



FREDERICO MIGUEL COSTA LOPES
Licenciatura em Engenharia Informática

HUMAN NETWORK
PLATAFORMA PROMOTORA DE CAUSAS SOCIAIS

MESTRADO EM ENGENHARIA INFORMÁTICA
Universidade NOVA de Lisboa
month, 2023

HUMAN NETWORK

PLATAFORMA PROMOTORA DE CAUSAS SOCIAIS

FREDERICO MIGUEL COSTA LOPES

Licenciatura em Engenharia Informática

Orientador: João Bernardes

KinetIT Consultant

Coorientador: João Araújo

NOVA University Lisbon

Júri

Presidente: Doutor João Ricardo Viegas da Costa Seco

Prof. Associado, FCT-NOVA

Arguente: Doutora Maria Armanda Simenta Rodrigues Grueau

Prof. Associado, FCT-NOVA

MESTRADO EM ENGENHARIA INFORMÁTICA

Universidade NOVA de Lisboa

month, 2023

Human Network

Plataforma promotora de causas sociais

Copyright © Frederico Miguel Costa Lopes, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

AGRADECIMENTOS

Queria agradecer ao meu orientador João Araújo, que me ajudou bastante e em tão pouco tempo, ao meu coordenador João Bernardes, por todo o apoio fora de horas, à minha empresa por ter possibilitado o desenvolvimento deste projeto, ao meu CEO e Manager por todo o apoio e por serem as pontes entre faculdade e empresa, à minha família por todo o apoio, principalmente aos meus pais, avó e tio, à minha namorada por todo o apoio emocional durante esta fase desafiante da minha vida académica e aos meus amigos por todo o apoio e por me terem auxiliado com os testes e por terem respondido a um questionário, referente à avaliação da aplicação.

RESUMO

Na sociedade atual encontramos desigualdades latentes que impedem muitos de atingirem um nível de vida confortável, lançando-os muitas vezes em desespero. É por isso, mais necessário do que nunca, a existência de instituições de cariz social que promovam ações que proporcionem uma rede de segurança a quem se encontra em situação precária e que lhes permitam uma base de onde poderão retomar o controlo das suas vidas.

Torna-se então fulcral aproximar, de forma eficaz e célere, quem procura ajuda das entidades que a oferecem. Até ao momento desconhece-se a existência de uma plataforma que facilite esta aproximação. É desta carência que nasce este projeto. Através da plataforma Outsystems pretende-se a criação de uma aplicação que opere de forma similar a uma rede social/profissional, promotora de causas sociais, tendo por base os perfis inscritos.

A aplicação aqui apresentada pretende ser pioneira ao permitir um maior outreach para projetos de cariz social de empresas, associações ou instituições, acelerando e facilitando o contacto entre o cidadão e as mesmas. Funcionará como uma agregadora de iniciativas que serão agrupadas segundo critérios diversos, facilitando a pesquisa e a organização da Ação Social, ultrapassando a presente dificuldade na ligação entre a necessidade e a oferta.

Palavras-chave: Cariz social, Outsystems, Associações, Empresas, Pessoas Individuais

ABSTRACT

In today's society we find latent inequalities that prevent many from reaching a comfortable level of life, often throwing them into despair. Therefore, it is more necessary than ever, the existence of institutions of a social nature that promote actions that provide a safety net to those who find themselves in a precarious situation and that give them a base from which they can take control of their lives.

Becomes essential to approach, in an effective and quick way, those who seek help from the entities that offer it. So far, the existence of a platform that facilitates this approach is unknown. It was from this lack of a resource, that this project was born. Through the Outsystems platform, it is intended to create an application that operates similarly to a social/professional network, promoting social causes, based on the registered profiles.

The application presented here intends to be a pioneer, allowing a greater scope for social projects of companies, associations or institutions, accelerating and facilitating the contact between the citizens and them. It will function as an aggregator of initiatives that will be grouped according to different criterias, facilitating the search and the organization of Social Action, overcoming the present difficulty in linking the need and the supply.

Keywords: Social Nature, Outsystems, Associations, Companies, Individuals

ÍNDICE

Índice de Figuras	ix
Índice de Tabelas	xiv
Siglas	xvii
1 Introdução	1
1.1 Contexto e Motivação	1
1.2 Objetivos	2
1.3 Contribuições	3
1.4 Contexto de investigação	3
1.5 Estrutura	3
2 Background e Trabalho Relacionado	4
2.1 Tecnologias de desenvolvimento web	4
2.1.1 Tecnologias Clássicas	4
2.1.2 Tecnologias Low-Code	7
2.1.3 Plataforma Outsystems	8
2.2 Tecnologias usadas	11
2.2.1 Outsystems	12
2.2.2 JIRA	16
2.2.3 Confluence	17
2.2.4 Bizagi Modeler	17
2.2.5 Postman	18
2.2.6 Scribe	18
2.2.7 Google LightHouse	18
2.2.8 JMeter	19
2.3 Agile Scrum	19
2.4 Trabalho relacionado	20
2.4.1 We help Ukraine	20

2.4.2	Free rice	21
2.4.3	Share the meal	21
2.4.4	Charity Miles	21
2.4.5	Human Network	21
2.5	Sumário	22
3	Metodologia de desenvolvimento e requisitos	23
3.1	Visão geral do desenvolvimento do projeto	23
3.2	Definição de requisitos	24
3.2.1	Requisitos funcionais	24
3.2.2	Requisitos não-funcionais	44
3.3	Trabalho desenvolvido	47
3.4	Integrações	48
3.4.1	OTP - Two Factor Authenticator	49
3.4.2	OAuth 2.0 - Google Authentication	50
3.4.3	Google Calendar	50
3.4.4	reCAPTCHA	51
3.4.5	Image Optimizer	52
3.4.6	TouchID	52
3.5	Sumário	52
4	Arquitetura Human Network	55
4.1	Proposta de arquitetura	55
4.1.1	Tipos de utilizadores de Human Network	55
4.1.2	Ambientes	57
4.1.3	Arquitetura Outsystems	57
4.1.4	Modelo de dados	59
4.2	Gestão de colaboradores	70
4.2.1	Crawler - Web Scraping	70
4.3	Sumário	73
5	Otimização	75
5.1	Otimização de perguntas	75
5.2	Otimização de ecrãs	79
5.2.1	LightHouse	80
5.2.2	Ecrã <i>Home</i>	81
5.2.3	Ecrã <i>Projects</i>	82
5.2.4	Ecrã <i>Network_Details</i>	82
5.2.5	Ecrã <i>Profile</i>	83
5.3	Segurança	84
5.3.1	Proteção contra injeção SQL	85
5.3.2	Regras de proteção de dados	86

5.4	Testes de carga	86
5.4.1	Ecrã Home	87
5.4.2	Ecrã Projects	88
5.5	Sumário	89
6	Validação	90
6.1	Cenários de uso	90
6.1.1	Criador de Projetos	90
6.1.2	Inscrito em Projetos	92
7	Avaliação	96
7.1	Testes de Usabilidade	96
7.2	Testes Funcionais	96
7.3	Resumo do questionário	97
7.4	Visão geral da avaliação	97
7.5	Sumário	103
8	Conclusões	104
8.1	Contribuições	104
8.2	Limitações	105
8.3	Trabalho futuro	105
	Bibliografia	107
	Apêndices	
A	Perguntas do questionários	110
	Anexos	
I	Perguntas SQL	114
II	Manual de Utilização - Criação de Projetos	119
III	Manual de Utilização - Inscrição em Projetos	127
IV	Manual de Utilização - Perfil de utilizador	132
V	Robots.txt	135
VI	Sitemap.xml	136

ÍNDICE DE FIGURAS

2.1	Diagrama de ambientes e ferramentas Outsystems	9
2.2	Lista de aplicações de um ambiente Outsystems	12
2.3	Informação de uma aplicação	12
2.4	Workspace de aplicações - IDE Outsystems	13
2.5	Integration Studio - Ferramenta de criação de componentes	13
2.6	Forge - Repositório de componentes Outsystems	14
2.7	Service Center - gestor de ambiente	14
2.8	Lifetime - Gestor dos ambientes existentes numa fábrica	15
2.9	Discovery - Ferramenta visual de gestão de arquitetura	15
2.10	AI Mentor Studio - Sistema de gestão de dívida técnica	16
2.11	JIRA - Sistema de monitorização de tarefas	17
2.12	Confluence - Sistema de gestão de documentação	17
2.13	Bizagi Modeler - Plataforma de modelação de processos	18
2.14	Postman - Plataforma de construção e testes de APIs	18
2.15	Scribe - Plataforma que gera manuais de utilização	18
2.16	LightHouse - Ferramenta de medição de qualidade de páginas web	19
2.17	JMeter - Plataforma de desenhada para testar, testes de carga	19
2.18	Ciclo de vida de Agile Scrum	20
3.1	Ecrã de registo de utilizador e Sistema reCAPTCHA	25
3.2	Ecrã de perfil de utilizador	25
3.3	Modelo de Processos - Registo de utilizador	26
3.4	Ecrã de Login	26
3.5	Modelo de Processos - Login	27
3.6	Modelo de Processos - Gestão de patrocinadores	27
3.7	Modelo de Processos - Gestão de categorias	28
3.8	Modelo de Processos - Gestão de categorias de utilizadores	29
3.9	Modelo de Processos - Gestão de redes	29
3.10	Gestão de Colaboradores	30

