no objeto *Contacts* e a figura 36 mostra os *record types* deste objeto.

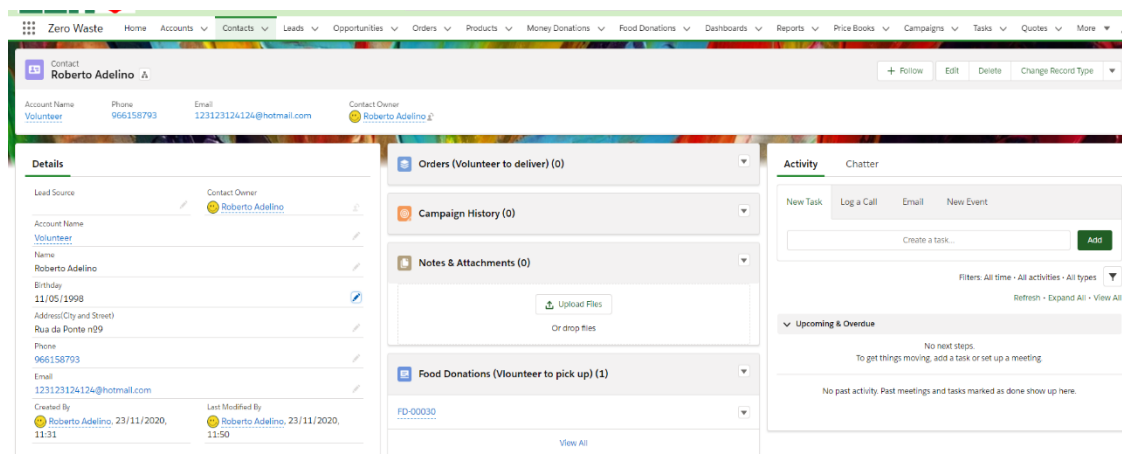


Figura 35-Página de um registo do objeto contacto

The screenshot shows a 'New Contact' form with a section titled 'Select a record type'. It contains seven radio button options: Family (selected), Donor, Institution, Partner, Restaurant, Supermarket, and Volunteer. At the bottom right, there are 'Cancel' and 'Next' buttons.

Figura 36-Record types criados

6.14.2.2 Page layouts

Como temos 7 record types diferentes, foi necessário criar 7 layouts diferentes, 1 para cada record type, uma vez que há campos que só faz sentido mostrar para esse tipo de registo. A figura 37 mostra a página de configuração dos layouts.

The screenshot displays the 'Page Layouts' configuration interface. On the left, a list of page layouts is shown, with 'Donor', 'Families', 'Institutions', 'Partner', 'Restaurant', 'Supermarket', and 'Volunteer' highlighted in a red box. The main area shows the configuration for the 'Contact Detail' layout. It includes a 'Fields' section with a search bar and a table of fields. Below this, there are sections for 'Additional Information', 'Contact Information', and 'Address Information', each with a 'Header visible on edit only' label. The 'Contact Information' section is expanded, showing fields like 'Account Name', 'Name', 'Number of people in the household', 'Mensual Revenue', 'Phone', 'Email', and 'Address(City and Street)'. The 'Address Information' section is also visible at the bottom.

Figura 37-Exemplo de um page layout dos contactos

6.14.2.3 Validation rules

Uma vez que é dos objetos mais importantes, existe uma maior quantidade de regras de validação para os campos. Existe também a regra para a validação do formato do IBAN igual á já existente no objeto das “Money donations”.

Foi criada uma regra que valida a data de nascimento, ou seja, apenas os maiores de 18 anos se podem registar como “Donors” ou “Volunteers” (Figura 38).

Validation Rule Detail	
Rule Name	datanascimento
Error Condition Formula	ADDMONTHS(Birthday__c ,216) > TODAY()
Error Message	Under 18.

Figura 38-Regra de validação da idade

A regra da figura 39 obriga a introdução de uma quantidade maior que zero no campo do número das pessoas do agregado familiar, apenas presente no record type “Family”.

Validation Rule Detail		Active
Rule Name	peessoasDoAgregadoMaiorQueZero	Active
Error Condition Formula	AND(RecordType.Name = 'Family', OR(Number_of_people_in_the_household__c < 0, ISNULL(Number_of_people_in_the_household__c)))	
Error Message	Enter a positive number.	Error Localized

Figura 39-Regra no campo do número de pessoas do agregado familiar

A figura 40 mostra uma regra semelhante à anterior, mas que se aplica ao record type “Institution” e no campo do número de pessoas que a instituição abrange.

Validation Rule Detail		Active
Rule Name	peessoasInstituicao	Active
Error Condition Formula	AND(RecordType.Name = 'Institution', OR(People_it_covers__c < 0, ISNULL(People_it_covers__c)))	
Error Message	Enter a positive number.	Error Localized
Description		

Figura 40- Regra no campo do número de pessoas que a instituição abrange

Regra que valida o campo de comprovativo do rendimento mensal das famílias para o record type ‘Family’ (Figura 41). Também existe outra regra semelhante, mas para os registos das instituições para validar o comprovativo do nº de pessoas que a instituição abrange.

Validation Rule Detail	
Rule Name	proofofrevenueottull
Error Condition Formula	AND(RecordType.Name == 'Family' , Proof_of_revenue__c == NULL)
Error Message	Insert comprovative
Description	
Created By	Roberto Adalberto 22/11/2020 10:47

Figura 41-Regra de validação do comprovativo nas famílias

Na figura 42 está a regra que valida se o contacto telefónico tem 9 dígitos e que não se pode deixar o campo em branco.

Error Condition Formula	OR((ISBLANK(Phone)), LEN(Phone) < 9)
Error Message	Please enter a 9 digit phone.

Figura 42-Regra de validação do campo do telefone

Na figura 43 está apresentada a regra que valida o formato do email introduzido.

Validation Rule Detail		Edit	Clone
Rule Name	validateEmail		
Error Condition Formula	NOT(REGEX(Email ,'{[a-zA-Z0-9_\\-\\.]+}@((\\[a-z]{1,3}\\.[a-z]{1,3}\\.[a-z]{1,3}) ((([a-zA-Z0-9\\-]+\\.)+))([a-zA-Z]{2,4}){0-3}'))		
Error Message	Insert a correct email address.		

Figura 43-Validação do email

6.14.2.4 Process builder

A figura 44 demonstra o *process builder* que envia um email cada vez que um novo contacto é registado na plataforma e o respetivo template de email criado.

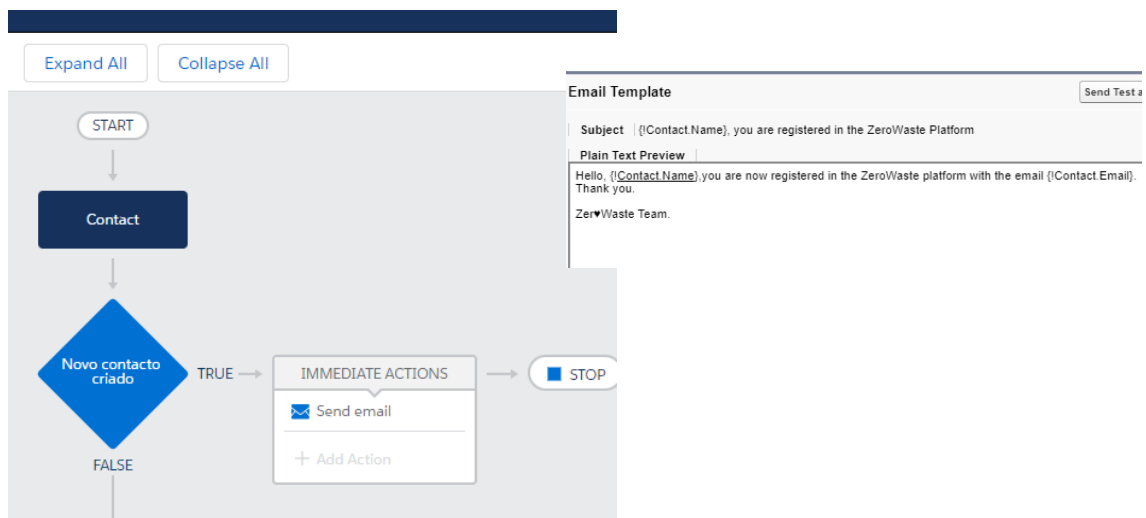


Figura 44-Process builder que envia um email ao novo contacto

6.14.2.5 Dashboards/Reports

A figura 45 mostra o *report* que foi criado na página inicial que serve para mostrar o número de contactos por cada tipo de *record type* registados na plataforma.

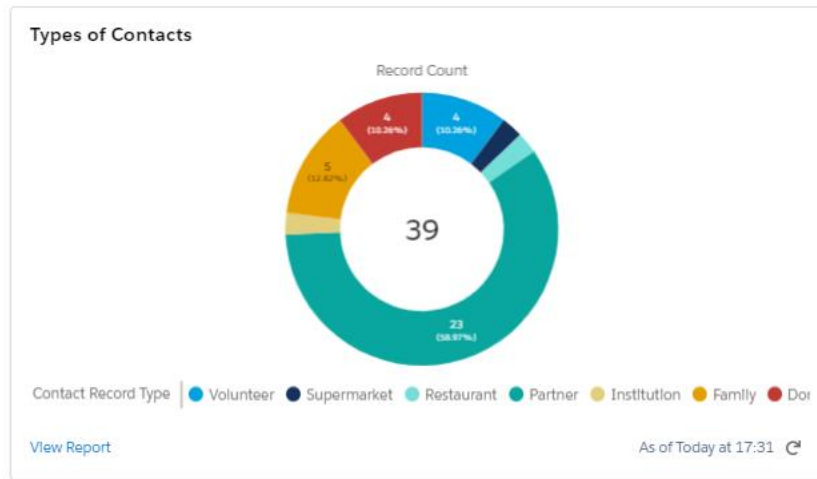


Figura 45-Report do tipo de contactos total

6.14.3 Leads

As leads podem vir do site ou podem ser adicionadas manualmente pelo administrador do sistema. A figura 46 mostra a página de um registo no objeto das “Leads” e tal como no objeto das doações também existe um *path* que permite uma melhor gestão dos registos através dos diferentes estados de cada registo.

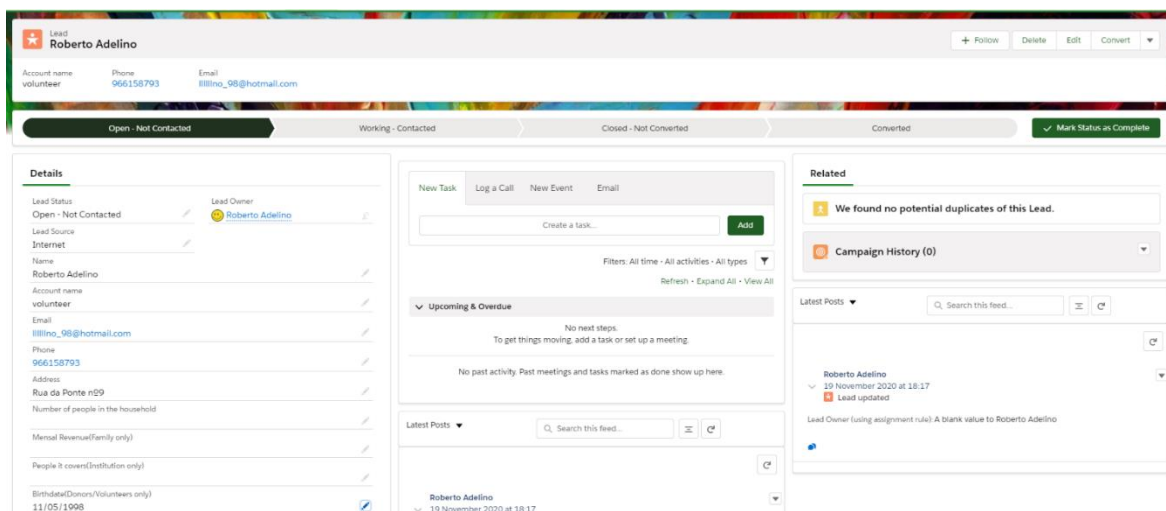


Figura 46-Página de um registo do objeto Lead

Como o objeto teve de ser modificado para os requisitos definidos, teve de ser feito o mapeamento dos campos adicionados para os campos dos contactos quando uma *lead* é convertida.

Para as ‘*Accounts*’ não é necessário, uma vez que não se vão criar novas contas, apenas se vão usar as que já estão criadas.

A figura 47 mostra os campos que vão ser preenchidos na tabela dos Contactos através dos campos do objeto das Leads.



ACCOUNT	CONTACT	OPPORTUNITY
 Lead Fields		 Contact Fields
Address		Address(City and Street)
Birthdate(Donors/Volunteers Only)		Birthday
Current Generator(S)		None
Mensual Revenue(Family Only)		Mensual Revenue
Number Of Locations		None
Number Of People In The Household		Number of people in the household
People It Covers(Institution Only)		People it covers
Primary		None
Product Interest		None

Figura 47-Mapeamento das leads

A figura 48 mostra a janela de conversão de uma *Lead* para um contacto.

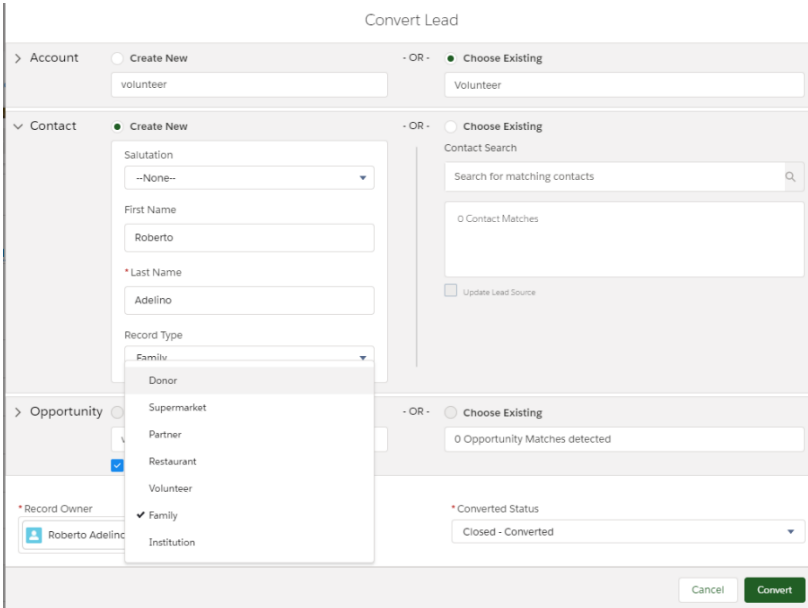


Figura 48-Conversão de uma lead

6.14.3.1 Validation rules

Não foi necessário criar regras de validação diferentes pois vai utilizar as mesmas regras dos campos do objeto *Contacts*.

6.14.3.2 Process builders

A figura seguinte mostra o process builder desenvolvido que envia uma notificação quando há uma *lead* nova proveniente do website.

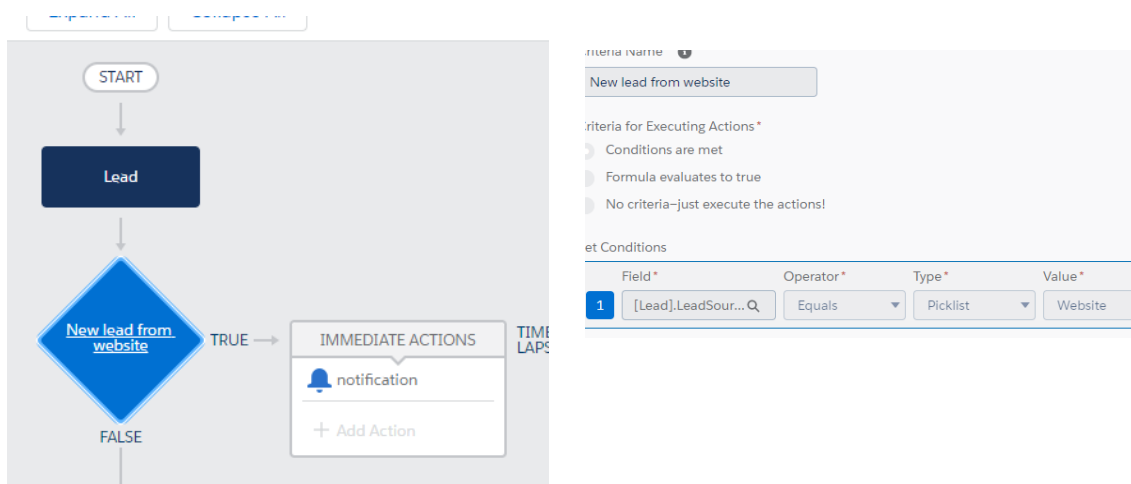


Figura 49-Process builder que envia uma notificação para nova lead

6.14.4 Products

Objeto onde são inseridos todos os produtos que são utilizados na plataforma quer sejam para venda ou para doação. É também onde está a quantidade de cada produto e o preço do mesmo. A figura 50 mostra a página dos produtos da plataforma, e as respetivas quantidades e preços.

The screenshot shows the Salesforce Products page. The table displays 15 items, sorted by Product Name. The columns are Product Name, Quantity Available, Price, and Product Description.

Product Name	Quantity Available	Price	Product Description
Donated products available	155	0.00	
Family Size Food Basket	65	0.00	Made with 10 products
Institution Size Food Basket	50	0.00	Made with 20 products
Monthly 10	9,999,999	10.00	
Monthly 20	9,999,999	20.00	
Monthly 30	9,999,999	30.00	
Monthly 50	9,999,999	50.00	
Weekly 10	9,999,999	10.00	
Weekly 15	9,999,999	15.00	
Weekly 5	9,999,999	5.00	
Yearly 100	9,999,999	100.00	
Yearly 150	9,999,999	150.00	
Yearly 200	9,999,999	200.00	
Yearly 250	9,999,999	250.00	
Yearly 50	9,999,999	50.00	

Figura 50-Todos os produtos da plataforma

6.14.4.1 Validation rules

Foi criada uma regra (figura 51) que não deixa que uma quantidade de um produto seja negativa.

Validation Rule Detail	
Rule Name	quantidade_negativa
Error Condition Formula	Quantity_Available__c < 0
Error Message	No more product available!
Description	
Created By	Roberto Adelino, 23/11/2020, 16:39

Figura 51-Regra de validação na quantidade

6.14.4.2 Process builder

A figura seguinte mostra o *process builder* que envia uma notificação ao administrador quando a quantidade de um produto chega a 0.

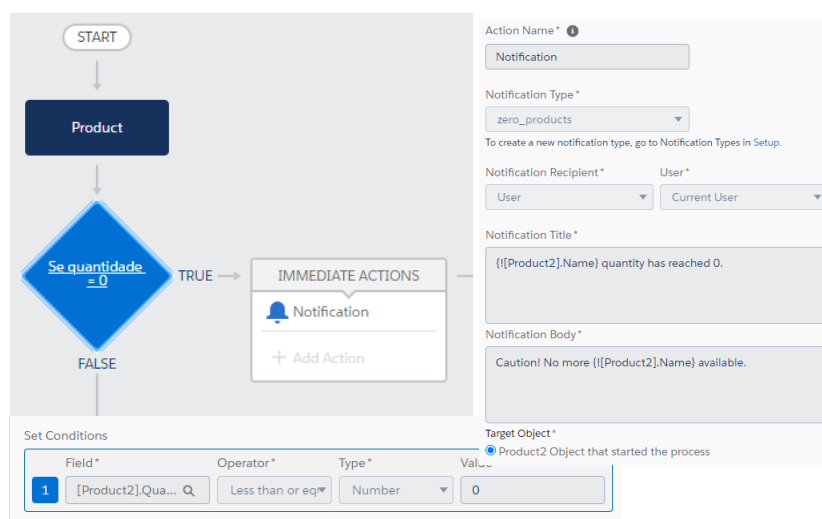


Figura 52-Process builder que envia uma notificação

6.14.5 Food requests and Sales (Opportunity)

Este objeto embora seja *Standard*, foi bastante modificado de modo a ir de encontro aos objetivos pretendidos. É provavelmente o objeto mais importante da aplicação pois a maioria das operações são baseadas à volta deste objeto. É o caso da venda dos planos de sócio e da atribuição dos cabazes a famílias e instituições.

Quando uma oportunidade passa para o estado “*Closed won*” e o record type é “*Food Request*” vai ser gerado um registo no objeto “*Food requests to deliver*” (*Order*) automaticamente para se proceder à entrega dos cabazes. Vai também ser criado um novo registo no objeto “*Food Requests and Sales*” (*Opportunity*), mas em que o campo *Close date* será 30 dias a mais do dia em que o pedido foi concluído, ou seja, cada família ou cada instituição só terá direito a uma entrega/pedido de cabazes por mês.

Quando se trata de uma venda de planos de sócio e o estado fica “*Closed won*” então será gerada uma nova oportunidade com mais 7 dias se for um plano semanal, 30 dias se for mensal e 365 se for um plano anual.

A figura 53 mostra a página de um registo de pedido de alimentos com o estado “*Closed won*” e pode-se verificar no lado direito a associação a um registo no objeto *Orders* (“*Requests to deliver*”).

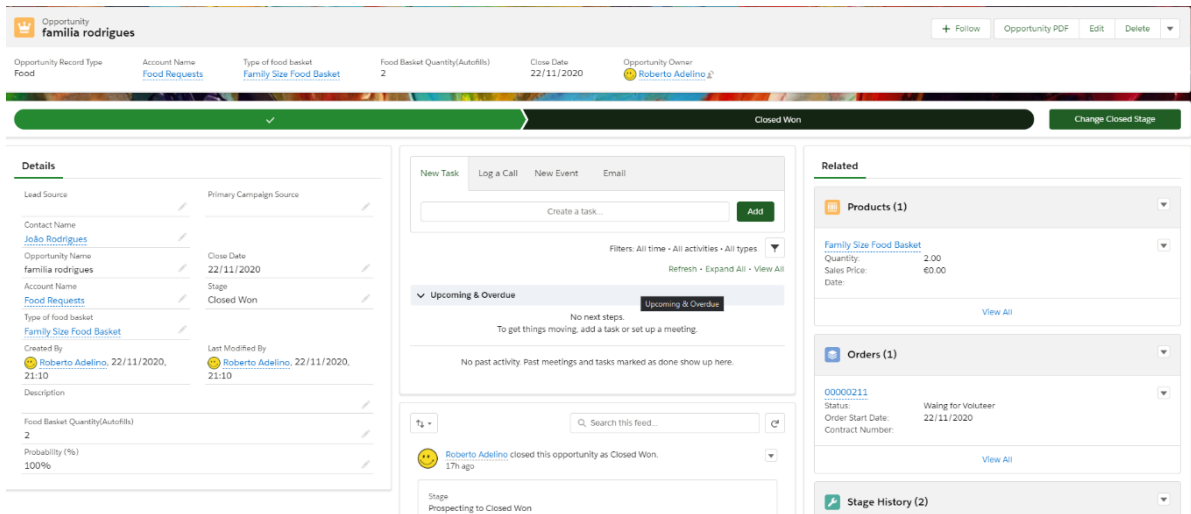


Figura 53-Página de um registo de pedido de alimentos

6.14.5.1 Record types

Uma vez que há a venda de planos de sócio e doações de cabazes a famílias e instituições foi necessário criar dois *record types*. Sendo um para os planos de sócio e outro para as doações de alimentos.

- *Food - Record type* para os pedidos de cabazes das famílias e instituições;
- *Partner plan - Record type* para a venda de planos de sócio;

New Request and Sale

Select a record type

Food
 Partner Plan

Figura 54-Record types do objeto Food requests and sales (Opportunity)

6.14.5.2 Page layouts

Como há dois record types, foram também criados dois page layouts de modo a apenas mostrar os campos de cada *record type* no seu respetivo layout.

6.14.5.3 Filtros nos campos

A figura 55 mostra o filtro no campo da *Account* para apenas mostrar as contas do tipo “*Food requests*” quando se trata de um *record type* “*Food*” e para mostrar a conta “*Partners*” quando o *record type* for “*Partner plan*”.

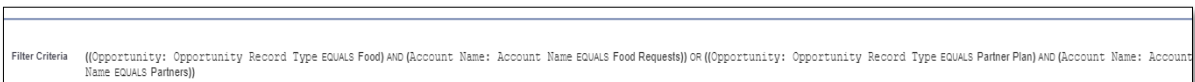


Figura 55-Filtro no campo das *Accounts*

Na figura 56 está o filtro para apenas mostrar os Planos de sócio no campo “*Partner plan*”.



Figura 56-Filtro no campo *Partner plan*

A figura seguinte mostra o filtro para apenas mostrar os contactos com *record type*

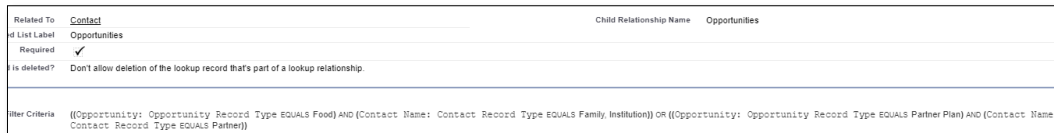


Figura 57-Filtro no campo dos contactos

“*Family*” e “*Institution*” quando se trata de uma novo registo no objeto *Food Requests and Sales* de *record type* “*Food*” ou para apenas mostrar os contactos de *record type* “*Partner*” nas oportunidades “*Partner plan*”

A figura 58 apresenta o filtro para apenas mostrar os cabazes num registo de *record type* “*Food*”



Figura 58-Filtro nos campos dos cabazes

6.14.5.4 Validation rules

A figura 59 mostra a regra desenvolvida para validar os tipos de cabazes para cada tipo de contacto. Ou seja, a uma família estará associado o cabaz das famílias e a uma instituição estará associado o cabaz das instituições.

Validation Rule Detail		Edit	Clone
Rule Name	family_food_basket		
Error Condition Formula	AND(Contact_Name__r.RecordType.Name = 'Family' , Type_of_food_basket__r.Name = 'Institution Size Food Basket')		
Error Message	Wrong food basket		

Validation Rule Detail		Edit	Clone
Rule Name	institiofoodbaskettype		
Error Condition Formula	AND(Contact_Name__r.RecordType.Name = 'Institution' , Type_of_food_basket__r.Name = 'Family Size Food Basket')		
Error Message	Wrong food basket.		

Figura 59-Regra no campo dos cabazes

Também foi criada uma regra semelhante às já apresentadas anteriormente que obriga a que a data do registo seja a do próprio dia.

6.14.5.5 Flows

A figura seguinte mostra o *flow* que vai gerar um novo registo no objeto “*Requests to deliver (orders)*” quando um pedido do objeto “*Food requests and Sales*” é “*Closed won*” (ou seja, concluído) e do record type ‘*Food*’

Este flow foi adaptado de um *package* [23] da *AppExchange* do *Salesforce* mas foi modificado de modo a cumprir os requisitos e objetivos da aplicação.

O *flow* começa por ir buscar os dados do registo no objeto das “*Food requests and sales*”(Oportunidades) e verifica se já existe algum registo no objeto das “*Requests to deliver*”(Orders) para esses dados, se não houver então vai buscar os dados do contacto (Morada, email, etc.) ao registo dos contactos e contas associados a essa oportunidade para poder preencher os campos no objeto das *Orders*. Um registo no objeto das *orders* tem de ter um produto associado, então o *flow* vai buscar o produto associado no registo da oportunidade e coloca-o na *order*, no final o registo é criado e o estado da order é atualizado.

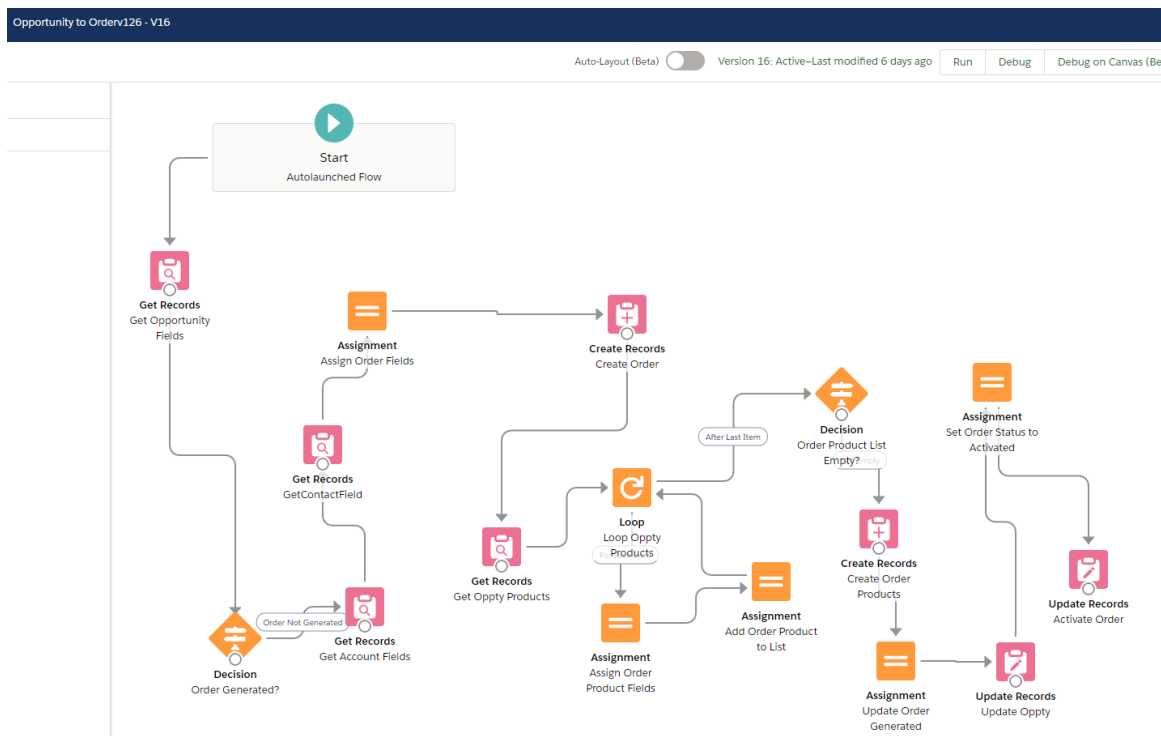


Figura 60-Flow que gera uma registo no objeto Requests to deliver

6.14.5.6 Process builder

A maior parte da automação encontra-se neste objeto.

O *flow* da figura anterior poderia ser substituído por um *process builder* que faria o mesmo, apresentado na figura seguinte, no entanto ao utilizar o *process builder* não seria possível associar um produto a uma *Order*.

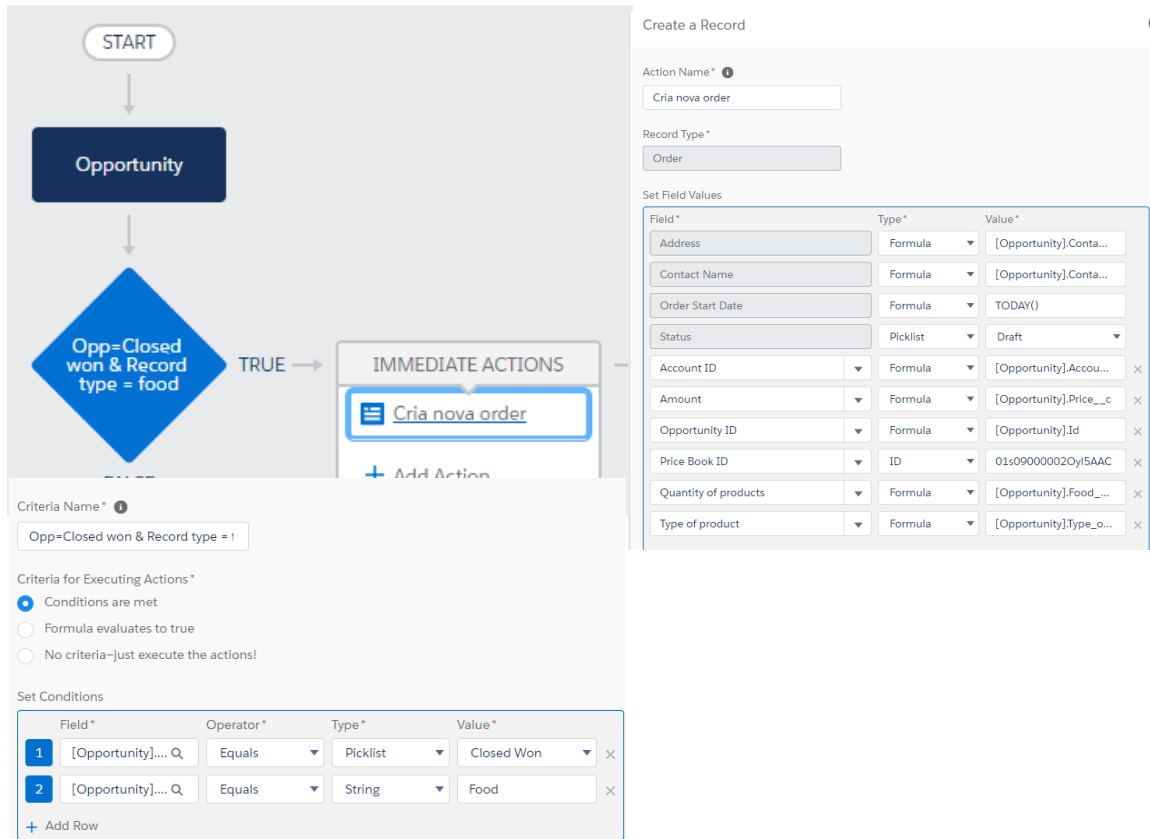


Figura 61-Process builder semelhante ao flow anterior

O seguinte processo atualiza automaticamente o campo “Price” dos registos do objeto “Food Requests and sales” de acordo com o preço de cada plano de sócio do objeto “Products”, o que permite que se o preço de um plano de sócio for alterado, então será atualizado automaticamente em cada registo deste objeto, caso contrário teria de ser fazer tudo manualmente.

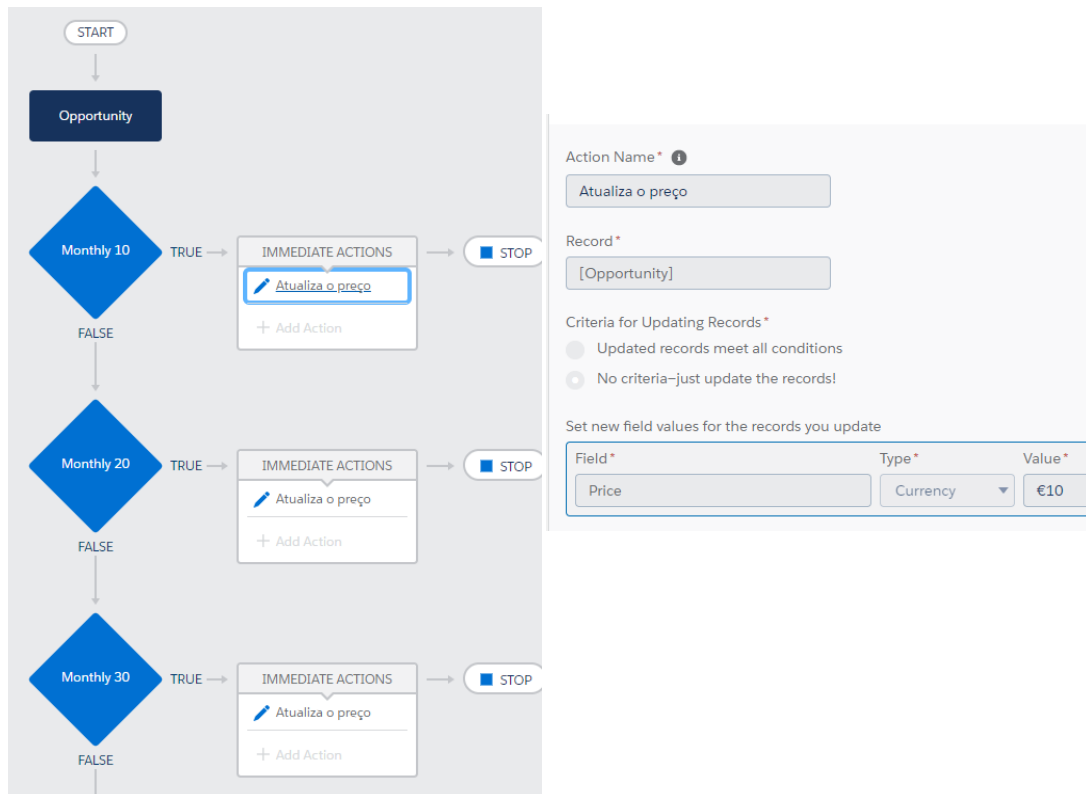


Figura 62-Process builder que atualiza o preço do plano de sócio

O processo apresentado na figura 63, vai atualizar automaticamente o campo da quantidade de Cabazes a que cada família ou instituição tem direito de acordo com os seus dados e de acordo com as regras definidas.