3.11	Modelo de Processos - Gestão de colaboradores	31
3.12	Modelo de Processos - Remover colaborador	32
3.13	Ecrã de edição de projetos	32
3.14	Ecrã de edição de projeto - lista de eventos	33
3.15	Ecrã de edição de projeto - lista de submissões	33
3.16	Ecrã de edição de projeto - lista de FAQs	33
3.17	Modelo de Processos - Gestão de projetos	34
3.18	Modelo de Processos - Remover projeto	34
3.19	Modelo de Processos - Gestão de categorias de projeto	35
3.20	Ecrã de lista de projetos	35
3.21	Ecrã de lista de projetos do utilizador	36
3.22	Ecrã de detalhe de projeto com uma lista de projetos sugeridos	36
3.23	Ecrã de lista de projetos sugeridos - aplicação mobile	36
3.24	Ecrã de edição de projeto - criação de eventos	37
3.25	Modelo de Processos - Gestão de eventos de projetos	38
3.26	Modelo de Processos - Remover eventos de projetos	38
3.27	Ecrã de detalhe de projeto - inscrição num evento	39
3.28	Ecrã de perfil de utilizador - calendário de eventos	39
3.29	Modelo de Processos - Inscrição em eventos por utilizadores	40
3.30	Modelo de Processos - Prescrição em eventos por utilizadores	40
3.31	Modelo de Processos - Gestão de FAQs de projetos	41
3.32	Sistema de mensagens - Chat	42
3.33	Modelo de Processos - Sistema de mensagens	42
3.34	Sistema de mudança de linguagem	43
3.35	Modelo de Processos - Sistema de extração de informação (Crawler)	44
3.36	Aplicação mobile - ecrã de login	45
3.37	Aplicação mobile - ecrã com lista de projetos sugeridos	46
3.38	Aplicação mobile - ecrã de perfil de utilizador	47
3.39	Aplicação mobile - ecrã calendário com lista de eventos	48
3.40	Aplicação mobile - sistema de mensagens	49
3.41	Microsoft Authenticator	49
3.42	Configuração de autenticação de dois fatores na aplicação	50
3.43	Estado da aplicação depois de Login	50
3.44	Google Authentication	50
3.45	Bloco de Login com o botão para autenticação pela Google	51
3.46	Google Calendar	51
3.47	Google Calendar - Evento criado	51
3.48	reCAPTCHA	51
3.49	Registo de utilizador com reCAPTCHA	52
3.50	Image Optimizer	52
3.51	TouchID	52

4.1	Arquitetura Human Network	56
4.2	Diagrama de ambientes e ferramentas Outsystems	57
4.3	Three layer canvas - Divisão de módulos Outsystems	58
4.4	Aplicação Web - Ferramenta Discovery	59
4.5	Aplicação Mobile - Ferramenta Discovery	59
4.6	Serviços Core da aplicação (Web/Mobile) - Ferramenta Discovery	60
4.7	Modelo de Entidades e os seus atributos	61
4.8	Modelo relacional	63
4.9	Modelo abstrato	66
4.10	Modelo físico	67
4.11	SQL Server em ambiente Outsystems	67
4.12	Camada de gestão de empresas e trabalhadores	68
4.13	Camada de gestão de projetos	68
4.14	Camada de gestão de configurações	68
4.15	Definições de entidades	70
4.16	Gestão de colaboradores	71
4.17	Ciclo de vida Web Scraping	71
4.18	Aplicação Crawler - ecrã inicial	72
4.19	Aplicação Crawler - ecrã de detalhe de websites	72
4.20	Aplicação Crawler - ecrã de configuração de palavras-chave	73
4.21	Parte essencial do código do Crawler	74
5.1	LightHouse- Métricas gerais do ecrã <i>Home</i>	81
5.2	LightHouse - Métricas de tempo de resposta do ecrã <i>Home</i>	82
5.3	LightHouse- Métricas gerais do ecrã <i>Projects</i>	82
5.4	LightHouse - Métricas de tempo de resposta do ecrã <i>Projects</i>	82
5.5	LightHouse- Métricas gerais do ecrã <i>Network_Details</i>	83
5.6	LightHouse - Métricas de tempo de resposta do ecrã <i>Network_Details</i>	83
5.7	LightHouse- Métricas gerais do ecrã <i>Profile</i>	84
5.8	LightHouse - Métricas de tempo de resposta do ecrã <i>Profile</i>	84
5.9	Tentativa de criação de projeto através do Postman	85
5.10	Teste de injeção de código SQL	86
5.11	Tentativa de extração de dados da aplicação Human Network	86
5.12	Teste JMeter - Ecrã Home com 100 utilizadores	87
5.13	Teste JMeter - Ecrã Home com 1000 utilizadores	88
5.14	Teste JMeter - Ecrã Projects com 100 utilizadores	88
5.15	Teste JMeter - Ecrã Projects com 1000 utilizadores	89
6.1	Manual de utilização - Criação de Projetos (passo 1)	91
6.2	Manual de utilização - Criação de Projetos (passos 8 e 9)	91
6.3	Manual de utilização - Criação de Projetos (passos 10 e 11)	92

6.4	Manual de utilização - Inscrição em Projetos (passos 2 e 3)	93
6.5	Manual de utilização - Inscrição em Projetos (passos 6 e 7)	94
6.6	Manual de utilização - Inscrição em Projetos (passos 8 e 9)	95
7.1	Questionário - Foi fácil encontrar onde criar redes?	98
7.2	Questionário - Foi fácil gerir redes?	98
7.3	Questionário - Foi fácil encontrar onde criar projetos?	99
7.4	Questionário - Foi fácil encontrar onde gerir os projetos?	99
7.5	Questionário - Foi fácil gerir projetos?	100
7.6	Questionário - Foi fácil criar eventos?	100
7.7	Questionário - Foi fácil inscrever em projetos?	101
7.8	Questionário - Foi fácil inscrever em eventos?	101
7.9	Questionário - Projetos sugeridos faziam sentido?	102
7.10	Questionário - Foi necessário o constante uso do Manual de utilização?	102
7.11	Questionário - Avaliação do tempo de resposta da aplicação	102
7.12	Questionário - Avaliação da facilidade de utilização da aplicação	103
A.1	Questionário - Pergunta 1	110
A.2	Questionário - Pergunta 2	110
A.3	Questionário - Pergunta 3	110
A.4	Questionário - Pergunta 4	111
A.5	Questionário - Pergunta 5	111
A.6	Questionário - Pergunta 6	111
A.7	Questionário - Pergunta 7	111
A.8	Questionário - Pergunta 8	111
A.9	Questionário - Pergunta 9	112
A.10	Questionário - Pergunta 10	112
A.11	Questionário - Pergunta 11	112
A.12	Questionário - Pergunta 12	112
A.13	Questionário - Pergunta 13	112
A.14	Questionário - Pergunta 14	113
A.15	Questionário - Pergunta 15	113
A.16	Questionário - Pergunta 16	113
A.17	Questionário - Pergunta 17	113
II.1	Manual de utilização - Criação de Projetos (passo 1)	119
II.2	Manual de utilização - Criação de Projetos (passos 2 e 3)	120
II.3	Manual de utilização - Criação de Projetos (passos 4 e 5)	121
II.4	Manual de utilização - Criação de Projetos (passos 6 e 7)	122
II.5	Manual de utilização - Criação de Projetos (passos 8 e 9)	123
II.6	Manual de utilização - Criação de Projetos (passos 10 e 11)	124
II.7	Manual de utilização - Criação de Projetos (passos 12 e 13)	125

II.8	Manual de utilização - Criação de Projetos (passos 14 e 15)	126
III.1	Manual de utilização - Inscrição em Projetos (passo 1)	127
III.2	Manual de utilização - Inscrição em Projetos (passos 2 e 3)	128
III.3	Manual de utilização - Inscrição em Projetos (passos 4 e 5)	129
III.4	Manual de utilização - Inscrição em Projetos (passos 6 e 7)	130
III.5	Manual de utilização - Inscrição em Projetos (passos 8 e 9)	131
IV.1	Manual de utilização - Perfil (passos 1 e 2)	132
IV.2	Manual de utilização - Perfil (passos 3 e 4)	133
IV.3	Manual de utilização - Perfil (passos 5 e 6)	134

ÍNDICE DE TABELAS

2.1	Web tradicional vs Outsystems	11
3.1	User Stories para utilizadores gerais criadas a partir dos modelos de processo	53
3.2	User Stories referentes a utilizadores de associações criadas a partir dos modelos de processo	54
3.3	User Stories referentes ao programador e a utilizadores <i>Backoffice</i> criadas a partir dos modelos de processo	54
8.1	User Stories para trabalho futuro	106