Tabela 11-Regras de atribuição de cabazes a famílias e instituições

Tipo de contacto	Regras	Quantidade de cabazes a atribuir	
Família	0<mensal_revenue ≤300 0<number_people_household ≤2	1	
	300<mensal_revenue <500 2≤number_people_household ≤3	2	
	300<mensal_revenue <500 4≤number_people_household ≤100	3	
	500≤mensal_revenue ≤700 3≤number_people_household <5	3	
	500≤mensal_revenue <750 5≤number_people_household ≤100	4	
	750≤mensal_revenue <1000 6≤number_people_household <100	5	
	Instituição	0<people_it_covers<30	3
		30≤people_it_covers<50	5
		50≤people_it_covers<100	8
		people_it_covers≤100	10

Ou seja:

A figura 63 apresenta o processo que atualiza automaticamente o campo da quantidade dos cabazes da oportunidade de acordo com a tabela anterior, este processo vai também retirar esta quantidade à quantidade total do stock de cabazes disponíveis da tabela dos produtos.

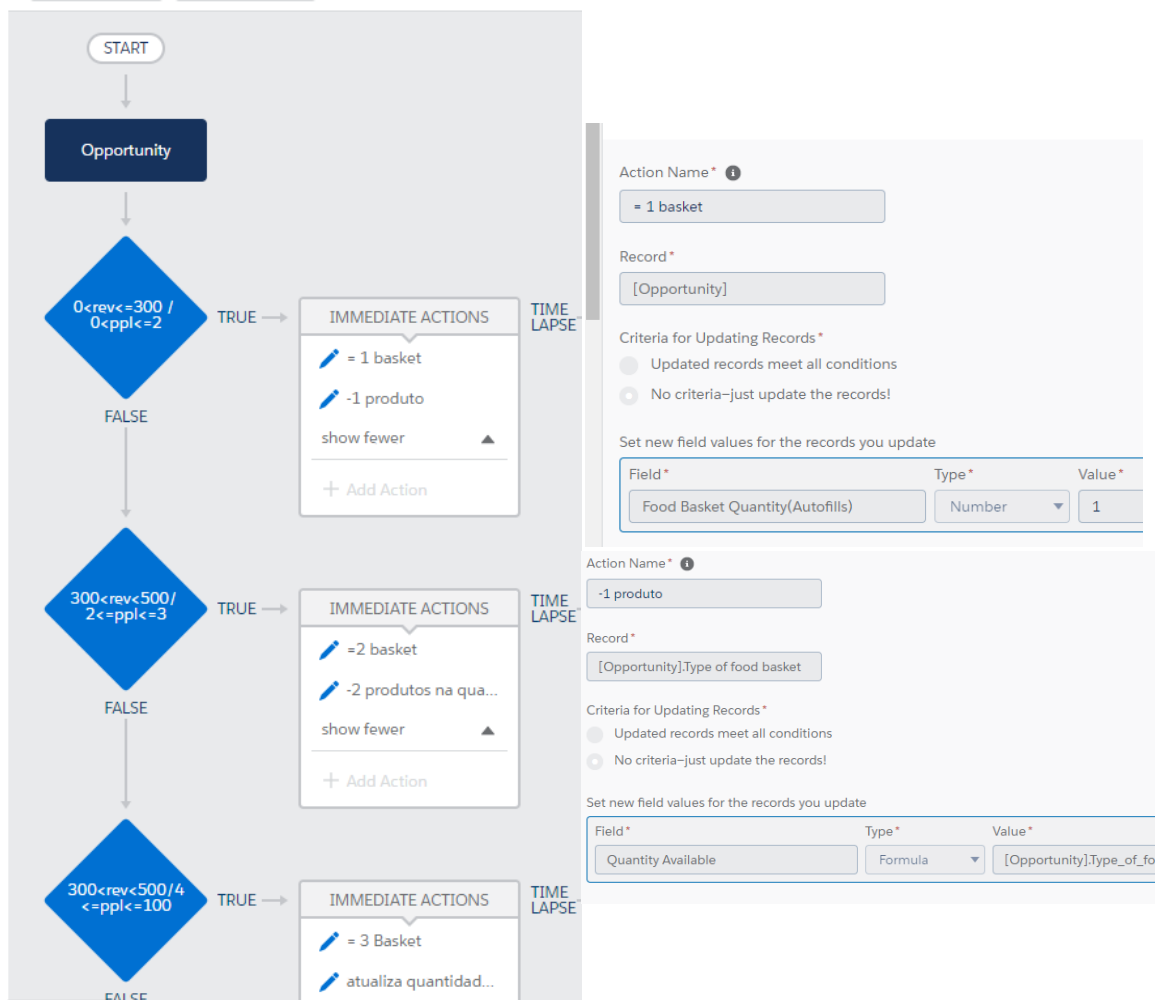


Figura 63-Process builder que atribui cabazes de acordo com as regras e requisitos

A figura 64 descreve o processo que envia uma notificação de erro quando um contacto não se enquadra nas regras de atribuição de cabazes da tabela anterior.

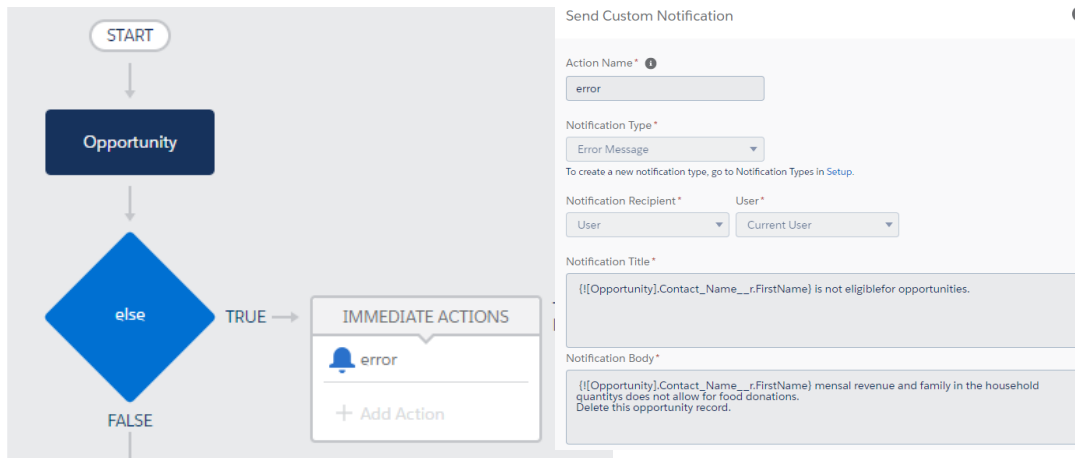


Figura 64-Process builder que envia notificação de erro

A figura 65 demonstra o processo que chama o flow apresentado na figura 60 para gerar um novo registo no objeto *Requests to deliver (orders)* quando o *record type* da do objeto dos pedidos é 'Food' e o estado está concluído.

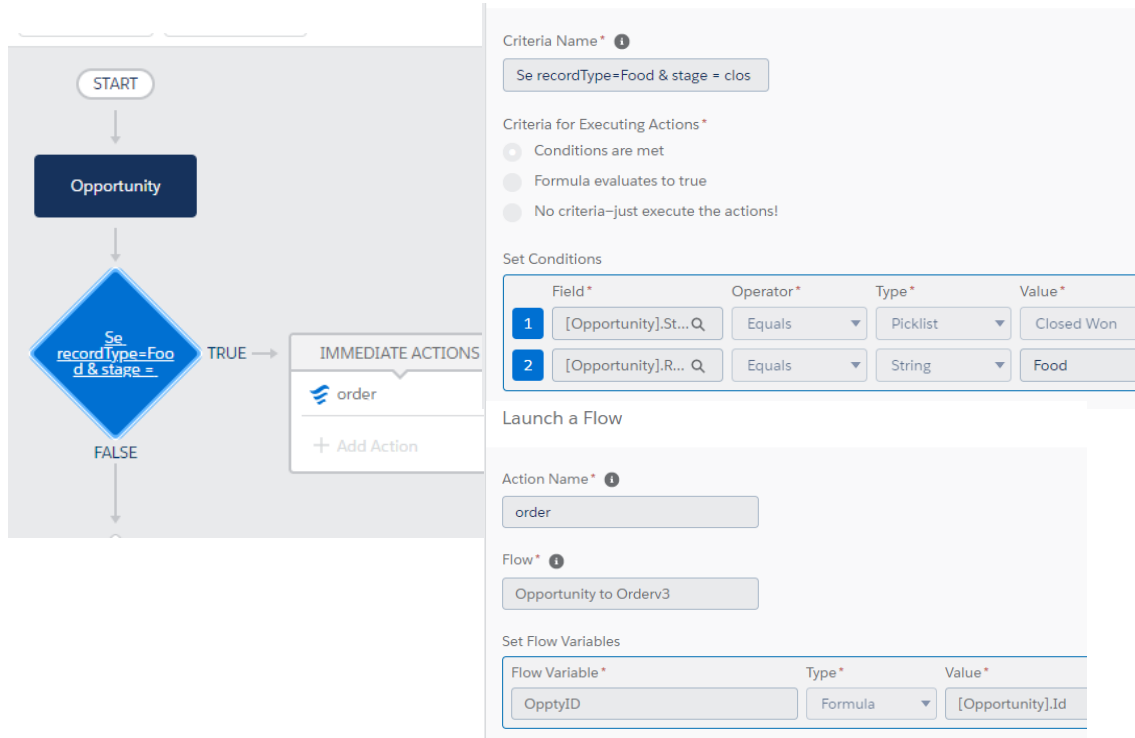


Figura 65-Processo que chama o flow da figura 60

6.14.5.7 Reports/Dasboards

A figura 66 mostra o report que foi criado na página inicial que mostra o tipo de planos de sócio vendidos e o total de planos vendidos.



Figura 66-Report que mostra o tipo de Plano de sócio vendidos

6.14.5.8 Custom actions

Foi criada uma *quick action* adaptada da AppExchange [24] do *Salesforce* que gera uma fatura em formato PDF de qualquer registo do objeto “*Food Requests and sales*”.

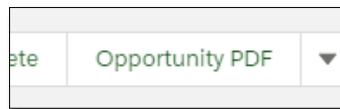


Figura 67-Botão para gerar PDF



ZEROWASTE
INVOICE

Prepared for: Roberto
Adelino

ZeroWaste Type: Partners

Address Information

Name: Roberto Adelino

Address:
Rua da Ponte nº9

Ship To:
Rua da Ponte nº9

Amount: € 250.00

Products

Product	Description	Quantity	Unit Price	Total Price
Weekly 15		1.0	€15.00	€15.00
				Total: €15.00

Delivered by

Signature:

Start Date:
Close Date: 22/11/2020

Payment :
PT12312312312331231231

Receiver signature

Signature:

Date:

Copyright ZeroWaste

Figura 68-Exemplo de fatura gerada

No Anexo A 3.1. apresenta-se o código construído/gerado.

6.14.5.9 Triggers

Uma vez que cada família e instituição apenas tem direito a um pedido de alimentos em cada mês, foi desenvolvido um *trigger* que renova automaticamente um pedido concluído que terá a data para o mês seguinte. Assim não vai ser necessário voltar a criar um registo para a família ou instituição em questão.

O código apresenta-se no Anexo A 3.2

Opportunity Record T...	Opportunity Name	Contact Name	Amount	Close Date	
1	Food	familia rodrigues	João Rodrigues	€0.00	22/11/2020
2	Food	familia rodriguesFood request -Auto Renewal	João Rodrigues	€0.00	22/12/2020

Figura 69-Novo pedido gerado automaticamente

Trigger semelhante ao anterior, mas que se aplica nas vendas dos planos de sócio em que a data vai variar de acordo com o tipo de plano comprado, ou seja, se for comprado um plano de sócio semanal então esse registo será renovado todas as semanas.

3	Partner Plan	Weekly partner plan opp	Roberto Adelino	€15.00	22/11/2020
4	Partner Plan	Weekly partner plan oppWeekly-Renewal	Roberto Adelino	€15.00	29/11/2020

Figura 70-Renovação de sócio automática

O código encontra-se no Anexo A 3.3.

6.14.6 Food Requests to deliver (Order)

Os registos neste objeto são gerados automaticamente quando o estado de um registo do objeto dos pedidos de alimentos passa a concluído e apenas para o *record type* 'Food'. Para poder atualizar o estado de um registo neste objeto, é necessário preencher o campo do voluntário que faz a entrega (Figura71).

Figura 71-Página de uma registo

6.14.6.1 Filtros nos campos

Apenas há validação no campo do voluntário, uma vez que é introduzido manualmente, enquanto que os restantes campos são introduzidos automaticamente.

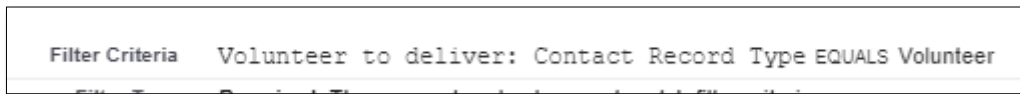


Figura 72-Filtro no campo dos Voluntários

6.14.6.2 Validation rules

Regra de validação que não permite alterar o estado da entrega enquanto não for adicionado um voluntário para fazer a entrega.

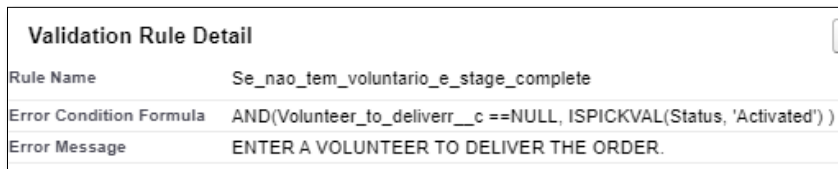


Figura 73-Regra de validação no estado da Order

6.14.6.3 Process Builders

A figura 74 mostra o processo que vai criar uma *task* para se adicionar um voluntário para fazer a entrega do pedido. Vai também enviar um email com o *template* criado ao voluntário que foi selecionado para fazer a entrega e também envia um email com outro *template* à família ou instituição associada ao pedido para avisar que a o voluntário vai proceder à entrega foi enviada.

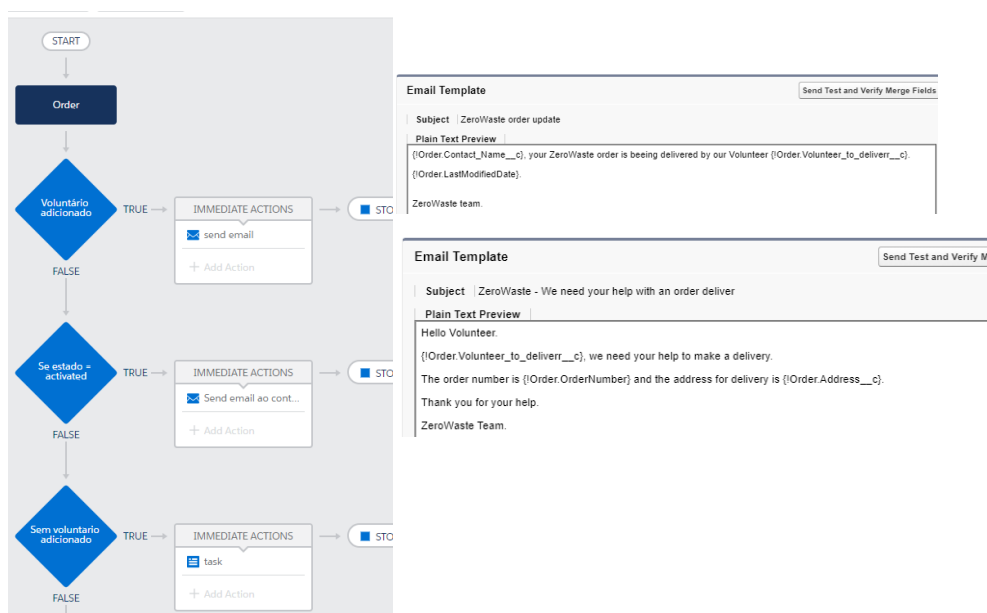


Figura 74-Process builder que cria uma task e envia emails

O *process builder* da figura 75, envia uma notificação sempre que é criado um novo registo no objeto *Requests to deliver*:

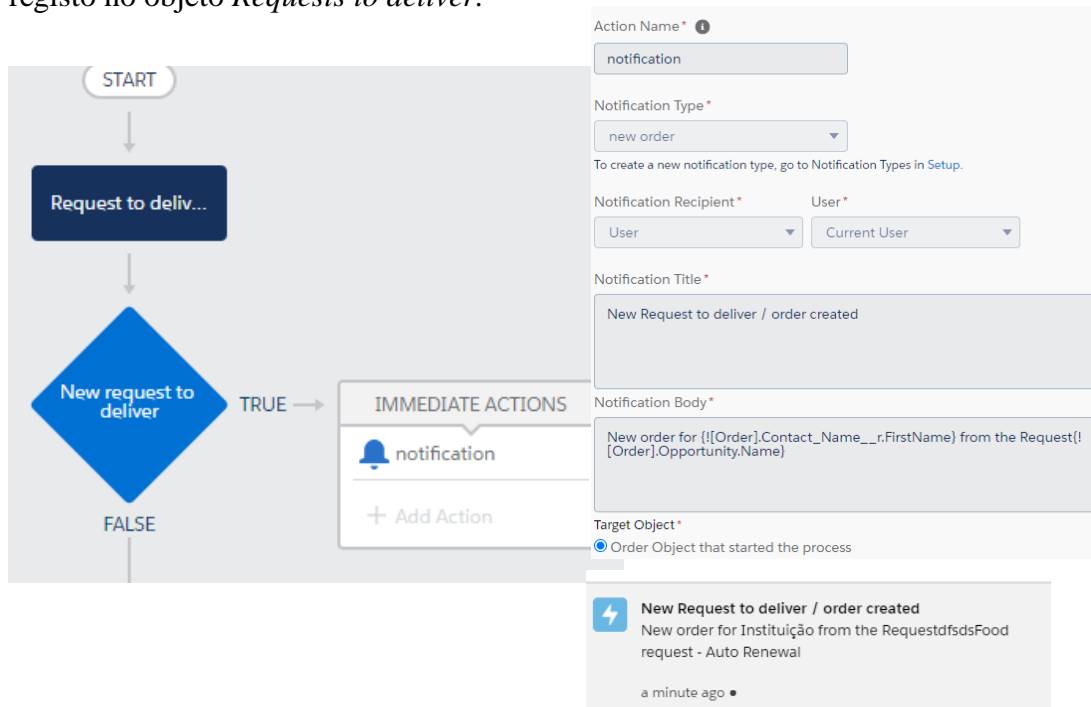


Figura 75-Process builder que envia uma notificação

6.14.7 Campaigns

Neste objeto encontra-se o registo de todas as campanhas realizadas, quer sejam para angariar fundos ou dar a conhecer a ONG.

Não foram criados automatismos para este objeto, apenas se modificou o layout de modo a respeitar os objetivos.