ÍNDICE DE LISTAGENS

3.1 Sistema de atualização de mensagens	43
---	----

SIGLAS

API	Application Programming Interface
AWS	Amazon Web Services
BPMN	Business Process Model and Notation
CDN	Content Delivery Network
CRUD	Create Read Update Delete
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
CSV	Comma-separated values
DDoS	Distributed denial-of-service
FAQ	Frequently Asked Questions
HMAC	Hash-based message authentication
HTML	HyperText Markup Language
HTTPS	Hyper Text Transfer Protocol Secure
IA	Inteligência Artificial
IDE	Integrated Development Environment
JS	JavaScript
JSON	Javascript Object Notation
LCDP	Low-code development platform
MVP	Minimum Value Product
MVT	Model View Template

PaaS	Platform as a Service
PAM	Programa Alimentar Mundial
PO	Product Owner
SEO	Search Engine Optimization
SQL	Structured Query Language
SSR	Server side rendering
TIN	Taxpayer Identification Number
TPB	Theory of planned behavior
URL	Uniform Resource Locator
US	User stories
WHU	We help Ukrain
XSS	Cross-Site Scripting

INTRODUÇÃO

Neste capítulo é apresentado o contexto, motivação, os objetivos, contribuições e estrutura do projeto.

1.1 Contexto e Motivação

É necessária a existência de uma plataforma que ajude instituições de cariz social a promover ações que proporcionem uma rede de segurança a quem se encontra em situação precária. É neste âmbito que surge a proposta desta tese. A elaboração de uma plataforma que promova estes projetos e que aproxime as empresas ou pessoas individuais que procurem ajudar, das entidades que a oferecem.

Human Network irá atender a essa necessidade, incentivando atividades vitais para o crescimento da nossa sociedade; também atuará como um acelerador na conclusão dessas tarefas, facilitando a comunicação e conexão entre associações e empresas ou indivíduos, permitindo que associações criem projetos de ajuda social e que empresas ou indivíduos se inscrevam nos projetos.

As tarefas criadas na plataforma são designadas por **projetos**. Os **projetos** são criados por utilizadores pertencentes a **associações**. **Eventos** são ocorrências de **projetos** em que utilizadores pertencentes a **companhias** ou utilizadores individuais se podem inscrever e, cada **projeto** pode ter um ou mais **eventos**.

Este tipo de plataformas está a tornar-se cada vez mais popular. Recentemente, a Outsystems criou a aplicação We Help Ukrain (**WHU**) para ajudar refugiados da guerra entre a Rússia e a Ucrânia [2]. A fim de acomodar melhor os refugiados, esta plataforma permite que os mesmos procurem cidadãos portugueses que estejam dispostos a ajudar com alimentação, roupas, abrigo, emprego, educação e outras informações pertinentes ao acolhimento dos mesmos no país.

Muitos funcionários da KinetIT, contribuíram para o desenvolvimento da aplicação **WHU**. Como resultado dessa ajuda, nasceu a ideia de desenvolver uma plataforma mais abrangente. Apesar da imensa adesão para o desenvolvimento, **WHU** exigiu dezenas de pessoas para garantir a sua entrega rápida. Se a **Human Network** já existisse, não

haveria necessidade de desenvolver esta aplicação, pois todas as suas funções poderiam ser substituídas pela criação de projetos na plataforma, cujo o propósito é acelerar este tipo de iniciativas.

Neste sentido, a tese é focada não só no desenvolvimento desta plataforma, mas também em garantir robustez e escalabilidade da aplicação, para que a sua utilização seja global e caso seja necessário alguma manutenção, seja facilmente integrada na plataforma.

1.2 Objetivos

O objetivo principal deste projeto é a conceção de um produto que apoia a criação de uma rede social/profissional promotora de causas sociais, com base nos perfis inscritos e atividade de cada um. Esta aplicação consistirá numa plataforma agregadora de iniciativas que poderão ser agrupadas segundo diversos critérios de forma a facilitar a pesquisa.

Esta plataforma foi desenvolvida tanto em web como em mobile utilizando a tecnologia low-code Outsystems, sendo que a maior parte da lógica cairá sobre a aplicação web e a aplicação mobile terá algumas funcionalidades clonadas de web, mas destaca-se nas funcionalidades offline.

Nesta aplicação à parte da gestão de empresas, indivíduos e os seus projetos serão implementadas outras funcionalidades, tais como:

- Implementação de um sistema de chat onde as associações e empresas ou indivíduos consigam obter informação e chegar a um acordo sobre os projetos publicados.
- Um sistema de notificações para fortalecer o sistema de chat e para enviar notificações para os dispositivos móveis sobre certos eventos.
- Um sistema de inteligência artificial que filtre imagens inapropriadas de forma a proteger a integridade da aplicação e da KinetIT.
- Um sistema de autenticação de dois passos para os utilizadores que queiram garantir maior segurança das suas contas, dentro de outras funcionalidades (explicado em maior detalhe na secção 3.4).
- Desenvolvimento de uma aplicação robusta e escalável, de fácil manutenção, de modo a que novos requisitos sejam fáceis de implementar e sem necessidade de reestruturar o modelo ou arquitetura da aplicação.

O projeto foi desenvolvido seguindo uma metodologia Agile Scrum, de forma a obter feedback do que foi desenvolvido e se existe alguma melhoria a ser feita (explicado em maior detalhe na secção 3.2).

1.3 Contribuições

Human Network é uma plataforma que não se limita a contribuir para uma melhor procura e exposição de projetos de cariz social. Em suma, as contribuições do projeto são os seguintes:

- Apoiar atividades de cariz social.
- Incentivar a participação de pessoas individuais/empresas em eventos de cariz social.
- Facilitar o intercâmbio de informação necessidade/procura entre associações.
- Encontrar com menos esforço causas sociais nas quais possamos participar.
- Prestar serviço a empresas que procuram causas sociais.
- Organizar eventos de cariz social por parte das Associações.
- Encontrar eventos de cariz social nos quais possamos participar.
- Ultrapassar a dificuldade existente em ligar as associações às empresas.
- Ultrapassar a dificuldade de pessoas individuais participarem em eventos de cariz social.

1.4 Contexto de investigação

Todas as fases de investigação e desenvolvimento serão realizadas em contexto empresarial na KinetIT sobre supervisão e acompanhamento do orientador.

1.5 Estrutura

- No capítulo 2 são analisadas e brevemente explicado o funcionamento de outras plataformas de cariz social, tanto como a comparação destas à plataforma proposta nesta tese e apresentada a tecnologia Outsystems, as suas vantagens e o motivo porque se escolheu esta tecnologia para desenvolver o projeto.
- No capítulo 3 é apresentada a definição de requisitos, as tecnologias utilizadas, integrações terceiras, arquitetura da solução a desenvolver e o plano de trabalho composto por um diagrama de Gantt, trabalho já desenvolvido e a desenvolver.
- No capítulo 4 serão feitos testes e validação dos requisitos e avaliação da aplicação a nível de performance, acessibilidade, usabilidade, segurança, entre outros princípios, através de questionários sobre a utilização da aplicação e utilização de ferramentas de auditoria como Lighthouse.

BACKGROUND E TRABALHO RELACIONADO

Neste capítulo é apresentada a plataforma Outsystems, explicada a sua escolha e vantagens em relação a outras tecnologias (low-code e tecnologias clássicas). Também são apresentadas a metodologia Agile Scrum, plataformas usada para o desenvolvimento, teste e manutenção da aplicação. Por fim é feita uma análise a plataformas promotoras de causas sociais, uma breve explicação sobre o seu funcionamento, em que áreas se focam, forma como ajudam as pessoas e no final uma comparação com a plataforma proposta nesta tese, **Human Network**. Também é explicada a escolha da tecnologia Outsystems e as vantagens em relação a outras tecnologias.

2.1 Tecnologias de desenvolvimento web

Nesta secção são apresentadas tecnologias web clássicas, comparando-as a tecnologias low-code. No final é apresentada a tecnologia escolhida para desenvolver o projeto (Outsystems), tais como os prós e contras da utilização desta tecnologia versus as outras tecnologias low-code.

Uma aplicação web é um software desenvolvido em tecnologias web (HTML, Javascript e CSS) [3] e acedido a partir de um navegador web.

Estas aplicações podem ser desenvolvidas usando vários tipos de tecnologias. Apesar de o mais popular ser a utilização de frameworks de Javascript como ReactJS, Svelte, Angular, Vue, com linguagens de programação como Go, C# ou Java no Backend, existem outras tecnologias também utilizadas como Django, uma framework web de Python, Ruby on rails, tecnologias low-code e no-code.

2.1.1 Tecnologias Clássicas

Nesta subsecção é feita uma análise a três tecnologias clássicas, ReactJS, Django e Ruby on Rails. Serão explicadas também algumas das vantagens e desvantagens de cada tecnologia, tal como a desvantagem da utilização de tecnologias clássicas.