Campaigns

All Active Campaigns ⌵ +

Sorted by Campaign Name • Filtered by All campaigns - Active • Updated a few seconds ago

Search this list... ⌵

<input type="checkbox"/> Campaign Name ⌵	<input type="checkbox"/> Start Date ⌵	<input type="checkbox"/> End Date ⌵	<input type="checkbox"/> Budgeted Cost In Campaign... ⌵	<input type="checkbox"/> Actual Cost In Campaign ⌵	<input type="checkbox"/> Type ⌵	<input type="checkbox"/> Status ⌵
<input type="checkbox"/> Outdoor Ads	16/11/2020	27/11/2020		€300	Banner Ads	In Progress
<input type="checkbox"/> Social networks	21/11/2020	01/02/2021	€0	€0	Advertisement	In Progress
<input type="checkbox"/> Website	21/11/2020	27/11/2027			Advertisement	In Progress

New Campaign

Campaign Information

Campaign Owner
Roberto Adelino

* Campaign Name

Active

* Type
Conference ⌵

* Status
Planned ⌵

* Start Date
 📅

* End Date
 📅

Expected Revenue In Campaign

Budgeted Cost In Campaign

Actual Cost In Campaign

Expected Response (%)
0.00%

Figura 76-Página das campanhas e formulário de criação de nova campanha

6.14.8 Tasks

Neste objeto estão todos os registos das tarefas, provenientes dos outros objetos, como por exemplo, fazer chamadas, enviar emails, adicionar voluntários nas entregas de pedidos, entre outras. A figura 77 mostra todos os registos deste objeto.

Subject	Name	Related To	Due Date ↑	Status	Priority	Assigned Alias	Last Modified Date/Time
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 17:42
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 17:42
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 17:42
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 18:56
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 18:56
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 18:56
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:15
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:15
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:15
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:35
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:35
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:35
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:40
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:40
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:40
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:43
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:43
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:43
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:47
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:47
Add a volunteer to deliver order			22/11/2020	In Progress	High	RAdeI	21/11/2020, 20:47

Figura 77-Página das tarefas

6.15 Implementação do website com recurso a Visualforce pages

A página web servirá não só para dar a conhecer a missão da ONG, mas também para angariar doações de alimentos, recrutar voluntários, vender planos de sócio, entre outras funcionalidades. A página pode ser acedida no link: <https://zerowaste-developer-edition.eu40.force.com/zerowaste>.

6.15.1 Página inicial do website

Criou-se um website baseado em *Visualforce pages* que permite dar a conhecer a missão da ONG, receber doações e recrutar voluntários. Código no Anexo A 3.4. As figuras 78 e 79 mostram a página inicial do *website*.

Type	Rule	Food basket quantity
Family	Family mensal revenue less than or equal to 300€ and nº of people in the household less than or equal to 2	1
	Family mensal revenue greater than 300 and less than 500 and nº of people in the household less than or equal to 3 and greater than or equal to 2	2
	Family mensal revenue greater than 300 and less than 500 and nº of people in the household greater than or equal to 4	3
	Family mensal revenue greater than or equal to 500 and less than or equal to 700 and nº of people in the household less than 5 and greater than or equal to 3	3
	Family mensal revenue greater than or equal to 500 and less than 750 and nº of people in the household greater than 5	4
	Family mensal revenue greater than or equal to 750 and less than 1000 and nº of people in the household greater than 6	5
Institution	Nº of people helped by the institution less than 30	3
	Nº of people helped by the institution greater than or equal to 30 and less than 50	5
	Nº of people helped by the institution greater than or equal to 50 and less than 100	8
	Nº of people helped by the institution greater than or equal to 100	10

If you are a family or institution and meet this rules please contact us

[Contact us](#)

Figura 78-Página inicial do site parte 1

Food baskets we donate

Family size food basket	Institution size food basket
10 Products in the basket	20 Products in the basket
Donate Food	Donate Food

Become a partner
We have many partner plans

Weekly plans

- [Become a partner](#)

Weekly	Weekly	Weekly
€ 5 <small>per week</small>	€ 10 <small>per week</small>	€ 15 <small>per week</small>

Monthly plans

- [Become a partner](#)

Monthly	Monthly	Monthly
€ 10 <small>per month</small>	€ 20 <small>per month</small>	€ 30 <small>per month</small>

Monthly		
€ 50 <small>per month</small>		

Yearly plans

- [Become a partner](#)


Yearly	Yearly	Yearly
€ 50 <small>per year</small>	€ 100 <small>per year</small>	€ 150 <small>per year</small>

Yearly	Yearly	
€ 200 <small>per year</small>	€ 250 <small>per year</small>	

ZEROWASTE

Salesforce project developed for:

2020



IPG and ACADEMIA

*This is a fictional project developed to obtain a bachelor's degree in computer engineering

[f](#)
[@](#)
[o](#)
[p](#)
[t](#)
[in](#)

ZeroWaste 2020

Figura 79-Página inicial do site parte 2

6.15.2 Página de doação de alimentos e de doação de dinheiro

A figura 80 mostra a página desenvolvida que permite receber doações de alimentos. Código da página no Anexo A 3.5 e do controlador no Anexo A 3.6.



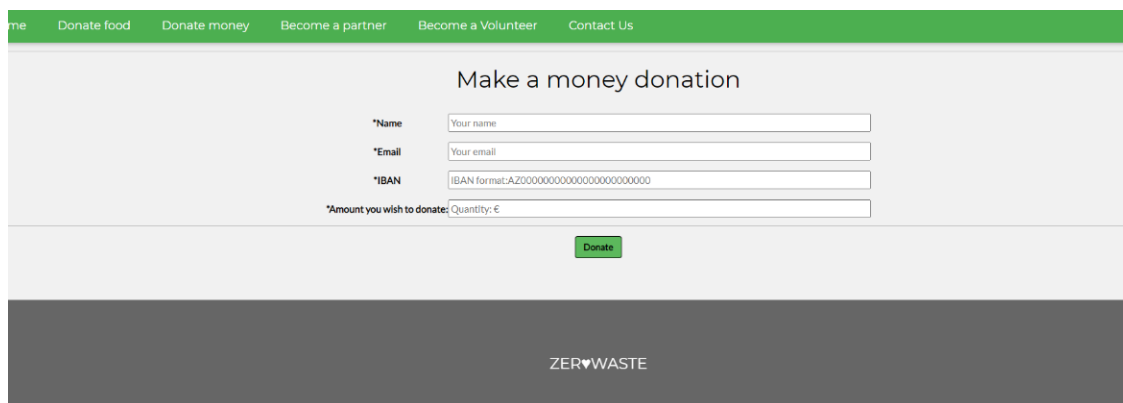
The screenshot shows a web form titled "Make a Food donation". At the top, there is a green navigation bar with links: Home, Donate food, Donate money, Become a partner, Become a Volunteer, and Contact Us. The form itself is on a light gray background and contains the following fields:

- *Name: Your name
- *Phone: 9 digit phone number
- *Address (city and street): Your address
- *Quantity of products to donate: Positive quantity

Below the fields is a green "Donate" button. At the bottom of the page, there is a dark gray footer with the text "ZERWASTE".

Figura 80-Página de doação de alimento

A figura 81 mostra a página desenvolvida que permite receber doações de dinheiro. Código da página no Anexo A 3.7 e do controlador no Anexo A 3.8.



The screenshot shows a web form titled "Make a money donation". At the top, there is a green navigation bar with links: Home, Donate food, Donate money, Become a partner, Become a Volunteer, and Contact Us. The form itself is on a light gray background and contains the following fields:

- *Name: Your name
- *Email: Your email
- *IBAN: IBAN format:AZ0000000000000000000000
- *Amount you wish to donate: Quantity: €

Below the fields is a green "Donate" button. At the bottom of the page, there is a dark gray footer with the text "ZERWASTE".

Figura 81-Página de doação de dinheiro

6.15.3 Página de venda de planos de sócio

A figura 82 mostra a página que permite vender planos de sócio.

Código do controlador no Anexo A 3.9

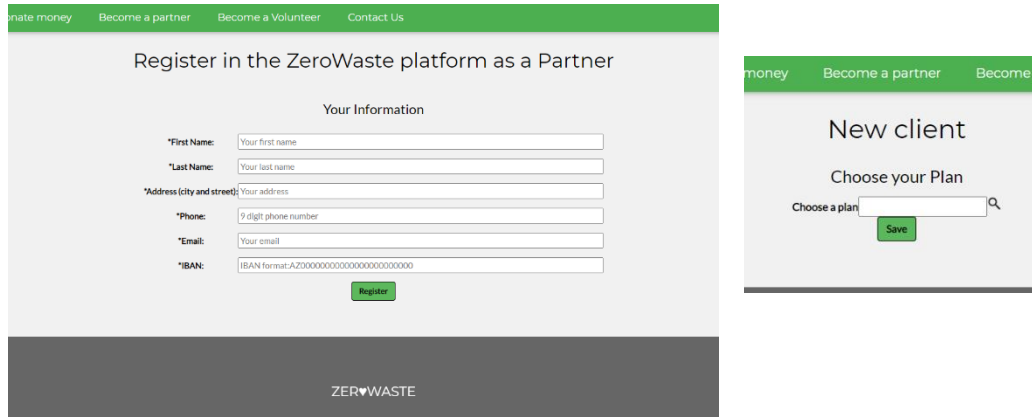


Figura 82-Páginas de compra de plano de sócio

6.15.4 Página de recruta de voluntários

A figura 83 mostra a página que permite o recrutamento de novos voluntários, código no Anexo A 3.10 .



Figura 83-Página de registo de novo voluntário

6.15.5 Página de contacto

Quando alguém precisa de entrar em contacto com a ONG, pode preencher o formulário da figura 84.

Código no anexo A 3.11.

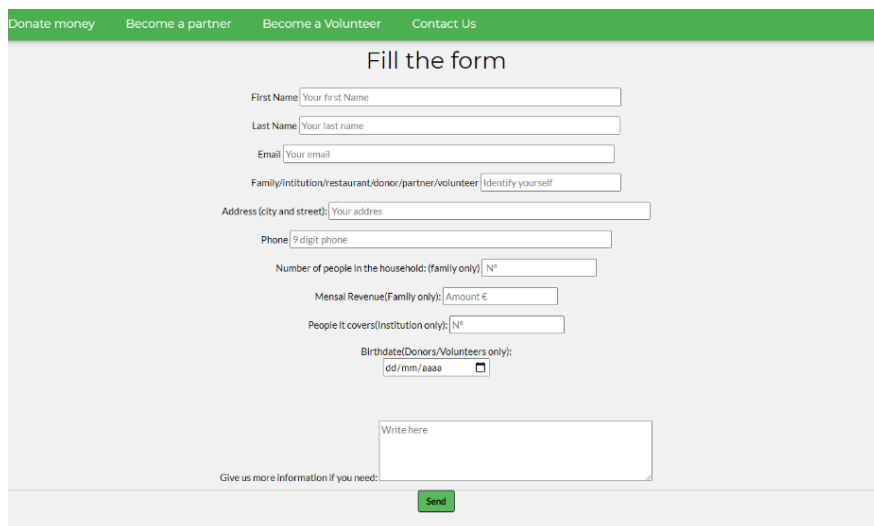


Figura 84-Página de contacto

6.16 Página de agradecimento

Após os doadores efetuarem uma doação serão redirecionados para a página de agradecimento (Figura 85).



Figura 85-Página de agradecimento

7 Verificação e validação

Ao longo de todo o projeto foram desenvolvidos vários tipos de testes visto que se optou pela metodologia ágil *SCRUM*, tal como foi explicado anteriormente. Muitos desses testes foram úteis na criação de todos os automatismos e na escolha das melhores ferramentas para os fazer, o que ajudou a resolver os erros de forma rápida. Nas figuras seguintes são apresentados alguns exemplos de testes.

A figura 86 mostra as mensagens de erro de quando se tenta criar um registo com campos por preencher.

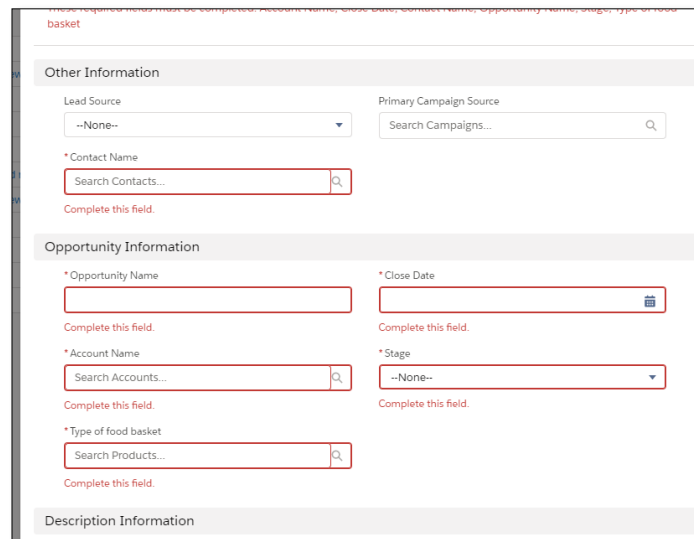


Figura 86-Testes de validação

A figura 87 mostra o teste efetuado ao campo dos cabazes, nos pedidos de uma família, em que apenas é possível selecionar um cabaz de família.

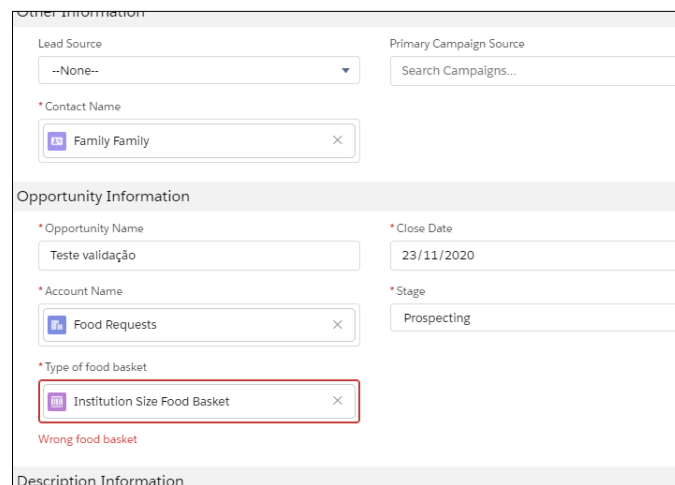
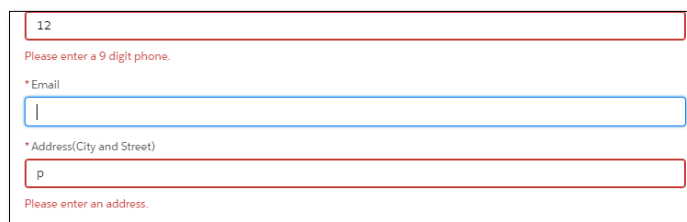


Figura 87-Teste de regras

A figura 88 mostra o teste efetuado à validação do formato do campo do telefone.



12
Please enter a 9 digit phone.

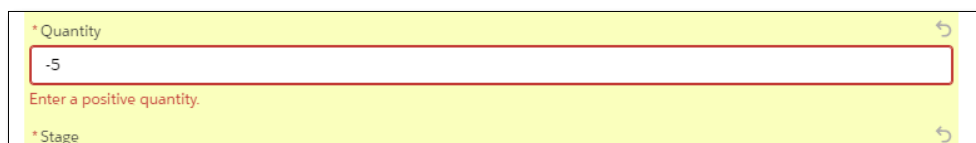
* Email

* Address(City and Street)

p
Please enter an address.

Figura 88-Testes de validação do formato de dados

A figura 89 mostra a mensagem de erro de quando se tenta doar uma quantidade negativa de alimentos.



* Quantity

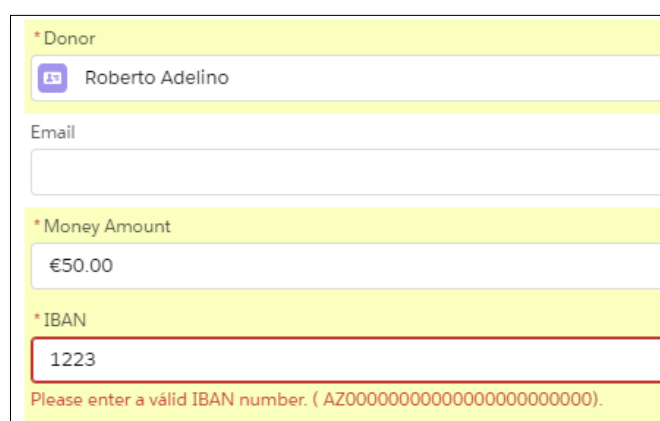
-5

Enter a positive quantity.

* Stage

Figura 89-Testes de validação do campo quantidade

A figura 90 demonstra a validação de dados no campo IBAN e a respetiva mensagem de erro.



* Donor

Roberto Adelino

Email

* Money Amount

€50.00

* IBAN

1223

Please enter a valid IBAN number. (AZ00000000000000000000000000000000).

Figura 90-Validação do formato do IBAN

A figura 91 mostra o erro de quando o estado de uma entrega é alterado sem ter o campo do voluntário preenchido, ou seja, para que o estado de uma entrega possa ser atualizado, tem de se obrigatoriamente adicionar um voluntário no campo.

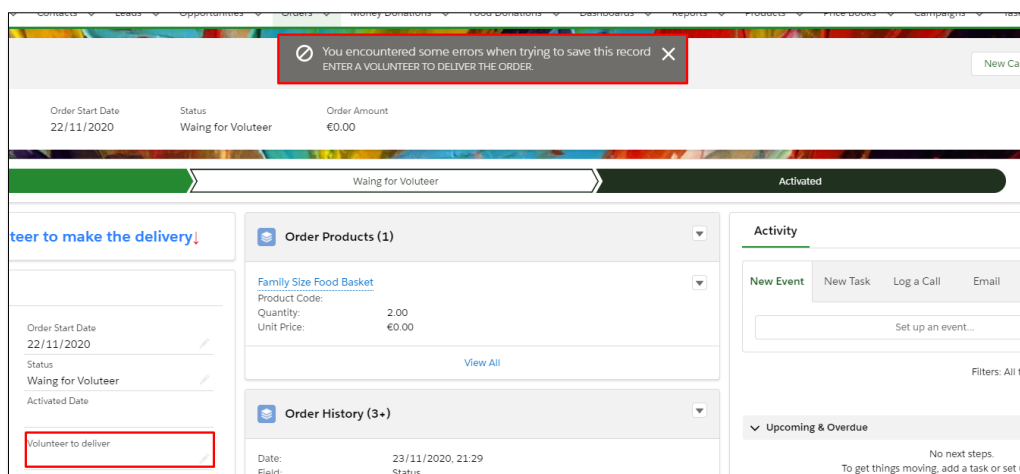


Figura 91-Teste de atualização do estado de uma entrega

A figura 92 mostra uma tarefa gerada automaticamente quando é criado um novo registo no objeto das *Orders*.