ReactJS

Uma biblioteca de Javascript de código aberto, criado pela Facebook em 2011, é a biblioteca mais popular de Javascript, sendo que maior parte dos projetos desenvolvidos em Javascript ou Typescript usam ReactJS. ReactJS ao contrário de Vanilla Javascript, é baseado em componentes [4].

Vantagens e desvantagens:

- Arquitetura baseada em componentes que divide a página em blocos independentes e permite uma criação, reutilização e gestão, sem depender de qualquer outro componente da página.
- Fácil de aprender para quem vem de Vanilla Javascript, pois maior parte dos recursos usados no desenvolvimento destas aplicações, são bibliotecas Javascript.
- Permite o desenvolvimento de aplicações mobile sem ter de aprender uma linguagem diferente.
- Apesar da popularidade da biblioteca, esta tem uma documentação fraca e está em constantes mudanças a nível de arquitetura (muitas funcionalidades que ficam descontinuadas).
- A extensão JSX que junta HTML com Javascript. Esta extensão muda a forma como HTML é escrito e complica a leitura de código, visto que um ficheiro contém tanto HTML como Javascript.
- Apesar do nome, esta biblioteca na realidade não é reativa, é necessário forçar o estado de modo a forçar mudanças no UI.
- Apesar das vantagens do Virtual DOM, esta característica pode tornar o ecrã lento, quando existe muita lógica de ecrã.

Django

É uma framework web escrita em Python, criada em 2003 por Lawrence Journal-World, que segue o padrão [MVT](#) e que incentiva o desenvolvimento rápido, design limpo e pragmático. [5]

Vantagens e desvantagens:

- Django é bastante seguro, encriptando informação sensível como passwords em serviços GET.
- Graças à sua arquitetura, esta framework oferece um desenvolvimento rápido, permitindo o trabalho em componentes em paralelo.
- Maior controlo sobre [CDNs](#), sendo este um dos motivos que aplicações como Instagram, Netflix e Amazon Prime usam Django.

- Django é monolítico, sendo que todo o tipo de código da aplicação vive numa só code base.
- Não é a melhor opção para projetos pequenos. Django contém muito código que ocupa processamento e tempo de servidor.

Ruby on Rails

É uma framework web escrita em Ruby, criada em 2004 por David Heinemeier Hansson. É uma das bibliotecas Ruby mais populares e uma das principais razões pelas quais os programadores escolhem aprender Ruby. Torna o desenvolvimento web mais fácil, fornecendo uma estrutura pré-construída para desenvolvimento. [6]

Vantagens e desvantagens:

- Contém um grande número de ferramentas e bibliotecas úteis.
- Fácil de aprender.
- Tem uma grande e ativa comunidade e uma boa documentação.
- Pouco flexível a nível de personalização de ferramentas oferecidas pela framework.
- Não é das melhores frameworks a nível de performance.
- Tal como ReactJS é uma framework fortemente baseada em utilização de bibliotecas desenvolvidas pela comunidade, que nunca é bom a nível de manutenção de código.

Aplicações web tradicional tendem a ter uma pilha de tecnologias (tech stack) bastante vasta, pois as linguagens de programação e frameworks para frontend e backend, não são as únicas escolhas a serem feitas. [7] É necessário ainda:

- Escolher entre javascript ou typescript.
- Framework vs não framework/Vanilla Javascript.
- Linguagem backend (existem várias opções).
- Bootstrap [8], Tailwind [9], outra biblioteca de estilo ou criar uma de raiz.
- Um agrupador de módulos como Webpack ou Vite.
- Repositório de código como Github, Gitlab ou Bitbucket.
- Que base de dados usar e que tipo usar (relacional, não relacional ou outra).
- Que fornecedor cloud utilizar (se é que se vai publicar a aplicação em cloud), entre outras opções para a pilha de tecnologias.

A organização desta pilha tende a ser uma parte complicada e demorada de um projeto. Low-code permite que o programador se abstraia de certas escolhas referentes à pilha de tecnologias, ajudando-o a focar-se na rápida entrega e num bom desenvolvimento dos requisitos.

2.1.2 Tecnologias Low-Code

Low-code é um método de desenvolvimento de aplicações que permite a criação de aplicações com o mínimo uso de código escrito, substituindo esta abordagem pelo uso de uma linguagem visual orientada por modelos, com uma interface gráfica de *drag&drop*.

Existem várias razões para se escolher ou migrar de web tradicional para *low-code*. Greg Whitten, diretor de arquitetura aplicacional na empresa Accelerated Focus, foi uma das pessoas que fez esta migração. “Trabalho como programador de web tradicional há 20 anos. Agora, faço parte do primeiro parceiro OutSystems no oeste do Canadá. E um dos equívocos mais comuns que ouço sobre low-code é o medo de que o low-code vá substituir os programadores, porque se toda a gente tem a capacidade de fazer aplicações com interfaces de drag & drop, então a necessidade de contratar talentos para aplicações de qualidade de código deixam de ser necessários. Sim, o low-code introduz uma mudança no paradigma de desenvolvimento tradicional, mas agora, mais do que nunca, os programadores são cruciais numa equipa.” [10]

Microsoft PowerApps

É uma plataforma low-code baseada em spreadsheet-like criada pela Microsoft em 2015. [11]

Vantagens e desvantagens:

- Fácil de aprender.
- Habilidade de ligar a aplicações Microsoft, tais como Excel, Sharepoint, Power BI e outras.
- Má reutilização de código devido à impossibilidade de criar funções.
- Não permite que múltiplos utilizadores usem a aplicação ao mesmo tempo.
- Muito difícil fazer debug.

UiPath

É uma plataforma de automação de processos robóticos para automação de ponta a ponta em alta escala. O software UiPath oferece soluções para empresas automatizarem tarefas repetitivas de escritório para uma rápida transformação dos negócios. Foi fundada pelo empresário romeno Daniel Dines no ano de 2005. [12]

Vantagens e desvantagens:

- Reforça boas práticas ajudando no desenvolvimento de robôs, fornecendo um esqueleto.
- IDE visual de drag & drop, que permite que utilizadores com pouca ou nenhuma base de programação consigam perceber melhor o fluxo e consigam desenvolver mais facilmente.
- É uma plataforma muito limitada, só permite automação de processos.
- Os relatórios de erros são muito limitados, tornando mais difícil a resolução de erros.

Outsystems

Outsystems [13] é uma das plataformas low-code mais poderosas e das mais usadas do mercado. Criada por Paulo Rosado em 2001, esta tem vindo a crescer desde então, não só pela facilidade de aprendizagem, mas também porque, “under the hood”, usa linguagens de programação e frameworks como C#(.NET) e Javascript(ReactJS, NextJS) que são bastante populares e com muito boa performance.

Dentro das tecnologias low-code, Outsystems é das que mais se destaca [14], não só por ser das mais antigas, mas também por acompanhar as tecnologias mais populares. A requisito da empresa KinetIT que usa Outsystems como tecnologia principal nos seus projetos e devido às vantagens 2.1.3 que esta tecnologia tem em comparação com tecnologias clássicas e outras tecnologias low-code escolheu-se Outsystems como tecnologia principal do projeto.

2.1.3 Plataforma Outsystems

Outsystems é uma [PaaS](#) (Platform as a Service), sendo que as suas aplicações são instaladas em [AWS](#) (Amazon Web Services) como se pode ver figura na 2.1, facilitando a publicação e criação de aplicações e a sua gestão. De forma a gerir as aplicações publicadas, Outsystems oferece um sistema chamado Service Center (ver figura 2.2.1), que é uma ferramenta de monitorização de aplicações.

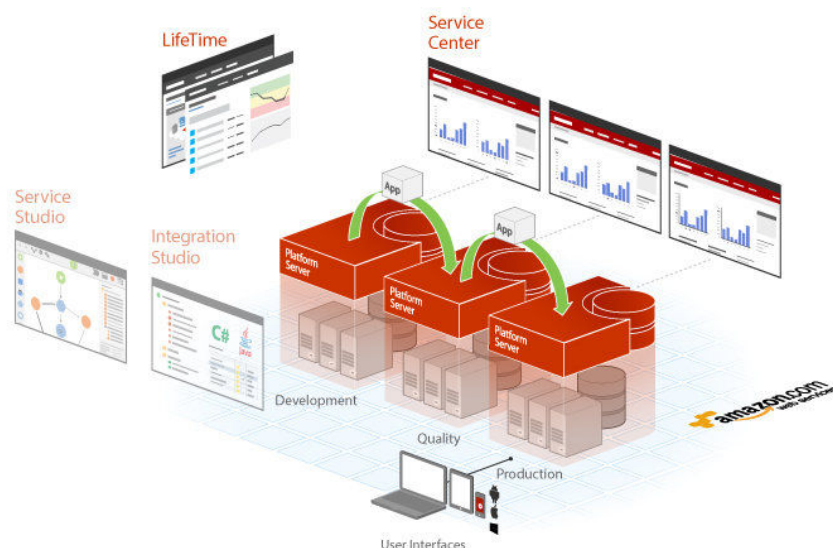


Figura 2.1: Diagrama de ambientes e ferramentas Outsystems

Várias empresas estão a apostar em ferramentas low-code para o desenvolvimento de aplicações. Dentro destas empresas destaca-se a NOS, que foi a primeira empresa a desenvolver uma aplicação Outsystems. A NOS tem vindo a mudar a sua fábrica de modo a migrar todo o tipo de aplicações para Outsystems, destacando-se as duas aplicações mais usadas da NOS, CinemasNOS e Portal de cliente (aplicação Mobile) que foram rescritas em Outsystems.