Figura 92- Tarefa gerada automaticamente

A figura 93 apresenta o teste feito ao envio de um email quando um novo contacto é registado na plataforma

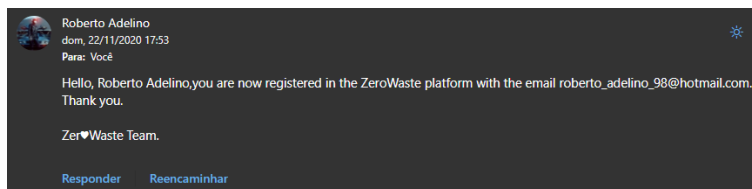


Figura 93-Email enviado ao novo contacto registado na plataforma

Quando um utilizador faz uma doação irá receber um email semelhante ao da figura 94 e o administrador do sistema vai receber uma notificação (Figura 95).



Figura 94-Email enviado ao utilizador que fez uma doação

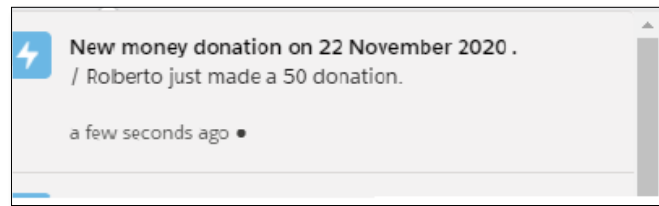


Figura 95-Notificação recebida pelo administrador

A figura 96 mostra a notificação que o administrador recebe quando alguém faz uma doação de alimentos.

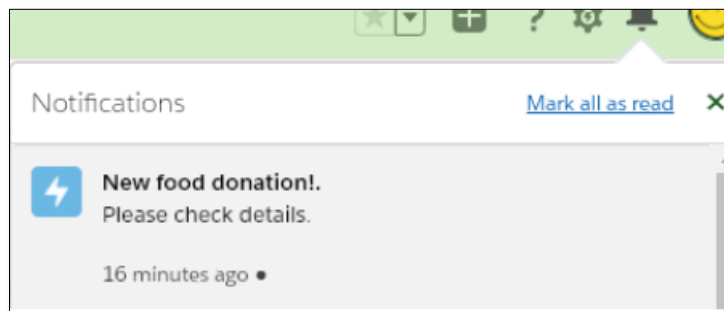


Figura 96-Notificação de doação de alimentos

Na página web também foram criadas regras para a validação dos dados introduzidos.

Por exemplo na figura 96 pode-se observar que quando é introduzido um número de telefone com o formato incorreto, o sistema mostra uma mensagem de erro.



Register in the ZeroWaste platform as a Volunteer

*First Name: Roberto

*Last Name: Adelino

*Birthday: 11/05/1998 [17/12/2020]

*Address(city and street): guarda rua nº x

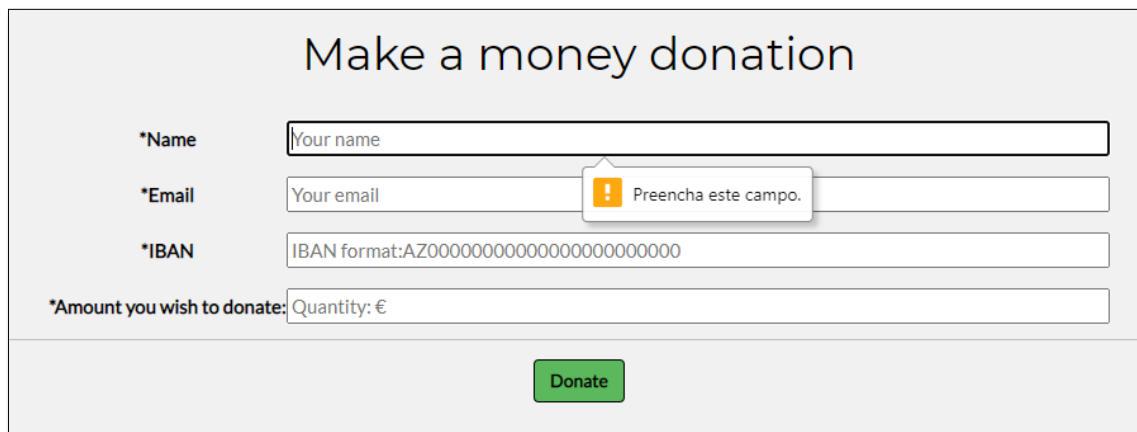
*Email: roberto_adelino_98@hotmail.com

*Phone: 271

! Faça corresponder o formato pedido.

Figura 97-Validação de dados no website

A figura 97 demonstra a mensagem de erro que o sistema apresenta quando o utilizador tenta fazer uma doação sem preencher nenhum campo.



Make a money donation

*Name: Your name

*Email: Your email ! Preencha este campo.

*IBAN: IBAN format:AZ0000000000000000000000

*Amount you wish to donate: Quantity: €

Donate

Figura 98-Validação de dados no website

Ao longo do desenvolvimento do projeto fizeram-se vários testes cada vez que algo novo se implementou tal como o funcionamento de *process builders*, a regras de validação, os filtros de campo, entre outros. A validação irá ser efetuada numa ONG.

8 Conclusões

Este relatório descreve o projeto realizado no âmbito da unidade curricular Projeto de Informática. Foi um projeto realizado com muito trabalho, empenho, tornando-se essencial para a perceção e compreensão do que é um projeto real de Engenharia Informática.

Quanto ao projeto inicialmente encontraram-se algumas dificuldades no desenvolvimento, uma vez que o *Salesforce* está mais direcionado para vendas de produtos e o tema escolhido é o oposto disso, pois são doações.

As ferramentas e linguagens utilizadas na programação foram estudadas e compreendidas, no entanto, é preciso continuar a praticar para dominar estas novas tecnologias.

Os testes funcionais foram feitos exaustivamente durante todo o desenvolvimento de cada fase do projeto e os resultados correspondem aos objetivos previstos. Como trabalho futuro, pretende-se efetuar a validação numa ONG e desta forma proceder a eventuais melhorias e novas ideias.

Por fim, importa referir que todos os objetivos definidos no início do projeto foram implementados com sucesso.

9 Bibliografia

- [1] Continente, [Online]. Available: <https://missao.continente.pt/o-fazemos/planeta/desperdicio-alimentar>. [Acedido em Maio 2020].
- [2] Jornal Económico, [Online]. Available: <https://jornaleconomico.sapo.pt/noticias/desperdicio-alimentar-comida-em-fim-de-prazo-vai-ter-espaco-proprio-nos-supermercados-205167>. [Acedido em Maio 2020].
- [3] Renascença, [Online]. Available: <https://rr.sapo.pt/2019/12/12/o-mundo-em-tres-dimensoes/desperdicio-alimentar-anual-daria-para-alimentar-dois-mil-milhoes-de-pessoas/artigo/174960/>. [Acedido em Maio 2020].
- [4] Nações Unidas, “ONU News,” [Online]. Available: <https://news.un.org/pt/story/2019/07/1680101>. [Acedido em Maio 2020].
- [5] M. Blakeney, “Food loss and food waste: Causes and solutions,” *Food loss and food waste: Causes and solutions*, 2019.
- [6] Baptista P, Campos I, Pires I, Vaz S., Do campo ao garfo. Desperdício Alimentar em Portugal, 2015.
- [7] Hospital Lusíadas, [Online]. Available: <https://rotasaude.lusiadas.pt/prevencao-e-estilo-de-vida/nutricao-e-dieta/evitar-o-desperdicio-alimentar/>. [Acedido em Maio 2020].
- [8] “Objetivos de Desenvolvimento Sustentável,” [Online]. Available: <https://www.ods.pt/>. [Acedido em Maio 2020].
- [9] Assembly, T. G., & Goals, T., “General Assembly. 16301,” 2015.
- [10] “GoodAfter,” [Online]. Available: <https://goodafter.com/pt/>. [Acedido em Maio 2020].
- [11] “TooGoodToGo,” [Online]. Available: <https://toogoodtogo.pt/pt>. [Acedido em Maio 2020].

- [12] T. G. T. Go. [Online]. Available: <https://play.google.com/store/apps/details?id=com.app.tgtg&hl=pt>.
- [13] “Fruta Feia,” [Online]. Available: <https://frutafeia.pt/>. [Acedido em Maio 2020].
- [14] “ReFood,” [Online]. Available: <https://www.re-food.org/pt>. [Acedido em Maio 2020].
- [15] DesenvolvimentoÁgil.com.br, “SCRUM,” [Online]. Available: <https://www.desenvolvimentoagil.com.br/scrum/>.
- [16] LOBA, [Online]. Available: <https://academiasalesforce.loba.pt/>. [Acedido em setembro 2020].
- [17] Salesforce, “Trailhead,” [Online]. Available: <https://trailhead.salesforce.com/pt-BR/home>.
- [18] wikipedia, “UML,” 2020. [Online]. Available: <https://pt.wikipedia.org/wiki/UML>.
- [19] Salesforce, [Online]. Available: <https://www.salesforce.com/eu/products/what-is-salesforce/>.
- [20] Salesforce, “Visualforce,” [Online]. Available: https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages/pages_intro_what_is_it.htm.
- [21] Salesforce, [Online]. Available: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_intro_what_is_apex.htm.
- [22] Microsoft, “Visual Studio Code,” [Online]. Available: <https://code.visualstudio.com/>.
- [23] Salesforce, “appexchange,” [Online]. Available: <https://appexchange.salesforce.com/appxListingDetail?listingId=a0N3000000B5iwREAR>.

- [24] Salesforce, “Appexchange,” [Online]. Available: <https://appexchange.salesforce.com/appxListingDetail?listingId=a0N30000001TLtcEAG>.
- [25] [Online]. Available: <https://tecnologia.culturamix.com/blog/wp-content/gallery/os-tres-pilares-do-scrum/Os-Tr%C3%AAs-Pilares-Do-Scrum-1.jpg>. [Acedido em Maio 2020].

ZeroWaste

Anexos

A 1. Schema builder da aplicação

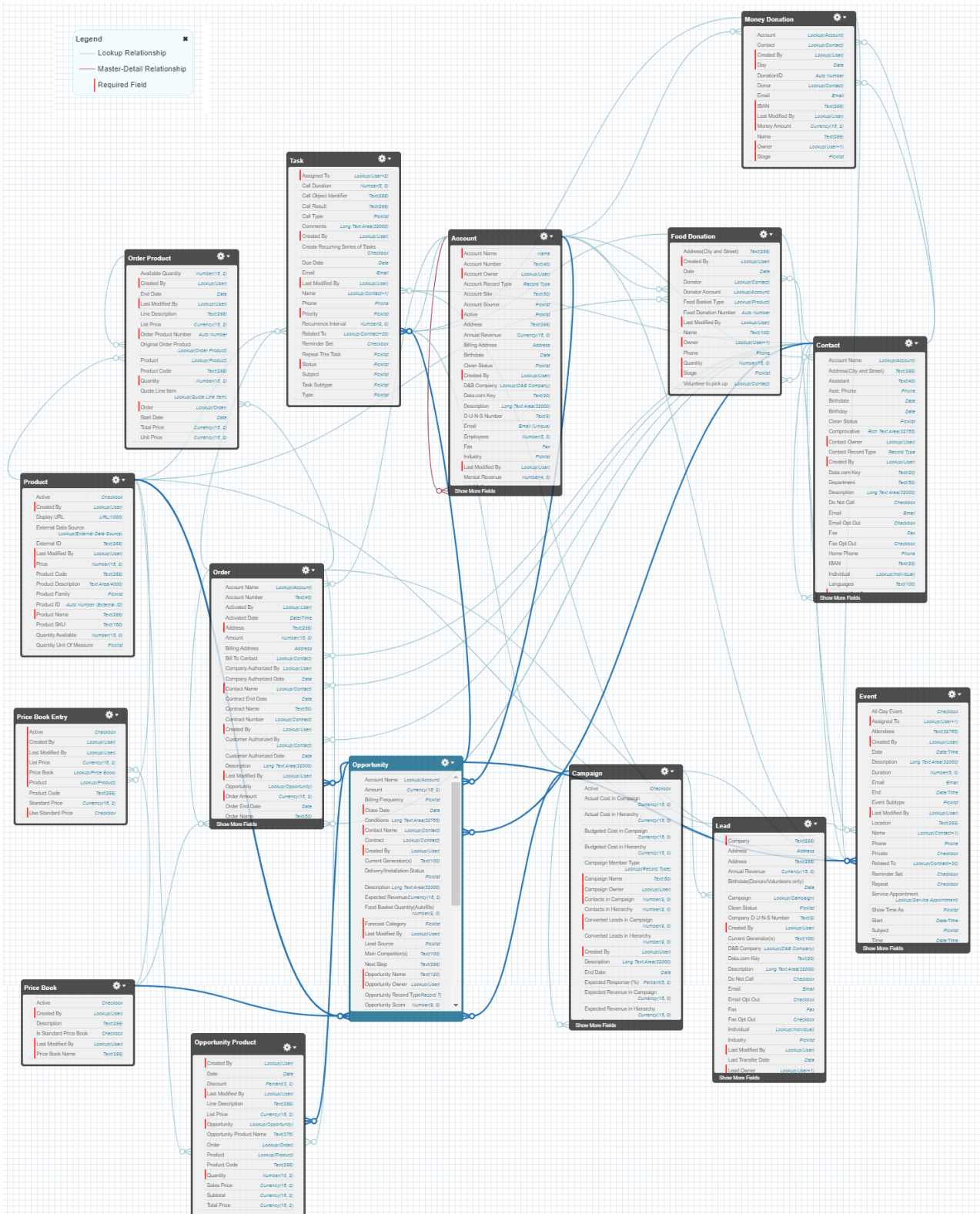


Figura 99-Modelo ER gerado pela plataforma

A 2. Dicionário de dados

A 2.1. Objeto Contacts

Tabela 12-Dicionário de dados do objeto Contacts

Contacts					
Nome	Descrição	Tipo de Dados	Tamanho	Formato	Restrições
IDContact(PK)	Chave primária do objeto.	Varchar	18	Entre 15 e 16 caracteres	Não nulo
Contact Record Type	Tipo de Registo	Record Type	#N/A	#N/A	Não nulo
Name	Nome da família	Varchar	60	Até 60 carateres	Não nulo
Number of People in the household	Número de pessoas do agregado familiar	Integer	2	Até 2 carateres	Não nulo
Mensal Revenue	Rendimento mensal da família	Integer	4	Até 4 carateres	Maiores que 0 Não nulo
Phone	Telefone da família	Integer	9	Até 9 carateres	Não nulo
Email	Email da família	Email	50	Até 50 carateres	Não nulo
Address	Morada da família	Varchar	255	Até 255 carateres	Não nulo
Comprovative	Comprovativo dos dados apresentados	Rich Text Area	32768	Imagem	Não nulo
IBAN	Número do IBAN	Varchar	25	25 caracteres	Restrição no formato
Birthday	Data de nascimento	Date	14	dd-mm-yyyy	Não nulo
People it covers	Número de pessoas que a instituição abrange	Integer	5	Até 5 caracteres	Não nulo Maiores que 0

A 2.2. Operações do Objeto Contacts

Tabela 13 -Conjunto de operações no objeto Contacts

Operações: Classe Contacts	
Inserir ()	Conjunto de operações que permitem inserir os campos da classe.
Editar ()	Conjunto de operações que permitem manipular os campos da classe.
Consultar ()	Conjunto de operações que permitem obter os campos da classe.
Eliminar()	Conjunto de operações que permitem eliminar o registo.
mudarRecordType()	Conjunto de operações que permitem alterar o record type do registo.

A 2.3. Objeto Accounts

Tabela 14- Dicionário de dados do objeto Accounts

Account					
Nome	Descrição	Tipo de Dados	Tamanho	Formato	Restrições
IDAccount(PK)	Chave primária do objeto.	Varchar	18	Entre 15 e 16 caracteres	Não nulo
Account Name	Nome da Conta	Varchar	64	Até 64 caracteres.	Não nulo.
Active	Se a conta está ativa ou não	Boolean	10	True/False	Não nulo.

A 2.4. Operações do Objeto Account

Tabela 15 -Conjunto de operações no objeto Account

Operações: Classe Account	
Criar ()	Conjunto de operações que permitem inserir os campos da classe.
Editar ()	Conjunto de operações que permitem manipular os campos da classe.
Consultar ()	Conjunto de operações que permitem obter os campos da classe.
Eliminar ()	Conjunto de operações que permitem eliminar o registo.

A 2.5. Objeto Requests and sales

Tabela 16 -Dicionário de dados do objeto Requests and sales (opportunity)

Requests and sales					
Nome	Descrição	Tipo de Dados	Tamanho	Formato	Restrições
IDOpportunity(PK)	Chave primária do objeto.	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDAccount(FK)	Chave estrangeira do objeto "Account"	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDContact(FK)	Chave estrangeira do objeto "Contact" que representa o registo da família/instituição ou parceiro	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDProduct(FK)	Chave estrangeira do objeto "Product"	Varchar	18	Entre 15 e 16 caracteres	Não nulo
Oportunity Name	Nome da Oportunidade	Integer	10	Até 10 caracteres	Maiores que 0. Não nulo
Close Date	Data em que o pedido expira	DateTime	14	dd-MM-yyyy	Não nulo

Stage	Estado	Picklist	#N/A	'dd-MM- yyyy HH:mm:ss'	Prospecting ValueProposition Negotiation/Review Closed Won Closed Lost
Quantity	Quantidade de produtos	Integer	2	Até 2 caracteres	Não nulo. Maiores que 0.
Price	Quantidade de alimentos entregues	Integer	2	Até 2 caracteres	Maiores que 0. Não nulo

A 2.6. Operações do Objeto Requests and sales

Tabela 17- Conjunto de operações no objeto Requests and Sales (opportunity)

Operações: Classe Requests and sales	
Criar ()	Conjunto de operações que permitem inserir os campos da classe.
Editar ()	Conjunto de operações que permitem manipular os campos da classe.
Consultar ()	Conjunto de operações que permitem obter os campos da classe.
atualizarQuantidade()	Conjunto de operações que permitem atualizar automaticamente a quantidade de cabazes para cada família ou instituição.
criarOrder()	Conjunto de operações que permitem gerar automaticamente um registo no objeto Requests to deliver.
gerarFatura()	Conjunto de operações que permitem alterar o record type do registo.
mudarRecordType	Conjunto de operações que permitem alterar o record type do registo.

A 2.7. Objeto leads

Tabela 18 Dicionário de dados do objeto Leads

Leads					
Nome	Descrição	Tipo de Dados	Tamanho	Formato	Restrições
IDLead(PK)	Chave primária do objeto.	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDAccount(FK)	Chave estrangeira do objeto "Account"	Varchar	18	Entre 15 e 16 caracteres	Não nulo
Lead Source	Origem da lead	Picklist	#N/A	Internet Ads PhoneInquiry PartnerReferral PurchasedList Word of mouth Website Other	Não nulo
Lead Status	Status da Lead	Picklist	#N/A	Open - Not Contacted Working-Contacted Closed -Converted Closed - Not Converted	Não nulo
Name	Nome	Varchar	255	Até 255 carateres	Não nulo
Number of People in the household	Número de pessoas do agregado familiar	Integer	2	Até 2 carateres	Não nulo
Mensal Revenue	Rendimento mensal da família	Integer	4	Até 4 carateres	Maiores que 0 Não nulo
Phone	Telefone da família	Integer	9	Até 9 carateres	Não nulo

Email	Email da família	Email	50	Até 50 carateres	Não nulo
Address	Morada da família	Varchar	255	Até 255 carateres	Não nulo
Comprovative	Comprovativo dos dados apresentados	Rich Text Area	32768	Imagem	Não nulo
IBAN	Número do IBAN	Varchar	25	25 caracteres	Restrição no formato
Birthday	Data de nascimento	Date	14	dd-mm-yyyy	Não nulo
People it covers	Número de pessoas que a instituição	Integer	5	Até 5 caracteres	Não nulo Maiores que 0

A 2.8. Operações do objeto Leads

Tabela 19 -Conjunto de operações no objeto Leads

Operações: Classe Leads	
Criar ()	Conjunto de operações que permitem inserir os campos da classe.
Editar ()	Conjunto de operações que permitem manipular os campos da classe.
Consultar ()	Conjunto de operações que permitem obter os campos da classe.
Eliminar ()	Conjunto de operações que permitem eliminar o registo.
Atualizar estado ()	Conjunto de operações que permitem alterar o record type do registo.
Converter()	Conjunto de operações que permitem converter uma Lead num Contact.

A 2.9. Objeto Requests to deliver

Tabela 20- Dicionário de dados do objeto Requests to deliver (orders)

Requests to deliver					
Nome	Descrição	Tipo de Dados	Tamanho	Formato	Restrições
IDOrder(PK)	Chave primária do objeto.	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDAccount(FK)	Chave estrangeira do objeto “Account”	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDContat(FK)(Voluntário)	Chave estrangeira do objeto “Account” que representa o voluntário que vai entregar o pedido.	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDContat(FK)(Família/instituição)	Chave estrangeira do objeto “Account” que representa a família ou instituição a que o registo se refere	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDOpportunity(FK)	Chave estrangeira do objeto “Requests and sales”.	Varchar	18	Entre 15 e 16 caracteres	Não nulo
Order start date	Data de início do registo de entrega de pedido	DateTime	14	dd-MM-yyyy	Não nulo
Status	Estado da entrega	Picklist	#N/A	Draft Waiting for volunteer Activated Delivering Delivered	Não nulo
Address	Morada de entrega	Varchar	255	Até 255 caracteres	Não nulo
Quantity	Quantidade a entregar	Integer	2	Até 2 caracteres	Não nulo. Maior que 0.

A 2.10. Operações do objeto Requests to deliver

Tabela 21-Conjunto de operações no objeto Requests to deliver (orders)

Operações: Requests to deliver	
Criar ()	Conjunto de operações que permitem inserir os campos da classe.
Editar ()	Conjunto de operações que permitem manipular os campos da classe.
Consultar ()	Conjunto de operações que permitem obter os campos da classe.
Eliminar ()	Conjunto de operações que permitem eliminar o registo.
Atualizar estado ()	Conjunto de operações que permitem alterar o record type do registo.

A 2.11. Objeto Money Donations

Tabela 22- Dicionário de dados do objeto Money Donations

Money donations					
Nome	Descrição	Tipo de Dados	Tamanho	Formato	Restrições
IDDonation(PK)	Chave primária do objeto.	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDAccount(FK)	Chave estrangeira do objeto "Account"	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDContact(FK)	Chave estrangeira do objeto "Contact"	Varchar	18	Entre 15 e 16 caracteres	Não nulo
Name	Nome do doador quando a doação é feita através do website	Varchar	255	Até 255 caracteres	-
Amount	Valor monetário a doar	Integer	5	Até 5 caracteres	Maior que 0. Não nulo.
IBAN	IBAN	Varchar	25	Formato de IBAN	Não nulo
Stage	Estado da doação	Picklist	#N/A	Initial Stage Transaction Completed	Não nulo
Day	Dia da doação	DateTime	14	dd-MM-yyyy	Não nulo

A 2.12. Operações do objeto Money Donations

Tabela 23 -Conjunto de operações no objeto Money Donations

Operações: Money Donations	
Criar ()	Conjunto de operações que permitem inserir os campos da classe.
Editar ()	Conjunto de operações que permitem manipular os campos da classe.
Consultar ()	Conjunto de operações que permitem obter os campos da classe.
Eliminar ()	Conjunto de operações que permitem eliminar o registo.
Atualizar estado ()	Conjunto de operações que permitem alterar o record type do registo.

A 2.13. Objeto Food Donations

Tabela 24 Dicionário de dados do objeto Food Donations

Food Donations					
Nome	Descrição	Tipo de Dados	Tamanho	Formato	Restrições
IDFoodDonation (PK)	Chave primária do objeto.	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDAccount(FK)	Chave estrangeira do objeto “Account”	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDContact(FK)(Restaurante/supermercado)	Chave estrangeira do objeto “Contact” que representa Restaurante/supermercado que fez a doação	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDContact(FK)(Voluntário)	Chave estrangeira do objeto “Contact” que representa o voluntario que vai recolher os alimentos	Varchar	18	Entre 15 e 16 caracteres	Não nulo
Name	Nome do doador quando a doação é feita através do website	Varchar	255	Até 255 caracteres	-
Address	Morada do doador	Varchar	255	Até 255 caracteres	Não nulo.
Quantity	Quantidade de alimentos a doar	Integer	4	Até 4 caracteres	Não nulo. Maior que 0.
Date	Dia da doação	DateTime	14	dd-MM-yyyy	Não nulo
Stage	Estado da doação	picklist	#N/A	Details provided Volunteer assigned Donation received.	Não nulo
Phone	telefone	Telefone da família	Integer	9	Até 9 caracteres
Email	email	Email da família	Email	50	Até 50 caracteres

A 2.1. Operações do Objeto Food Donations

Tabela 25 -Conjunto de operações no objeto Food Donations

Operações: Food Donations	
Criar ()	Conjunto de operações que permitem inserir os campos da classe.
Editar ()	Conjunto de operações que permitem manipular os campos da classe.
Consultar ()	Conjunto de operações que permitem obter os campos da classe.
Eliminar ()	Conjunto de operações que permitem eliminar o registo.
Atualizar estado ()	Conjunto de operações que permitem alterar o record type do registo.

A 2.2. Objeto Campaigns

Tabela 26- Dicionário de dados do objeto Campaigns

Campaigns					
Nome	Descrição	Tipo de Dados	Tamanho	Formato	Restrições
IDCampaign(PK)	Chave primária do objeto.	Varchar	18	Entre 15 e 16 caracteres	Não nulo
Name	Nome da campanha	Varchar	255	Até 255 caracteres	Não nulo
Active	Se está ativa ou não	Varchar	18	Entre 15 e 16 caracteres	Não nulo
Type	Tipo de campanha	Picklist	#N/A	Conference Webinar Trade Show Public Relations Partners Referral Program Advertisement Banner Ads Direct Mail Email Telemarketing Other	Não nulo.
Status	Estado da campanha	Picklist		Planned In Progress Completed Aborted	
Start date	Data de início	DateTime	14	dd-MM-yyyy	Não nulo
End Date	Data de término	DateTime	14	dd-MM-yyyy	Não nulo
Cost	Custo da campanha	Integer	4	Até 4 caracteres	Não nulo Maior que zero
Nr of leads	Número de leads geradas com a campanha	Integer	#N/A	#N/A	Não nulo

A 2.3. Operações do objeto Campaigns

Tabela 27- Conjunto de operações no objeto Campaigns

Operações: Campaigns	
Criar ()	Conjunto de operações que permitem inserir os campos da classe.
Editar ()	Conjunto de operações que permitem manipular os campos da classe.
Consultar ()	Conjunto de operações que permitem obter os campos da classe.
Eliminar ()	Conjunto de operações que permitem eliminar o registo.
Atualizar estado ()	Conjunto de operações que permitem alterar o record type do registo.

A 2.4. Objeto Tasks

Tabela 28- Dicionário de dados do objeto Tasks

Tasks					
Nome	Descrição	Tipo de Dados	Tamanho	Formato	Restrições
IDTask(PK)	Chave primária do objeto.	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDOrder(FK)	Chave estrangeira do objeto “Requests to deliver”	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDOpportunity((FK)	Chave estrangeira do objeto “Requests and sales”	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDDonation(FK)	Chave estrangeira do objeto “Money Donation”	Varchar	18	Entre 15 e 16 caracteres	Não nulo
IDFoodDonation (FK)	Chave estrangeira do objeto “Food Donations”	Varchar	18	Entre 15 e 16 caracteres	Não nulo
Subject	Descrição da tarefa	Varchar	255	Até 255 caracteres	Não nulo
Due Date	Data de fim	DateTime	14	dd-MM-yyyy	Não nulo

Priority	Pioridade da tarefa	picklist	#N/A	Low Normal High	Não nulo
Status	Estado da tarefa	picklist	#N/A	Not Started In Progress Completed Waiting on someone else Deferred	Não nulo

A 2.5. Operações do Objeto Tasks

Tabela 29 -Conjunto de operações no objeto Tasks

Operações: Tasks	
Criar ()	Conjunto de operações que permitem inserir os campos da classe.
Editar ()	Conjunto de operações que permitem manipular os campos da classe.
Consultar ()	Conjunto de operações que permitem obter os campos da classe.

A 2.6. Objeto Products

Tabela 30- Dicionário de dados do objeto Products

Products					
Nome	Descrição	Tipo de Dados	Tamanho	Formato	Restrições
IDProduct(PK)	Chave primária do objeto.	Varchar	18	Entre 15 e 16 caracteres	Não nulo
Name	Chave estrangeira do objeto “Requests to deliver”	Varchar	255	Até 255 caracteres	Não nulo
Price	Preço do produto	Integer	6	Até 6 caracteres	Não nulo Maior que 0
Quantity	Quantidade disponível	Integer	6	Até 6 caracteres	Não nulo Maior que 0
Active	Se esta ativo ou não	Boolean	10	True/False	Não nulo

A 2.7. Operações do objeto Products

Tabela 31- Conjunto de operações no objeto Products

Operações: Products	
Criar ()	Conjunto de operações que permitem inserir os campos da classe.
Editar ()	Conjunto de operações que permitem manipular os campos da classe.
Consultar ()	Conjunto de operações que permitem obter os campos da classe.
AtualizarQuantidade ()	Conjunto de operações que permitem atualizar a quantidade de alimentos disponíveis para doação.

A 3. Código

A 3.1. Código da classe que gera uma fatura

```

<apex:page standardController="Opportunity" showHeader="false" renderAs="pdf"
>

    <table border="0" cellspacing="0" cellpadding="5" width="100%" height="200%" id="table1">
        <tr height="410%">
            <th width="30%">

                <img src='{!URLFOR($Resource.zerowastelogo)}' title="logo" width="150" height="150" />
            </th>

            <!-- Template Header -->

            <th align="center" width="40%">
                <font face="Arial">
                    <b>ZEROWASTE</b>
                    <br/>INVOICE
                    <br/>
                    <br/>
                    <br/>
                </font>
            </th>

            <!-- Right side of header -->

            <th width="40%" align="right">
                <font face="Arial">
                    <b>Prepared for: {!Opportunity.Contact_Name__r.FirstName}
                    {!Opportunity.Contact_Name__r.LastName}</b>
                </font>
                <br/>
                <b></b>
            </th>
        </tr>

    </table>
    <tr>
        <td width="50%">
            <b>ZeroWaste</b>
        </td>
        <td width="50%">
            <b>Type: {!Opportunity.Account.Name}</b>
        </td>
    </tr>

```



```

</table>
<br/>
<hr/>
<p>
  <b>
    <font color="#000080" face="Arial">Delivered by</font>
  </b>
</p>

<table width="100%" id="table5">

  <tr>
    <td width="75%">Signature:
      <hr color="black" size="1" />
    </td>
  </tr>

</table>

<table border="0" width="100%" id="table3">
  <tr>
    <td width="30%">
      <font face="Arial">
        Start Date:&nbsp;
        <apex:OutputField value="{!Opportunity.Start_Date__c}" />
        <br/> Close Date:&nbsp;
        <apex:OutputField value="{!Opportunity.CloseDate}" />
        <br/>
      </font>
    </td>
    <td width="30%"></td>
    <td width="30%">
      <font face="Arial">
        Payment :&nbsp;
        <apex:OutputField value="{!Opportunity.Contact_Name__r.IB
AN__c }" />
      </font>
    </td>
  </tr>
</table>

<hr/>
<table width="100%" id="table5">
  <tr>
    <td width="50%">
      <b>Receiver signature</b>

```

```
        </td>

    </tr>
    <tr>
        <td width="50%">&nbsp;</td>
        <td width="50%">&nbsp;</td>
    </tr>
    <tr>
        <td width="75%">Signature:
            <hr color="black" size="1" />
        </td>
    </tr>
    <tr>
        <td width="70%">Date:
            <hr color="black" size="1" />
        </td>

    </tr>
</table>

<!-- Footer -->

<div class="footer">
    <p align="center">
        <font face="Arial">
            <i>Copyright ZeroWaste ♥</i>
        </font>
    </p>
</div>

</apex:page>
```

A 3.2. Trigger De renovação de Pedido de famílias e instituições

```

// Automatically create a Renewal Opp for closed won deals
trigger oppRenewal on Opportunity (after update ) {

    List <Opportunity> renewalsfood = new List<Opportunity>();
    if(TriggerHandler1.isFirstTime){
        TriggerHandler1.isFirstTime = false;
        for (Opportunity opp : Trigger.new) {
            // Only create renewal opps for closed won deals

            if (opp.RecordTypeId=='012090000000b87AAA' && opp.StageName == 'Closed Won') {
                Opportunity renewal = new Opportunity();

                renewal.AccountId = opp.AccountId;
                renewal.Contact_Name__c=opp.Contact_Name__c;
                renewal.Name = opp.Name + 'Food request - Auto Renewal';
                renewal.CloseDate = opp.CloseDate + 30; // Add a month
                renewal.Type_of_food_basket__c = opp.Type_of_food_basket__c;
                renewal.Food_Basket_Quantity__c = opp.Food_Basket_Quantity__c;
                renewal.StageName = 'Prospecting';
                renewal.RecordTypeId = opp.RecordTypeId;
                renewal.OwnerId = opp.OwnerId;
                renewalsfood.add(renewal);
            }
        }
        if (renewalsfood.size() > 0 )
            insert renewalsfood;
    }
}

```

A 3.3. Trigger de renovação de plano de sócio

```

// Automatically create a Renewal Opp for closed won deals
trigger oppRenewalPartners on Opportunity (after update) {
    List <Opportunity> renewals = new List<Opportunity>();

    if(TriggerHandler.isFirstTime){
        TriggerHandler.isFirstTime = false;
        for (Opportunity opp : Trigger.new) {
            // Only create renewal opps for closed won deals

            if (opp.StageName == 'Closed Won' && opp.RecordTypeId == '012090000000b8CAA
Q' ) {

                Opportunity renewal = new Opportunity();
                renewal.AccountId = opp.AccountId;
                renewal.Contact_Name__c=opp.Contact_Name__c;
                renewal.Partner_plan__c=opp.Partner_plan__c;

                if(opp.Partner_plan__c =='01t09000000Ae2nAAC' || opp.Partner_plan__c =='01t
09000000Ae2mAAC' ||opp.Partner_plan__c =='01t09000000Ae2hAAC' ||opp.Partner_plan__c =='
01t09000000Ae2qAAC' || opp.Partner_plan__c =='01t09000000Ae2eAAC'){
                    renewal.CloseDate = opp.CloseDate + 365; // Add a year
                    renewal.Name = opp.Name + 'Yearly-Renewal';

                }else if(opp.Partner_plan__c =='01t09000000AetzAAC' || opp.Partner_plan__c
=='01t09000000Ae2dAAC' || opp.Partner_plan__c =='01t09000000Ae2cAAC' ||opp.Partner_plan
__c =='01t09000000Ae2bAAC' ){
                    renewal.CloseDate = opp.CloseDate + 30;//add a month
                    renewal.Name = opp.Name + 'Monthly-Renewal';

                }else if(opp.Partner_plan__c =='01t09000000AdeiAAC' || opp.Partner_plan__c
=='01t09000000Ae2gAAC' ||opp.Partner_plan__c=='01t09000000Ae2lAAC'){
                    renewal.CloseDate = opp.CloseDate + 7;//add a week
                    renewal.Name = opp.Name + 'Weekly-Renewal';

                }else{
                    renewal.CloseDate = opp.CloseDate + 1;//teste
                    renewal.Name = opp.Name + 'ERROR?';

                }

                renewal.Price__c=opp.Price__c;
                renewal.StageName = 'Prospecting';
                renewal.RecordTypeId = opp.RecordTypeId;
                renewal.OwnerId = opp.OwnerId;
                renewals.add(renewal);
            }
        }
    }

    if (renewals.size() > 0 )
        insert renewals;
}

```

A 3.4. Página inicial do website

```

<apex:page standardStylesheets="false" showHeader="false" lightningStyles
heets="false">

  <html lang="en">
  <title>ZeroWaste-HomePage</title>
  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-
scale=1" />
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css
" />
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=
Lato" />
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=
Montserrat" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/f
ont-awesome/4.7.0/css/font-awesome.min.css" />
  <style>
    body,
    h1,h2,h3,h4,h5,h6 {
      font-family: "Lato", sans-serif
    }

    .w3-bar, h1,button {
      font-family: "Montserrat", sans-serif
    }

    .fa-anchor, .fa-coffee {
      font-size: 200px
    }

    header {
      background: url('https://i.imgur.com/9ZHA76k.png');
      background-size: 100% 100%;
    }

    .academia {
      width: 100%;
      height: auto;
    }

    .center {
      margin: auto;
      width: 50%;
    }
  </style>