Esta plataforma é cada vez mais usada tal como outras tecnologias low-code.

“Digitalização, desenvolvimento rápido e de baixo custo e uma nova base de comunicação entre negócio e programadores estão a impulsionar o aparecimento de plataformas low-code (LCDPs)” [15]

Em Abril de 2022, a Amazon lançou Amplify Studio, uma plataforma low-code. Várias empresas estão a apostar em plataformas low-code, demonstrando que este tipo de plataformas são o futuro.

“A plataforma low-code permite a criação e entrega rápida de aplicações de negócio com o mínimo de esforço para escrever numa linguagem de programação e exige o mínimo de esforço possível para instalação e configuração de ambientes e implementação.” [16]

Outsystems tem várias vantagens, sendo que a principal é a abstração da linguagem de programação. Esta no início era escrita em Java no Backend e ASP.NET no Frontend, tendo havido várias evoluções, como a passagem de Java para C# e Web tradicional para ReactJS, sendo que estas mudanças pouco afetam a forma como a plataforma é utilizada, mantendo os mesmos métodos de desenvolvimento e em casos com a biblioteca ReactJS, a criação de ecrãs e o ciclo de vida dos mesmos, replica a da biblioteca, tornando mais fácil para programadores desta biblioteca migrarem para Outsystems.

Mas esta não é a única vantagem desta plataforma:

- Outsystems torna muito simples a migração de dados de sistemas externos para a

plataforma tal como, se assim for o requisito, manter os dados no sistema externo e integrar os mesmos na plataforma.

- É mais fácil obter boa performance, escalabilidade, segurança, CI/CD e manter um código limpo. A plataforma contém vários programas que ajudam no desenvolvimento e manutenção de aplicações (explicado em pormenor na sub-secção [2.2.1](#)).
- Publicação de aplicações no poder de um botão. Enquanto que muitas outras infraestruturas cloud precisam de definições prévias de forma a ser possível publicar aplicações, no caso de Outsystems, estas definições são feitas pela própria plataforma, simplificando bastante a parte inicial, a publicação da aplicação.
- Gestão de utilizadores, dentro de outros temas importantes para o desenvolvimento de uma aplicação, já desenvolvidas “out of the box”. Outsystems oferece um sistema de gestão de utilizadores completo, desde uma tabela de utilizadores já criada, tabela de roles, um utilizador admin já criado e abstrai o programador do sistema de autenticação de serviços. A plataforma utiliza um sistema de autenticação de tokens, refresh/access token, que é o sistema mais usado e bastante seguro para a autenticação de utilizadores em aplicações web.
- Acompanhamento das novas tecnologias. Outsystems utiliza C# no Backend e ReactJS no Frontend, tal como a metaframework NextJS para aplicações [SSR](#) e React Native em mobile, permitindo o desenvolvimento de aplicações Android e IOS.
- Fácil integração de componentes de terceiros. Existe um site, que se chama Forge [\[17\]](#), que contém vários componentes criadas por utilizadores de Outsystems, tal como integrações feitas pela a equipa de Outsystems no próprio Service Studio.
- Fácil de escalar, com ferramentas que ajudam a reduzir a dívida técnica, a encontrar erros e prevenir certos erros.
- Em comparação com Microsoft PowerApps, é possível criar funções, garantindo reutilização de código.
- Os relatórios de erro contém bastante informação relevante para ajudar à rápida correção de erros.
- Não é limitada a desenvolvimento de aplicações focadas num tipo de negócio, como é o caso de Appian ou Salesforce.

Destaca-se ainda as principais diferenças entre web tradicional e a tecnologia Outsystems na seguinte tabela [2.1](#).

Apesar das imensas vantagens, Outsystems tem algumas desvantagens, tais como custo muito elevado de licenças, tornando mais difícil empresas com pouco orçamento de

PRINCIPAIS DIFERENÇAS	WEB TRADICIONAL	OUTSYSTEMS
EXPERIÊNCIA	A MUDANÇA DE LINGUAGEM DE PROGRAMAÇÃO PODE SER DIFÍCIL E PARECER QUE SE ESTÁ A COMEÇAR DE NOVO	A INTERFACE DE DRAG & DROP FACILITA A COMPREENSÃO DOS FLUXOS POR PARTE DE NÃO PROGRAMADORES E UTILIZADORES CORPORATIVOS. O BENEFÍCIO DA VELOCIDADE DO LOW-CODE É ENORME.
TEMPO CONSUMIDO NO COMEÇO DO PROJETO	PARA INICIAR UM PROJETO, É NECESSÁRIO UM SERVIDOR WEB, UM SERVIDOR DE BASE DE DADOS E UM REPOSITÓRIO DE CÓDIGO.	PLATAFORMAS LOW-CODE JÁ TÊM ISTO DEFINIDO DE RAIZ.
PUBLICAÇÃO	OS AMBIENTES TÊM DE SER CONFIGURADOS. ISSO MULTIPLICA TODO O ESFORÇO DE CONFIGURAÇÃO DO AMBIENTE DE DESENVOLVIMENTO. EM SEGUIDA, É NECESSÁRIO UM PLANO DE PUBLICAÇÃO PARA GARANTIR MANUALMENTE QUE AS DEPENDÊNCIAS CORRETAS SEJAM INCLUÍDAS E ATUALIZADAS.	COM OUTSYSTEMS, OS AMBIENTES SÃO CONFIGURADOS AUTOMATICAMENTE, E CRIAR UM PLANO DE PUBLICAÇÃO É TÃO FÁCIL COMO SELECIONAR A APLICAÇÃO QUE PRETENDE PUBLICAR. A PLATAFORMA ENCARRREGA-SE DE ENCONTRAR AS DEPENDÊNCIAS QUE FORAM ALTERADAS E ADICIONA-AS AUTOMATICAMENTE AO PLANO, BEM COMO QUAISQUER OUTRAS APLICAÇÕES QUE SEJAM AFETADAS POR ESSAS DEPENDÊNCIAS.
REUTILIZAÇÃO	NECESSÁRIO COMEÇAR OS PROJETOS DE NOVO	EM OUTSYSTEMS É POSSÍVEL A REUTILIZAÇÃO DE MÓDULOS ANTERIORMENTE DESENVOLVIDOS NOUTROS PROJETOS.

Tabela 2.1: Web tradicional vs Outsystems

usarem Outsystems, debug ainda está longe de ser perfeito, documentação, em algumas bibliotecas e componentes, é pouco robusta, dentro de outras desvantagens. Mas apesar das desvantagens desta tecnologia, as vantagens, comparando Outsystems a outras tecnologias low-code e a tecnologias clássicas, superam as desvantagens.

2.2 Tecnologias usadas

Nesta secção são apresentadas todas as tecnologias e ferramentas usadas para o desenvolvimento da aplicação.

No desenvolvimento do projeto serão usadas várias tecnologias, sendo que a principal é a plataforma low-code Outsystems.

2.2.1 Outsystems

Outsystems oferece várias ferramentas para a manutenção de ambientes, aplicações, bibliotecas, componentes, entre outras funcionalidades.

Service Studio

Ambiente low-code de desenvolvimento visual que permite:

- A criação de aplicações e módulos no servidor.
- A criação de interfaces para Web Traditional(ASP.NET), ReactJS e aplicações mobile.
- Define o modelo de dados.
- Define processos de negócio e timers.
- Permite fazer debug.

Na área de trabalho consegue ver a lista de aplicações que estão no servidor (ver figura 2.2) e em cada aplicação consegue ver a lista de dependências, os módulos e os detalhes da aplicação (ver figura 2.3).