```



```

<body>

  <!-- Navbar -->
  <div class="w3-top">
    <div class="w3-bar w3-green w3-card w3-left-align w3-large">
      <a class="w3-bar-item w3-button w3-hide-medium w3-hide-
large w3-right w3-padding-large w3-hover-white w3-large w3-
red" href="javascript:void(0);"
        onclick="myFunction()" title="Toggle Navigation Menu"
      >
        <i class="fa fa-bars"></i>
      </a>

      <a href="https://zerowaste-developer-
edition.eu40.force.com/zerowaste" class="w3-bar-item w3-button w3-
padding-large w3-dark-blue">Home</a>
      <a href="https://zerowaste-developer-
edition.eu40.force.com/newfooddonation" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Donate food </a>
      <a href="https://zerowaste-developer-
edition.eu40.force.com/newmoneydonation" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Donate money</a>
      <a href="https://zerowaste-developer-
edition.eu40.force.com/partnersInitial" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Become a partner</a>
      <a href="https://zerowaste-developer-
edition.eu40.force.com/newVolunteer" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Become a Volunteer</a>
      <a href="https://zerowaste-developer-
edition.eu40.force.com/webtolead" class="w3-bar-item w3-button w3-hide-
small w3-padding-large w3-hover-white">Contact Us</a>
    </div>

    <!-- Navbar on small screens -->
    <div id="navDemo" class="w3-bar-block w3-white w3-hide w3-
hide-large w3-hide-medium w3-large">
      <a href="https://zerowaste-developer-
edition.eu40.force.com/zerowaste" class="w3-bar-item w3-button w3-
padding-large w3-dark-blue">Home</a>
      <a href="https://zerowaste-developer-
edition.eu40.force.com/newfooddonation" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Donate food </a>
      <a href="https://zerowaste-developer-
edition.eu40.force.com/newmoneydonation" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Donate money</a>
      <a href="https://zerowaste-developer-
edition.eu40.force.com/partnersInitial" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Become a partner</a>

```

```

        <a href="https://zerowaste-developer-
edition.eu40.force.com/newVolunteer" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Become a Volunteer</a>
        <a href="https://zerowaste-developer-
edition.eu40.force.com/webtolead" class="w3-bar-item w3-button w3-hide-
small w3-padding-large w3-hover-white">Contact Us</a>
    </div>
</div>

<!-- Header -->
<header style="padding:128px 36px; text-align: center; ">
    <h1 class="w3-margin w3-jumbo " style="font-
family:courier; font-weight: bold; ">WELCOME TO THE ZERO WASTE CAUSE</h1>

    <!-- Header    <button class="w3-button w3-black w3-padding-
large w3-large w3-margin-top" >logo se ve</button> -->
</header>
<!-- Second Grid -->
<div class="w3-row-padding w3-light-grey w3-padding-64 w3-
container">
    <div class="w3-content">

        <div class="w3-third w3-center">
            
        </div>

        <div class="w3-twothird">
            <h1>Help us fight food waste</h1>

            <h5 class="w3-padding-
32"> According to the Food and Agriculture Organization of the United Nat
ions (FAO), the global food waste
                per year is approximately 1.3 billion tonnes. Acc
ording to this study, this waste corresponds to
                one third of the world's food production and woul
d be enough to feed twice as many people who are
                affected by hunger worldwide.
            <p>
            </p>
                We are looking for partnering with Restaurants an
d Supermarkets to give the leftover meals and products to people in need.
            </h5>

            <h5 class="w3-padding-
32"> If you are a Restaurant, a Supermarket or a person who wants to help
please hit the button bellow, fill
                the required fields in the form and help us </h5>

```

```

        <button onclick="location.href='https://zerowaste-
developer-
edition.eu40.force.com/webtolead'" type="button">Contact us</button>
        <p class="w3-text-grey">Thank You</p>
    </div>
</div>
</div>
<!-- third Grid -->
<div class="w3-row-padding w3-padding-64 w3-container">
    <div class="w3-content">

        <div class="w3-third w3-center">
            
        </div>

        <div class="w3-twothird">
            <h1>Consider donating money</h1>
            <h5 class="w3-padding-
32"> Picking up and delivering food requires funds, if you can please con
sider donating to our cause</h5>
            <h5 class="w3-padding-
32"> Hit the donate button </h5>
            <button onclick="location.href='https://zerowaste-
developer-
edition.eu40.force.com/newMoneyDonation'" type="button">Donate</button>
            <p class="w3-text-grey">Thank You</p>
        </div>
    </div>
</div>
<!-- 4th Grid -->
<div class="w3-row-padding w3-light-grey w3-padding-64 w3-
container">
    <div class="w3-content">

        <div class="w3-third w3-center">
            
        </div>

        <div class="w3-twothird">
            <h1>Donate food</h1>
            <h5 class="w3-padding-32"> You can donate food </h5>
            <p>Pleas donate products with long shelf life</p>
            <p>We donate food baskets to people in need</p>
            <p>

```

```

                We have 2 types of food baskets : For institution
s and for familys </p>
                Institution Size food baskets are made with 20 produc
ts each and Family Size Food baskets have 10 products
                <h5 class="w3-padding-
32"> Click the button to donate food </h5>
                <button onclick="location.href='https://zerowaste-
developer-
edition.eu40.force.com/newfooddonation'" type="button">Donate</button>
                <p class="w3-text-grey">Thank You</p>
            </div>
        </div>
    </div>

<!-- 5th Grid -->
<div class="w3-row-padding w3-padding-64 w3-container">
    <div class="w3-content">

        <div class="w3-third w3-center">
            
        </div>

        <div class="w3-twothird">
            <h1>Become a Volunteer</h1>
            <h5 class="w3-padding-
32"> Our organization works with volunteers only</h5>
            <h5 class="w3-padding-
32"> Please volunteer and help us deliver and pickup food </h5>
            <button onclick="location.href='https://zerowaste-
developer-
edition.eu40.force.com/newVolunteer'" type="button">Become a volunteer</b
utton>
            <p class="w3-text-grey">Thank You</p>
        </div>
    </div>
</div>
<style>
    table {
        border: 1px solid #000000;
        border-collapse: collapse;
        margin: 0 auto;
        padding: 0px;
        table-layout: fixed;
        min-width: 100%;
    }

    table th {
        text-align: center;

```

```

padding: 4px;
border: 1px solid #000000;
background: #1b601a;
color: #FFFFFF;
font-weight: bold;
text-align: center
}

table td {
padding: 4px;
border: 1px solid #000000;
}

table tr {
background-color: #dbeec9;
color: #000000;
text-align: center;
}
</style>
<table>
<tr>
<th>Type</th>
<th>Rule</th>
<th>Food basket quantity</th>

</tr>
<tr>
<td rowspan='6' colspan='1'>Family</td>
<td>Family mensal revenue less than or equal to 300€ and
nº of people in the household less than or equal to
2
</td>
<td>1</td>
</tr>
<tr>
<td>Family mensal revenue greater than 300 and less than
500 and nº of people in the household less than or equal
to 3 and greater than or equal to 2</td>
<td>2</td>
</tr>
<tr>
<td>Family mensal revenue greater than 300 and less than
500 and nº of people in the household greater than or
equal to 4</td>
<td>3</td>
</tr>
<tr>
<td>Family mensal revenue greater than or equal 500 and 1
ess than or equal to 700 and nº of people in the household

```

```

        less than 5 and greater than or equal to 3</td>
        <td>3</td>
    </tr>
    <tr>
        <td>Family mensal revenue greater than or equal 500 and l
ess than 750 and nº of people in the household greater
        than 5 </td>
        <td>4</td>
    </tr>
    <tr>
        <td>Family mensal revenue greater than or equal 750 and l
ess than 1000 and nº of people in the household greater
        than 6</td>
        <td>5</td>
    </tr>
    <tr>
        <td rowspan='4' colspan='1'>Intitution</td>
        <td>Nº of people helped by the institution less than 30</
td>
        <td>3</td>
    </tr>
    <tr>
        <td>Nº of people helped by the institution greater than o
r equal to 30 and less than 50</td>
        <td>5</td>
    </tr>
    <tr>
        <td>Nº of people helped by the institution greater than o
r equal to 50 and less than 100</td>
        <td>8</td>
    </tr>
    <tr>
        <td>Nº of people helped by the institution greater than o
r equal to 100</td>
        <td>10</td>
    </tr>
</table>
<center>
    <h5 class="w3-padding-
32"> If you are a family or institution and meet this rules please contac
t us </h5>
    <button onclick="location.href='https://zerowaste-developer-
edition.eu40.force.com/webtolead'" type="button">Contact us</button>
</center>
<p></p>
<center>

    <div class="w3-row-padding w3-light-grey w3-padding-64 w3-
container">

```

```

    <h2>Food baskets we donate</h2>
    <p></p>
    <center>
      <div class="w3-third w3-margin-bottom">
        <ul class="w3-ul w3-border w3-hover-shadow">
          <li class="w3-theme">
            <p class="w3-
xlarge">Family size food basket</p>
            </li>

            <li class="w3-padding-16">
              <h2 class="w3-
wide"> 10 Products in the basket</h2>

              </li>

          </ul>

          <button onclick="location.href='https://zerowaste
-developer-
edition.eu40.force.com/newfooddonation'" type="button">Donate Food</butto
n>

        </div>

      </center>
    <center>
      <div class="w3-third w3-margin-bottom">
        <ul class="w3-ul w3-border w3-hover-shadow">
          <li class="w3-theme">
            <p class="w3-
xlarge">Institution size food basket</p>
            </li>

            <li class="w3-padding-16">
              <h2 class="w3-
wide"> 20 Products in the basket</h2>

              </li>

          </ul>

          <button onclick="location.href='https://zerowaste
-developer-
edition.eu40.force.com/newfooddonation'" type="button">Donate Food</butto
n>

        </div>

      </center>
    </div>

```

```

</center>

<!-- Pricing Row WEEKLY-->
<div class="w3-row-padding w3-center w3-padding-64" id="pricing">
  <h2 style="font-weight: bold;">Become a partner</h2>
  <p>We have many partner plans</p>
  <br></br>

  <p></p>

  <h2 style="text-align: center;">Weekly plans</h2>
  <li class="w3-theme-15 w3-padding-24">

    <button onclick="location.href='https://zerowaste-
developer-
edition.eu40.force.com/partnersInitial'" type="button">Become a partner</
button>

  </li>
  <p></p>

  <div class="w3-third w3-margin-bottom">
    <ul class="w3-ul w3-border w3-hover-shadow">
      <li class="w3-theme">
        <p class="w3-xlarge">Weekly</p>
      </li>

      <li class="w3-padding-16">
        <h2 class="w3-wide">
          <i class="fa fa-eur"></i> 5</h2>
          <span class="w3-opacity">per week</span>
        </li>

    </ul>
  </div>

  <div class="w3-third w3-margin-bottom">
    <ul class="w3-ul w3-border w3-hover-shadow">
      <li class="w3-theme-12">
        <p class="w3-xlarge">Weekly</p>
      </li>

      <li class="w3-padding-16">
        <h2 class="w3-wide">
          <i class="fa fa-eur"></i> 10</h2>
          <span class="w3-opacity">per week</span>
        </li>

    </ul>
  </div>

```



```

    <div class="w3-third w3-margin-bottom">
      <ul class="w3-ul w3-border w3-hover-shadow">
        <li class="w3-theme">
          <p class="w3-xlarge">Weekly</p>
        </li>

        <li class="w3-padding-16">
          <h2 class="w3-wide">
            <i class="fa fa-eur"></i> 15</h2>
          <span class="w3-opacity">per week</span>
        </li>

      </ul>
    </div>
  </div>

<p></p>
<center>
  <h2 style="text-align:center;">Monthly plans</h2>
  <li class="w3-theme-15 w3-padding-24">

    <button onclick="location.href='https://zerowaste-
developer-
edition.eu40.force.com/partnersInitial'" type="button">Become a partner</
button>

  </li>
</center>
<p></p>
<!-- Pricing Row MONTHLYYYYYYYYYYYYY-->
<div class="w3-row-padding w3-center w3-padding-64" id="pricing">

  <div class="w3-third w3-margin-bottom">
    <ul class="w3-ul w3-border w3-hover-shadow">
      <li class="w3-theme">
        <p class="w3-xlarge">Monthly</p>
      </li>

      <li class="w3-padding-16">
        <h2 class="w3-wide">
          <i class="fa fa-eur"></i> 10</h2>
        <span class="w3-opacity">per month</span>
      </li>

    </ul>
  </div>

  <div class="w3-third w3-margin-bottom">
    <ul class="w3-ul w3-border w3-hover-shadow">

```

```

<li class="w3-theme-12">
  <p class="w3-xlarge">Monthly</p>
</li>

<li class="w3-padding-16">
  <h2 class="w3-wide">
    <i class="fa fa-eur"></i> 20</h2>
  <span class="w3-opacity">per month</span>
</li>

</ul>
</div>

<div class="w3-third w3-margin-bottom">
  <ul class="w3-ul w3-border w3-hover-shadow">
    <li class="w3-theme">
      <p class="w3-xlarge">Monthly</p>
    </li>

    <li class="w3-padding-16">
      <h2 class="w3-wide">
        <i class="fa fa-eur"></i> 30</h2>
      <span class="w3-opacity">per month</span>
    </li>

  </ul>
</div>

<div class="w3-third w3-margin-bottom">
  <ul class="w3-ul w3-border w3-hover-shadow">
    <li class="w3-theme">
      <p class="w3-xlarge">Monthly</p>
    </li>

    <li class="w3-padding-16">
      <h2 class="w3-wide">
        <i class="fa fa-eur"></i> 50</h2>
      <span class="w3-opacity">per month</span>
    </li>

  </ul>
</div>
</div>

<center>
  <p></p>

  <h2 style="text-align: center;">Yearly plans</h2>
  <li class="w3-theme-15 w3-padding-24">

```

```

        <button onclick="location.href='https://zerowaste-
developer-
edition.eu40.force.com/partnersInitial'" type="button">Become a partner</
button>
    </li>
    <p></p>
</center>
<!-- Pricing Row yearlyyyyyyyyyy-->
<div class="w3-row-padding w3-center w3-padding-64" id="pricing">

    <div class="w3-third w3-margin-bottom">
        <ul class="w3-ul w3-border w3-hover-shadow">
            <li class="w3-theme">
                <p class="w3-xlarge">Yearly</p>
            </li>

            <li class="w3-padding-16">
                <h2 class="w3-wide">
                    <i class="fa fa-eur"></i> 50</h2>
                <span class="w3-opacity">per year</span>
            </li>

        </ul>
    </div>

    <div class="w3-third w3-margin-bottom">
        <ul class="w3-ul w3-border w3-hover-shadow">
            <li class="w3-theme">
                <p class="w3-xlarge">Yearly</p>
            </li>

            <li class="w3-padding-16">
                <h2 class="w3-wide">
                    <i class="fa fa-eur"></i> 100</h2>
                <span class="w3-opacity">per year</span>
            </li>

        </ul>
    </div>

    <div class="w3-third w3-margin-bottom">
        <ul class="w3-ul w3-border w3-hover-shadow">
            <li class="w3-theme">
                <p class="w3-xlarge">Yearly</p>
            </li>

            <li class="w3-padding-16">
                <h2 class="w3-wide">

```

```

        <i class="fa fa-eur"></i> 150</h2>
        <span class="w3-opacity">per year</span>
    </li>

</ul>
</div>

<div class="w3-third w3-margin-bottom">
    <ul class="w3-ul w3-border w3-hover-shadow">
        <li class="w3-theme">
            <p class="w3-xlarge">Yearly</p>
        </li>

        <li class="w3-padding-16">
            <h2 class="w3-wide">
                <i class="fa fa-eur"></i> 200</h2>
                <span class="w3-opacity">per year</span>
            </li>

    </ul>
</div>

<div class="w3-third w3-margin-bottom">
    <ul class="w3-ul w3-border w3-hover-shadow">
        <li class="w3-theme">
            <p class="w3-xlarge">Yearly</p>
        </li>

        <li class="w3-padding-16">
            <h2 class="w3-wide">
                <i class="fa fa-eur"></i> 250</h2>
                <span class="w3-opacity">per year</span>
            </li>
    </ul>
</div>
</div>

<div class="w3-container w3-black w3-center w3-opacity w3-
padding-64">
    <h1 class="w3-margin w3-xlarge">ZER♥WASTE</h1>
</div>
<!-- First Grid -->
<div class="w3-row-padding w3-padding-64 w3-container">
    <div class="w3-content">
        <div class="w3-twothird">
            <h1>Salesforce project developed for:</h1>
            <h5 class="w3-padding-32">2020</h5>
        </div>

```

```

         and
        
    </div>

    <h4 class="w3-text-black" font-
weight="bold"> *This is a fictional project developed to obtain a bachelo
r's degree in computer engineering</h4>
    </div>

</div>
</div>

<!-- Footer -->
<footer class="w3-container w3-padding-64 w3-center w3-opacity">
    <div class="w3-xlarge w3-padding-32">
        <i class="fa fa-facebook-official w3-hover-opacity"></i>
        <i class="fa fa-instagram w3-hover-opacity"></i>
        <i class="fa fa-snapchat w3-hover-opacity"></i>
        <i class="fa fa-pinterest-p w3-hover-opacity"></i>
        <i class="fa fa-twitter w3-hover-opacity"></i>
        <i class="fa fa-linkedin w3-hover-opacity"></i>
    </div>
    <p>ZeroWaste 2020

    </p>
</footer>

<script>
    // Used to toggle the menu on small screens when clicking on
the menu button
    function myFunction() {
        var x = document.getElementById("navDemo");
        if (x.className.indexOf("w3-show") == -1) {
            x.className += " w3-show";
        } else {
            x.className = x.className.replace(" w3-show", "");
        }
    }
</script>

</body>

</html>

</apex:page>

```

A 3.5. Página de doação de alimentos

```

<apex:page standardStylesheets="false" showHeader="false" lightningStylesheet
s="false" controller="foodDonationController"
  docType="html-5.0">

  <html lang="en">
  <title>Institution Food Donation</title>
  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css" />
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lato
" />
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Mont
serrat" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css" />
  <style>
    body, h1,h2,h3,h4,h5, h6 {
      font-family: "Lato", sans-serif
    }

    .w3-bar, h1,button {
      font-family: "Montserrat", sans-serif
    }

    .fa-anchor,.fa-coffee {
      font-size: 200px
    }

    header {
      background: url('https://i.ibb.co/6mx4cLQ/Zero-Waste-
Logo12.png');
      background-size: 100% 100%;
    }
  </style>

  <body>
    <!-- Navbar -->
    <div class="w3-top">
      <div class="w3-bar w3-green w3-card w3-left-align w3-large">
        <a class="w3-bar-item w3-button w3-hide-medium w3-hide-
large w3-right w3-padding-large w3-hover-white w3-large w3-
red" href="javascript:void(0);"
          onclick="myFunction()" title="Toggle Navigation Menu">
          <i class="fa fa-bars"></i>
        </a>

```

```

        <a href="https://zerowaste-developer-
edition.eu40.force.com/zerowaste" class="w3-bar-item w3-button w3-padding-
large w3-dark-blue">Home</a>
        <a href="https://zerowaste-developer-
edition.eu40.force.com/newfooddonation" class="w3-bar-item w3-button w3-hide-
small w3-padding-large w3-hover-white">Donate food </a>
        <a href="https://zerowaste-developer-
edition.eu40.force.com/newmoneydonation" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Donate money</a>
        <a href="https://zerowaste-developer-
edition.eu40.force.com/partnersInitial" class="w3-bar-item w3-button w3-hide-
small w3-padding-large w3-hover-white">Become a partner</a>
        <a href="https://zerowaste-developer-
edition.eu40.force.com/newVolunteer" class="w3-bar-item w3-button w3-hide-
small w3-padding-large w3-hover-white">Become a Volunteer</a>
        <a href="https://zerowaste-developer-
edition.eu40.force.com/webtolead" class="w3-bar-item w3-button w3-hide-
small w3-padding-large w3-hover-white">Contact Us</a>
        <!-- Navbar on small screens -->
        <div id="navDemo" class="w3-bar-block w3-white w3-hide w3-
hide-large w3-hide-medium w3-large">
        </div>
    </div>
</div>
</div>
<apex:form>

    <center>
        <div class="w3-row-padding w3-light-grey w3-padding-64 w3-
container">
            <fieldset>
                <h1>Make a Food donation </h1>

                <apex:pageBlock>
                    <apex:pageBlockSection columns="1">

                        <div class="w3-section">
                            <apex:inputText label="Name:" value="{!fo
odDonationId.name__c}" required="true" style="width:650px" html-
placeholder="Your name"
                            />
                        </div>

                        <div class="w3-section">
                            <apex:inputField label="Phone:" type="tel
" value="{!foodDonationId.phone__c}" required="true" html-pattern="[0-
9]{3}[0-9]{3}[0-9]{3}"

```

```

                style="width:650px" html-
placeholder="9 digit phone number" />
            </div>
            <div class="w3-section">
                <apex:inputText label="Address (city and
street):" value="{!foodDonationId.address__c}" required="true" style="width:6
50px"
                    html-placeholder="Your address" />
            </div>

            <div class="w3-section">
                <apex:inputField html-
min="1" type="number" label="Quantity of products to donate:" value="{!foodDo
nationId.quantity__c}"
                    required="true" style="width:650px" h
html-placeholder="Positive quantity" />
            </div>

        </apex:pageBlockSection>
    </apex:pageBlock>
</fieldset>
<p></p>
<div>

</div>
<center>
    <apex:commandButton style="color: rgb(0, 0, 0); backg
round-color: #5cb85c; border-color: #281c5a; border: 1px solid ; border-
radius: 4px;
        display: inline-block; padding: 6px 12px; margin-bottom: 0; font-
size: 14px; font-weight: bold; line-height: 1.42857143;
        text-align: center; white-space: nowrap; vertical-
align: middle; -ms-touch-action: manipulation; touch-action: manipulation;"
            value="Donate" id="theButton" action="{!Save}" />
    </center>

</div>
</center>
</apex:form>

    <div class="w3-container w3-black w3-center w3-opacity w3-
padding-64">
        <h1 class="w3-margin w3-xlarge">ZER♥WASTE</h1>
    </div>

</body>

</html>
</apex:page>

```


A 3.6. Controlador da página de doação de alimentos

```

public with sharing class foodDonationController {
    String[] basket = new String[]{};
    public Food_Donations__c foodDonationId {get; set;}
    public foodDonationController(){

        foodDonationId= new Food_Donations__c();
    }

    public PageReference test() {
        return null;
    }

    public List<SelectOption> getItems() {
        List<SelectOption> options = new List<SelectOption>();
        options.add(new SelectOption('family','Family Size Food Basket'))
;
        options.add(new SelectOption('intitution','Institution Size Food
Basket'));

        return options;
    }

    public String[] getBasket() {
        return basket;
    }

    public void setBasket(String[] basket) {
        this.basket = basket;
    }

    public PageReference Save() {

        Food_Donations__c foodDonationId2 = new Food_Donations__c();
        foodDonationId2.Quantity__c = foodDonationId.Quantity__c;
        foodDonationId2.stage__c='Details provided';
        foodDonationId2.address__c=foodDonationId.address__c;
        foodDonationId2.phone__c=foodDonationId.phone__c;
        foodDonationId2.name__c=foodDonationId.name__c;
        foodDonationId2.Date__c = Date.today();
        foodDonationId2.Food_Basket_Type__c = '01t09000000Ax1YAAS';

        Insert foodDonationId2;
        PageReference pr = new PageReference('/apex/ThankYou');
        pr.setRedirect(false);
        return pr;
    }
}

```

A 3.7. Página de doação de dinheiro

```
<apex:page standardStylesheets="false" showHeader="false" lightningStylesheets="false" controller="moneyController" docType="html-5.0">

  <html lang="en">
  <title>Money Donation</title>
  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css" />
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lato" />
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Montserrat" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css" />
  <style>
    body,h1,h2,h3,h4,h5,h6 {
      font-family: "Lato", sans-serif
    }

    .w3-bar,h1,button {
      font-family: "Montserrat", sans-serif
    }

    .fa-anchor,
    .fa-coffee {
      font-size: 200px
    }

    header {
      background: url('https://i.ibb.co/6mx4cLQ/Zero-Waste-Logo12.png');
      background-size: 100% 100%;
    }
  </style>

  <body>

    <!-- Navbar -->
    <div class="w3-top">
      <div class="w3-bar w3-green w3-card w3-left-align w3-large">
        <a class="w3-bar-item w3-button w3-hide-medium w3-hide-large w3-right w3-padding-large w3-hover-white w3-large w3-red" href="javascript:void(0);">
```

```

        onclick="myFunction()" title="Toggle Navigation Menu"
    >
        <i class="fa fa-bars"></i>
    </a>

    <a href="https://zerowaste-developer-
edition.eu40.force.com/zerowaste" class="w3-bar-item w3-button w3-
padding-large w3-dark-blue">Home</a>
    <a href="https://zerowaste-developer-
edition.eu40.force.com/newfooddonation" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Donate food </a>
    <a href="https://zerowaste-developer-
edition.eu40.force.com/newmoneydonation" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Donate money</a>
    <a href="https://zerowaste-developer-
edition.eu40.force.com/partnersInitial" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Become a partner</a>
    <a href="https://zerowaste-developer-
edition.eu40.force.com/newVolunteer" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Become a Volunteer</a>
    <a href="https://zerowaste-developer-
edition.eu40.force.com/webtolead" class="w3-bar-item w3-button w3-hide-
small w3-padding-large w3-hover-white">Contact Us</a>
    </div>
    <!-- Navbar on small screens -->
    <div id="navDemo" class="w3-bar-block w3-white w3-hide w3-
hide-large w3-hide-medium w3-large">
    </div>
</div>

<apex:form>
    <center>
        <div class="w3-row-padding w3-light-grey w3-padding-
64 w3-container">

            <fieldset>
                <h1>Make a money donation </h1>

                <apex:pageBlock>

                    <apex:pageBlockSection columns="1">
                        <div class="w3-section">
                            <apex:inputField value="{!donationId.
name__c}" style="width:650px" required="true" html-
placeholder="Your name" />

                        </div>
                    </apex:pageBlockSection>
                </div>
            </div>
        </div>
    </center>
</apex:form>

```

```

                <apex:inputField type="email" value="{
!donationId.email__c}" style="width:650px" required="true" html-
placeholder="Your email"
                />

            </div>
            <div class="w3-section">
                <apex:inputField type="text" value="{
!donationId.iban__c}" style="width:650px" html-pattern="[A-Z]{2}[0-
9]{23}" html-placeholder="IBAN format:AZ000000000000000000000000"
                required="true" />

            </div>
            <div class="w3-section">

                <apex:inputField type="text" label="A
mount you wish to donate:" value="{!donationId.money_amount__c}" style="w
idth:650px"
                required="true" html-
placeholder="Quantity: €" />

            </div>
        </apex:pageBlockSection>
    </apex:pageBlock>

</fieldset>

<p></p>
<center>
    <apex:commandButton style="color: rgb(0, 0, 0); b
ackground-color: #5cb85c; border-
color: #281c5a; border: 1px solid ; border-radius: 4px;
        display: inline-block; padding: 6px 12px; margin-
bottom: 0; font-size: 14px; font-weight: bold; line-height: 1.42857143;
        text-align: center; white-
space: nowrap; vertical-align: middle; -ms-touch-
action: manipulation; touch-action: manipulation;"
        value="Donate" id="theButton" action="{!save}
    " />
    </center>
</div>
</center>
</apex:form>

    <div class="w3-container w3-black w3-center w3-opacity w3-
padding-64">
        <h1 class="w3-margin w3-xlarge">ZER♥WASTE</h1>
    </div>

</body>

```

```
</html>  
</apex:page>
```

A 3.8. Controlador da página de doação de dinheiro

```
public with sharing class moneyController {  
  
    public MoneyDonations__c donationId {get; set;}  
  
    public moneyController(){  
  
        donationId= new MoneyDonations__c();  
  
    }  
  
    public PageReference save() {  
  
        MoneyDonations__c donationId2 = New MoneyDonations__c();  
        donationId2.iban__c = donationId.iban__c;  
        donationId2.money_amount__c = donationId.money_amount__c;  
        donationId2.email__c=donationId.email__c;  
        donationId2.name__c=donationId.name__c;  
        donationId2.day__c=Date.today();  
  
        Insert donationId2;  
        PageReference pr = new PageReference('/apex/ThankYou');  
        pr.setRedirect(false);  
        return pr;  
  
    }  
}
```

A 3.9. Controlador das páginas de venda de planos de sócio

```

public with sharing class PartnerPlanController {

    public String fName{get;set;}
    public String lName{get;set;}
    public String accName{get;set;}
    public String address{get;set;}
    public String phone{get;set;}
    public String email{get;set;}
    public String iban{get;set;}

    public List<Account> accList {get;set;}

    Id rtId = Schema.SObjectType.Contact.getRecordTypeInfoByName().get('Partner').getRecordTypeId();

    Contact con = new Contact();

    public PageReference save(){
        if(lName == null || lName == ''){
            ApexPages.addmessage(new ApexPages.message(ApexPages.severity.WARNING,'Name is Required!'));

            return null;
        }
        else{
            con.FirstName = fName;
            con.LastName = lName;
            con.AccountId = '00109000004Px9YAAS';
            con.Address__c=address;
            con.Email=email;
            con.phone=phone;
            con.IBAN__c=iban;
            con.recordTypeID=rtId;
        }
    }
}

```

```

        insert con;

        PageReference pr = new PageReference('https://zerowaste-
developer-edition.eu40.force.com/added');

        pr.setRedirect(true);

        return pr;
    }
}

public String oppName{get;set;}

public Date oppDate{get;set;}

    Id opprtId = Schema.SObjectType.Opportunity.getRecordTypeInfoByN
ame().get('Partner Plan').getRecordTypeId();

public String prodName{get;set;}

public Id PartnerId{get;set;}

public List<Product2> partnerList {get;set;}

public PageReference callLookup(){
    partnerList = [SELECT Name, Id FROM Product2 WHERE Price__c >
0 ORDER BY Name];

    PageReference pr = new PageReference('https://zerowaste-
developer-edition.eu40.force.com/partnerSecond');

    pr.setRedirect(false);

    return pr;
}

public PageReference goBack(){
    prodName = ApexPages.currentPage().getParameters().get('ProdN
ame');

    PartnerId = ApexPages.currentPage().getParameters().get('Part
nerId');

    PageReference pr = new PageReference('https://zerowaste-
developer-edition.eu40.force.com/added');

    pr.setRedirect(false);

    return pr;
}

```

```
}  
  
public PageReference save2(){  
    Opportunity opp = new Opportunity();  
    opp.Name = 'Partner plan bought from Website';  
    opp.LeadSource='Website';  
    opp.Contact_Name__c = con.Id;  
    opp.AccountId = '00109000004Px9YAAS';  
    opp.StageName='Negotiation/Review';  
    opp.CloseDate=Date.today()+7;  
    opp.Partner_Plan__c=PartnerId;  
    opp.recordTypeID=oppprtId;  
    insert opp;  
    PageReference pr = new PageReference('/ThankYou');  
    pr.setRedirect(false);  
    return pr;  
}  
  
}
```


A 3.10. Controlador da página de recruta de voluntários

```

public with sharing class newVolunteerController {

    public Contact newVolunteerID {get; set;}
        public String fName{get;set;}
        public String lName{get;set;}
        public String accName{get;set;}
        public String address{get;set;}
        public String phone{get;set;}
        public String email{get;set;}
        public Date bday{get;set;}

    public newVolunteerController(){

        newVolunteerID= new Contact();

    }

    public PageReference save(){

        Contact newVolunteerID2 = new Contact();

        newVolunteerID2.FirstName = newVolunteerID.FirstName;
        newVolunteerID2.LastName = newVolunteerID.LastName;
        newVolunteerID2.AccountId = '00109000004PxAfAAK';
        newVolunteerID2.Address__c=newVolunteerID.Address__c;
        newVolunteerID2.Email=newVolunteerID.Email;
        newVolunteerID2.Phone=newVolunteerID.Phone;
        newVolunteerID2.Birthday__c=newVolunteerID.Birthday__c;
        newVolunteerID2.recordTypeID='012090000000eIHAAAY';
        newVolunteerID2.LeadSource='Website';

        insert newVolunteerID2;

        PageReference pr = new PageReference('/apex/ThankYou');
        pr.setRedirect(false);

        return pr;
    }
}

```

A 3.11. Página de contacto

```

<apex:page standardStylesheets="false" showHeader="false" lightningStyles
heets="false">
  <html lang="en">
  <title>Contact us</title>
  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-
scale=1" />
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css
" />
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=
Lato" />
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=
Montserrat" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/f
ont-awesome/4.7.0/css/font-awesome.min.css" />
  <style>
    body,h1,h2,h3,h4,h5,h6 {
      font-family: "Lato", sans-serif
    }
    .w3-bar,h1,button {
      font-family: "Montserrat", sans-serif
    }
    .fa-anchor,.fa-coffee {
      font-size: 200px
    }

    header {
      background: url('https://i.ibb.co/6mx4cLQ/Zero-Waste-
Logo12.png');
      background-size: 100% 100%;
    }
  </style>

  <body>

    <!-- Navbar -->
    <div class="w3-top">
      <div class="w3-bar w3-green w3-card w3-left-align w3-large">
        <a class="w3-bar-item w3-button w3-hide-medium w3-hide-
large w3-right w3-padding-large w3-hover-white w3-large w3-
red" href="javascript:void(0);"
          onclick="myFunction()" title="Toggle Navigation Menu"
        >
          <i class="fa fa-bars"></i>
        </a>

```

```

        <a href="https://zerowaste-developer-
edition.eu40.force.com/zerowaste" class="w3-bar-item w3-button w3-
padding-large w3-dark-blue">Home</a>
        <a href="https://zerowaste-developer-
edition.eu40.force.com/newfooddonation" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Donate food </a>
        <a href="https://zerowaste-developer-
edition.eu40.force.com/newmoneydonation" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Donate money</a>
        <a href="https://zerowaste-developer-
edition.eu40.force.com/partnersInitial" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Become a partner</a>
        <a href="https://zerowaste-developer-
edition.eu40.force.com/newVolunteer" class="w3-bar-item w3-button w3-
hide-small w3-padding-large w3-hover-white">Become a Volunteer</a>
        <a href="https://zerowaste-developer-
edition.eu40.force.com/webtolead" class="w3-bar-item w3-button w3-hide-
small w3-padding-large w3-hover-white">Contact Us</a>
    </div>

</div>
<script type="text/javascript">
    function redirect() {
        var url = "https://www.google.com";
        window.location(url);
    }
</script>

<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=UTF-
8" />

<form action="https://webto.salesforce.com/servlet/servlet.WebToL
ead?encoding=UTF-8" method="POST">
    <center>

        <div class="w3-row-padding w3-light-grey w3-padding-
64 w3-container">
            <div>
                <h1>Fill the form</h1>
            </div>
            <input type="hidden" name="oid" value="00D09000002YQB
u" />
            <input type="hidden" name="retURL" value="https://zer
owaste-developer-edition.eu40.force.com/thankyou" />

            <div class="w3-section">

                <label for="first_name">First Name </label>

```

```

        <input id="first_name" maxlength="40" name="first
_name" size="60" type="text" placeholder="Your first Name" required="true
"
        />
    </div>
    <div class="w3-section">
        <label for="last_name">Last Name</label>
        <input id="last_name" maxlength="80" name="last_n
ame" size="60" type="text" placeholder="Your last name" required="true"
        />
        <br></br>
    </div>
    <div class="w3-section">
        <label for="email">Email</label>
        <input id="email" maxlength="80" name="email" siz
e="62" type="email" required="true" placeholder="Your email" />
        <br></br>
    </div>
    <div class="w3-section">
        <label for="company">Family/intitution/restaurant
/donor/partner/volunteer</label>
        <input id="company" maxlength="40" name="company"
size="25" type="text" required="true" placeholder="Identify yourself" />
        <br></br>
    </div>
    <div class="w3-section">
        Address (city and street):
        <input id="00N09000003PkXN" maxlength="255" name=
"00N09000003PkXN" size="60" type="text" placeholder="Your address" />
        <br></br>
    </div>
    <div class="w3-section">
        <label for="phone">Phone</label>
        <input id="phone" maxlength="40" name="phone" siz
e="60" type="tel" pattern="[0-9]{3}[0-9]{3}[0-
9]{3}" required="true" placeholder="9 digit phone"
        />
        <br></br>
    </div>
    <div class="w3-section">
        Number of people in the household: (family only)
        <input id="00N09000003Q8G1" name="00N09000003Q8G1
" size="45" type="number" min="0" placeholder=" N°" />
        <br></br>
    </div>
    <div class="w3-section">
        Mensal Revenue(Family only):
        <input id="00N09000003Q8G6" name="00N09000003Q8G6
" size="60" type="number" min="0" placeholder="Amount €" />

```

```

        <br></br>
    </div>
    <div class="w3-section">
        People it covers(Institution only):
        <input id="00N09000003Q8GB" name="00N09000003Q8GB
" size="60" type="number" min="0" placeholder="Nº" />
        <br></br>
    </div>
    <div class="w3-section">
        Birthdate(Donors/Volunteers only):
        <div>

                <input id="00N09000003Q8GG" name="00N09000003
Q8GG" size="12" type="date" />

        </div>
        <br></br>
        <br></br>
    </div>

    <div class="w3-input">
        <label for="description">Give us more information
if you need: </label>
        <textarea name="description" rows="4" cols="50" p
laceholder="Write here"></textarea>
        <br></br>
    </div>
    <input style="color: rgb(0, 0, 0); background-
color: #5cb85c; border-color: #281c5a; border: 1px solid ; border-
radius: 4px;
        display: inline-block; padding: 6px 12px; margin-
bottom: 0; font-size: 14px; font-weight: bold; line-height: 1.42857143;
        text-align: center; white-space: nowrap; vertical-
align: middle; -ms-touch-action: manipulation; touch-
action: manipulation;"
        value="Send" type="submit" name="submit" onClick=
"redirect()" />
    </div>
</center>
</form>

    <div class="w3-container w3-black w3-center w3-opacity w3-
padding-64">
        <h1 class="w3-margin w3-xlarge">ZER♥WASTE</h1>
    </div>

</body>
</html>
</apex:page>

```