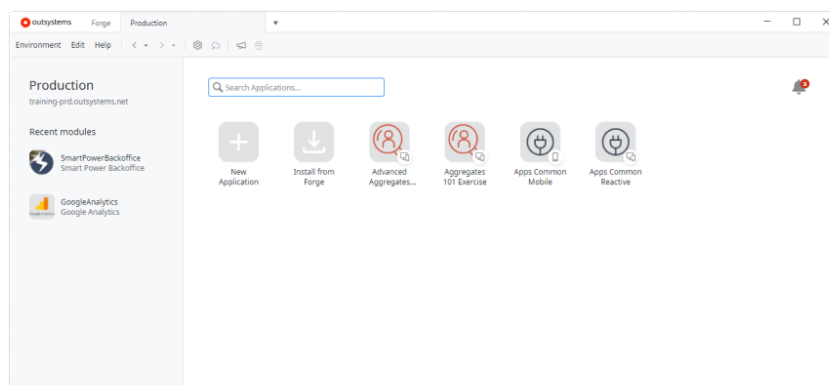


Figura 2.2: Lista de aplicações de um ambiente Outsystems

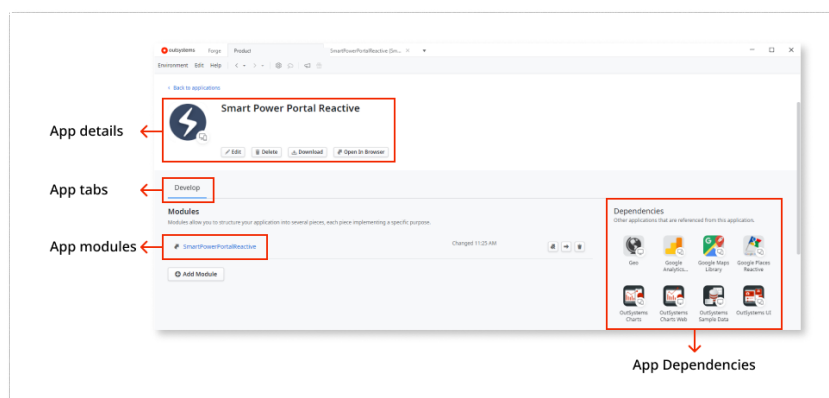


Figura 2.3: Informação de uma aplicação

Esta Aplicação alerta o programador de más práticas e eventuais problemas no seu código, como a exposição de variáveis numa pergunta sem a sanitização das mesmas, ciclos infinitos, várias chamadas ao servidor numa função do lado do cliente, código morto (não utilizado), dentro de outros (ver figura 2.4).

Service Studio contém ainda aceleradores como Scaffolding, temas gerados pela plataforma, ecrãs de login, menu, layouts e ecrã de permissões inválidas já feitos, debug e sistema de controle de versões distribuído.

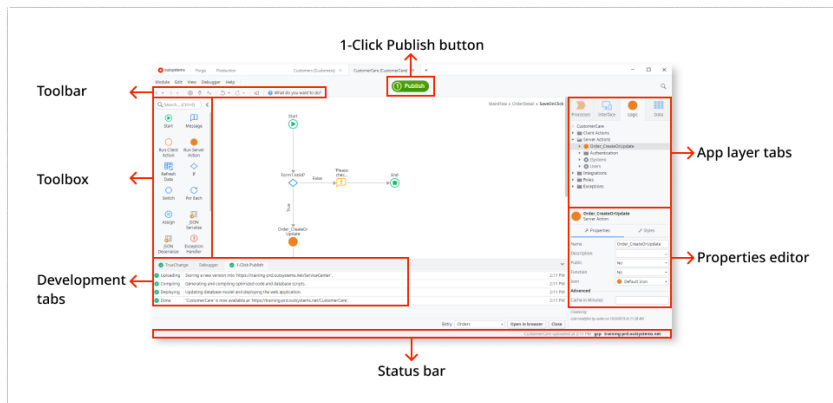


Figura 2.4: Workspace de aplicações - IDE Outsystems

Integration Studio

Ferramenta que permite a criação e gestão de extensões. Estas extensões são desenvolvidas em C#.

Apesar do vasto número de componentes da Forge (ver figura 2.6) da Outsystems, algumas funcionalidades podem ainda não estar implementadas e caso seja necessário, Integration Studio é a ferramenta para isso (ver figura 2.5).

A cada ano que passa fica mais improvável a necessidade de desenvolvimento de componentes através do Integration Studio, no entanto esta ferramenta impede a limitação de certas funcionalidades.

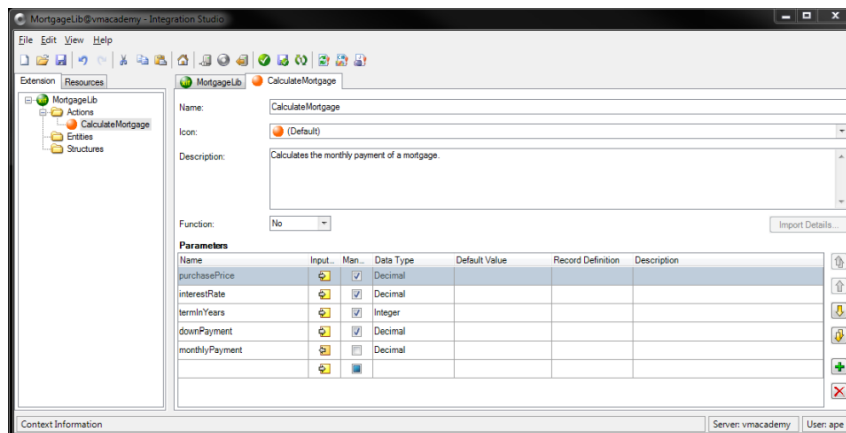


Figura 2.5: Integration Studio - Ferramenta de criação de componentes

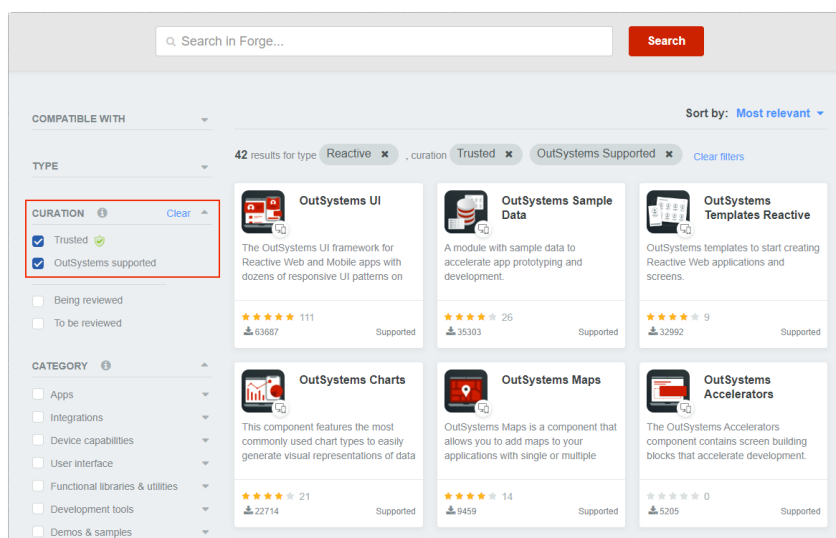


Figura 2.6: Forge - Repositório de componentes Outsystems

Service Center

Ambiente de gestão de aplicações, logs, propriedades dos ambientes, integrações, APIs, soluções, segurança, análise de dados, entre outras funcionalidades (ver figura 2.7).

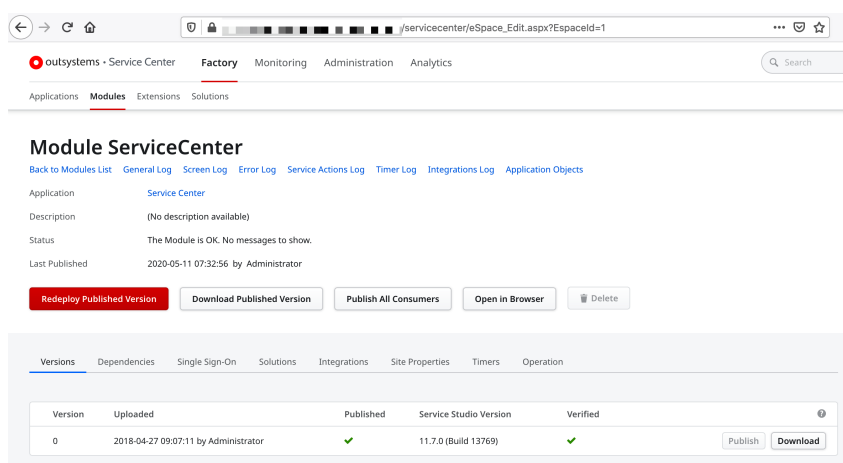


Figura 2.7: Service Center - gestor de ambiente

Grande parte da sua utilização é focada na análise de dados, processos, performance de serviços, perguntas ou lógica e, logs, sendo que em outras tecnologias, estas utilidades têm de ser implementadas de raiz ou integrar plataformas ou componentes terceiros, normalmente com um custo adicional. Outsystems oferece este tipo de ferramentas numa só.

Lifetime

Ferramenta que gere os vários ambientes, subida de aplicações de ambientes, utilizadores da plataforma, entre outras funcionalidades (ver figura 2.8).

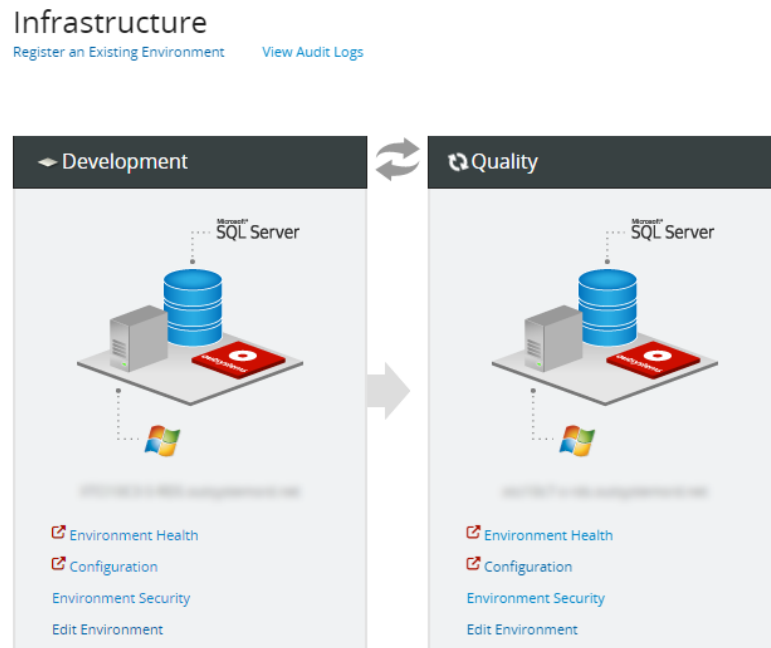


Figura 2.8: Lifetime - Gestor dos ambientes existentes numa fábrica

Grande parte da sua utilização é focada na manutenção das aplicações a nível de versões por ambientes e a sua estabilidade e os recursos utilizados em cada ambiente.

Discovery

Ferramenta visual que ajuda a analisar, medir e perceber como melhorar a arquitetura da fábrica (ver figura 2.9).

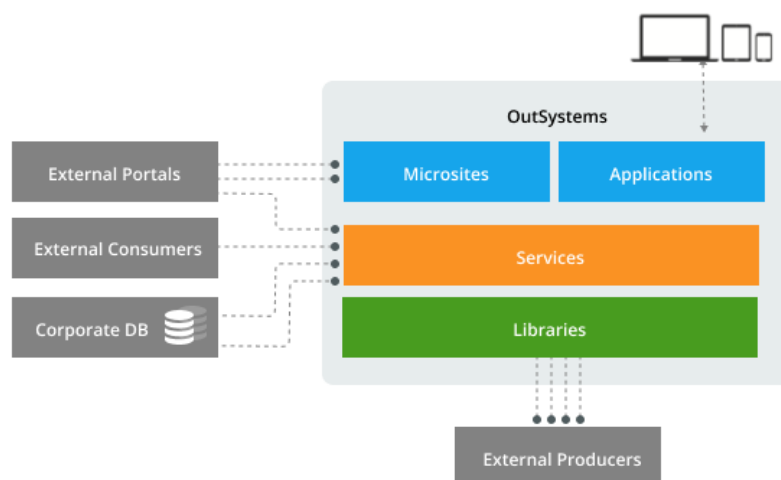


Figura 2.9: Discovery - Ferramenta visual de gestão de arquitetura

Ajuda a perceber se as aplicações estão a seguir as boas práticas, senão existem referências erradas, como módulos de serviços referenciarem módulos de frontend, ou referências circulares.

AI Mentor Studio

Ferramenta de inteligência artificial que calcula a dívida técnica de uma aplicação (ver figura 2.10), priorização de velocidade em relação a qualidade. Esta ferramenta mostra todo o tipo de falhas de uma aplicação:

- Falhas de segurança.
- Má performance em certas acções.
- Más práticas frontend como inline CSS, chamadas ao servidor em eventos JavaScript, entre outros.
- Repetição de código.
- Módulos monolíticos.
- Erros de arquitetura.
- Falta de documentação no código, entre outras falhas.

O melhor desta plataforma é que não só identifica os erros, mas como oferece soluções.



Figura 2.10: AI Mentor Studio - Sistema de gestão de dívida técnica

O que distancia Outsystems de outras tecnologias low-code é o leque de ferramentas oferecidas sem qualquer custo adicional.

2.2.2 JIRA

Software desenvolvido pela Atlassian [18]. É uma ferramenta que permite a monitorização de projetos, garantido a gestão de atividades (ver figura 2.11).

De forma a garantir uma boa gestão e organização do projeto, decidiu-se utilizar esta ferramenta como plataforma de gestão do projeto Agile Scrum.

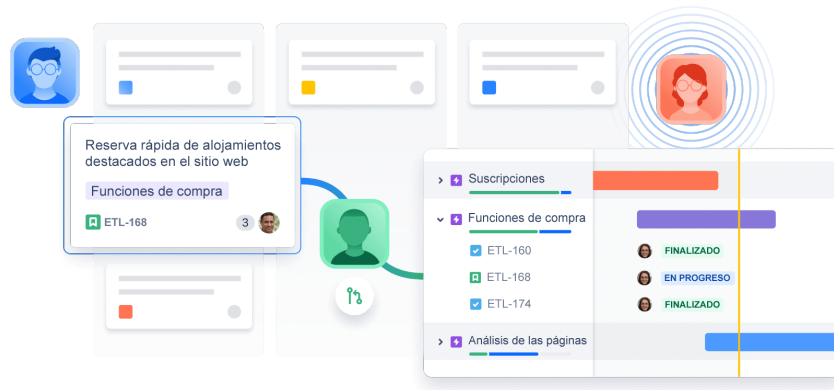


Figura 2.11: JIRA - Sistema de monitorização de tarefas

2.2.3 Confluence

Wiki desenvolvido pela Atlassian [19]. É uma ferramenta usada para manter documentação de projetos (ver figura 2.12).

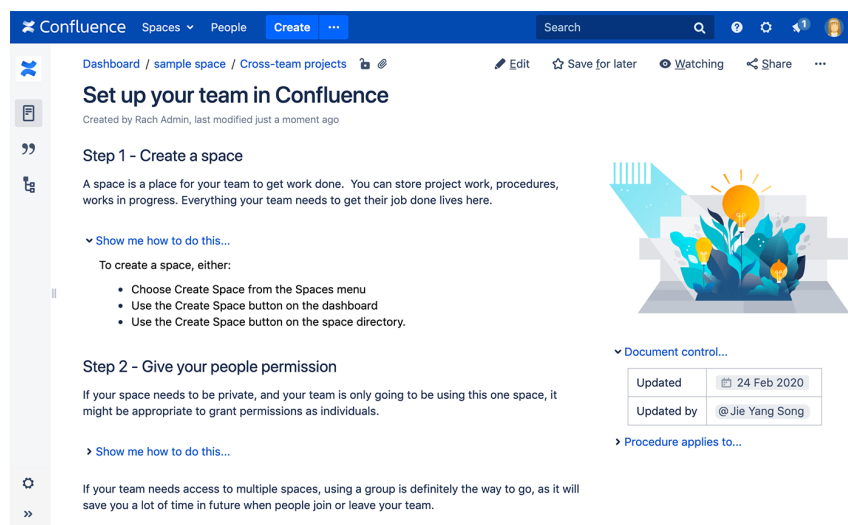


Figura 2.12: Confluence - Sistema de gestão de documentação

Esta ferramenta é usada para manter tanto a documentação funcional, como a documentação mais lógica (boas práticas, ações gerais, dentro de outros temas) do projeto

2.2.4 Bizagi Modeler

Esta ferramenta [20] foi usada para a criação dos Business Process Model and Notation (BPMN). BPMN é uma notação da metodologia de gestão de processos de negócio e trata-se de uma série de ícones padrões para o desenho de processos, o que facilita o entendimento por parte do utilizador. A modelagem é uma etapa importante da automação pois é nela

que os processos são descobertos, desenhados e que pode ser feita alguma alteração no percurso do processo visando a sua otimização.



Figura 2.13: Bizagi Modeler - Plataforma de modelação de processos

2.2.5 Postman

Plataforma [21] de construção e testes de APIs. Esta ferramenta foi usada para testes de segurança.



Figura 2.14: Postman - Plataforma de construção e testes de APIs

2.2.6 Scribe

Plataforma [22] que gera guias visuais como por exemplo manuais de utilização de aplicações. Esta ferramenta foi utilizada na criação dos manuais de utilização.



Figura 2.15: Scribe - Plataforma que gera manuais de utilização

2.2.7 Google LightHouse

Plataforma [23] é uma ferramenta que mede a qualidade de páginas da web.



Figura 2.16: LightHouse - Ferramenta de medição de qualidade de páginas web

2.2.8 JMeter

Plataforma [24] desenhada para fazer testes à disponibilidade, escalabilidade e tempo de resposta de uma aplicação. Esta ferramenta foi utilizada nos testes de carga.



Figura 2.17: JMeter - Plataforma de desenhada para testar, testes de carga

2.3 Agile Scrum

O Agile [25] é uma metodologia de projeto baseada na colaboração e auto-organização de uma equipa. Ao contrário de metodologias como Waterfall, esta metodologia é iterativa, ou seja, o desenvolvimento, testes e aceitação dos desenvolvimento acontece na mesma fase do projeto, garantindo uma maior pro-atividade e colaboração entre toda a equipa, a existência de uma equipa mais organizada, reduzida e uma maior resposta por parte do cliente em relação ao que foi desenvolvido, assim a resolução de eventuais bugs que possam surgir é feita não no final do projeto, mas durante o seu desenvolvimento.

Scrum é uma framework que é usada para implementar Agile. Consiste na separação do trabalho em várias fases (épicos/releases) e em cada fase existe pelo menos um entrega do produto (subida a ambiente produtivo), em que estas fases estão ainda divididas em pequenas porções, normalmente entre uma a três semanas (Sprints), no qual é planeado antes do seu começo, qual a carga de trabalho a ser desenvolvido durante esse tempo. No final é feita uma demo ao cliente e uma retrospectiva sobre o Sprint corrente. Na retrospectiva são discutida as dificuldades encontradas pela a equipa, se as funcionalidades a desenvolver foram bem distribuídas, as funcionalidades que não foram desenvolvidas e se devem ou não migrar para o próximo Sprint, erros encontrados na aplicação, entre outros fatores importantes e que possam impactar o projeto.

Foi escolhido Agile Scrum por ser uma metodologia que auxilia o projeto a ser mais organizado, cooperativo e com entrega de um produto final mais rápido (ver figura com ciclo de vida de Agile Scrum 2.18). Este projeto foi organizado, a nível de épicos e Sprints da seguinte maneira:

- Cada épico está dividido em vários Sprints de duas semanas.
- Cada Sprint contém várias histórias (requisitos) cada uma com pontos (tempo estimado para a finalização da mesma).
- No final de cada Sprint é feita uma demo ao cliente e ao orientador, de forma a verificar se as histórias foram bem implementadas ou se precisam de ajustes.
- Depois da demo dá-se o fecho do Sprint a decorrer e o início do seguinte.

"Muitos problemas que surgem ao longo do desenvolvimento de uma aplicação, podem ser evitados se a sua documentação for elaborada adequadamente."[26] Ter uma boa documentação e atualizada e, um bom planeamento e modelagem do produto adequada, torna não só mais fácil o começo da fase de desenvolvimento, como também acelera esta fase. A melhor forma de organizar esta fase, é definir os requisitos não funcionais, requisitos funcionais e respetivos modelos de processos (ver 3.2).

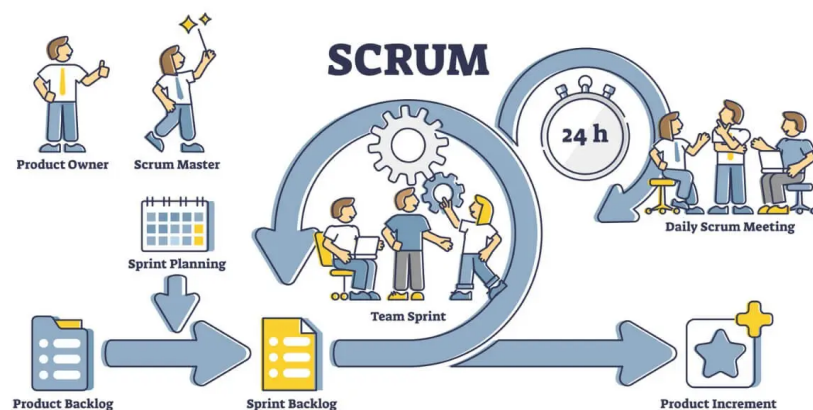


Figura 2.18: Ciclo de vida de Agile Scrum

2.4 Trabalho relacionado

Nesta secção são detalhadas as plataformas semelhantes à plataforma que foi desenvolvida no âmbito da tese, **Human Network**. No final é feita uma comparação entre estas plataformas e em que pontos a plataforma da tese se foca, que as outras não ou as melhorias que a mesma tem sobre as outras.

2.4.1 We help Ukraine

We help Ukraine[2] assumiu-se inicialmente como um ponto de encontro entre aqueles que precisam de ajuda e querem deixar a Ucrânia e aqueles que podem ajudar a prestar esse apoio em termos de alojamento, trabalho e medicamentos. No final a ideia estabilizou-se como uma plataforma capaz de agregar todas as outras ofertas de ajuda, transformando-se numa meta-plataforma integradora.

Esta plataforma nasce e completa-se com ideias e pessoas e vem da sociedade civil.

Se pessoas, empresas, instituições e as suas ofertas podem ser trazidas para cobrir as necessidades dos refugiados, então a plataforma cumpre a sua missão. Com duas dimensões fundamentais: a utilização de meios digitais e o recurso a voluntários que estão envolvidos, para além do seu trabalho, na gestão dos fluxos, da plataforma, do funcionamento, da comunicação e das pontes com todo o ecossistema com o qual interagem.

2.4.2 Free rice

Freerice [27] é um jogo de trivialidades educativo que a cada pergunta que se responde corretamente, 10 grãos de arroz são fornecidos para o Programa Alimentar Mundial (PAM).

Patrocinadores privados igualam as doações de grãos de arroz geradas em Freerice, gerando um pagamento financeiro ao PAM. Este dinheiro vai para o pote das "maiores necessidades", para apoiar as emergências em curso da organização. 100% de todos os fundos gerados através do Freerice vão para o Programa Alimentar Mundial. A Freerice não ganha nem guarda nenhum do dinheiro que angaria.

Desde 2010, a Freerice já angariou mais de 214 mil milhões de grãos de arroz para pessoas necessitadas.

2.4.3 Share the meal

ShareTheMeal [28] é uma iniciativa do Programa Alimentar Mundial das Nações Unidas (PAM) - a maior agência humanitária do mundo que combate a fome. Todos os anos, o PAM chega a mais de 86 milhões de pessoas com assistência alimentar em cerca de 83 países. O PAM é financiado a 100% por voluntários.

Os donativos da aplicação ShareTheMeal apoiam várias operações do PAM, que vão desde a construção de resiliência e programas de alimentação escolar, até ao fornecimento de assistência alimentar em situações de emergência.

2.4.4 Charity Miles

Charity Miles [29] é uma aplicação para iPhone e Android que o ajuda a transformar todos os quilómetros que percorre, corre e anda de bicicleta em dinheiro para caridade. A aplicação utiliza o pedómetro interno do telefone e o GPS para rastrear a sua distância.

2.4.5 Human Network

Existem muitas pessoas que não doam para caridade por não confiarem nas associações ou por simplesmente não saberem como o fazer. Existem estudos que comprovaram que a existência de uma plataforma digital aumenta o número de doações.

“Ao determinar os caminhos que sustentam os motivos dos dadores de sangue colaborativos, os resultados deste estudo fornecem forte suporte para a utilidade do modelo [TPB](#) no contexto do uso da plataforma digital e da doação de sangue.” [30]

Human Network é uma plataforma que promove projetos de cariz social, incentivando atividades vitais para o crescimento da nossa sociedade; esta também atua como um acelerador na conclusão de tarefas, facilitando a comunicação e conexão entre associações e empresas ou indivíduos.

Existem vários fatores que separam as plataformas acima descritas da **Human Network**:

- A nível de países abrangidos e tipos de projetos, enquanto que aplicações como We Help Ukrain que são limitadas a um país ou zona e, tipos de ajuda, **Human Network** permitirá a criação de qualquer tipo de projetos e é a nível mundial.
- A nível de tipos de doações, enquanto que aplicações como Share the meal ou Free Rice que são focados somente em distribuição de alimentos, **Human Network** não terá limitação, cada associação pode criar vários projetos, focados em vários tipos de doações, desde distribuição de alimentos, ajuda monetária, abrigo, emprego, educação entre outras.

2.5 Sumário

Neste capítulo foi feita uma comparação entre várias tecnologias de desenvolvimento Web e Outsystems e o motivo da escolha da plataforma Outsystems como principal tecnologia de desenvolvimento do projeto, apresentou-se a metodologia de projeto escolhida (Agile Scrum) e uma comparação entre aplicações com finalidades parecidas à **Human Network** e onde esta sobressai em relação às outras.

Todo o estudo feito para este capítulo ajudou no desenvolvimento e organização do projeto, tal como o desenvolvimento do capítulo seguinte.

METODOLOGIA DE DESENVOLVIMENTO E REQUISITOS

Nesta capítulo serão apresentadas as várias fases do projeto, levantamento de requisitos, uma descrição da arquitetura do sistema, o modelo de dados, integrações e o plano de trabalho.

3.1 Visão geral do desenvolvimento do projeto

Nesta secção é apresentada a organização do projeto, onde cada fase do projeto (épico) é detalhado. Um épico é um conjunto de Sprints.

Foi-nos fornecido um documento com alguns requisitos e em conjunto com o orientador, este documento foi analisado e desta análise nasceram novos requisitos e melhorou-se alguns dos requisitos originais, de forma a tornar a plataforma mais rica e com mais ferramentas.

De forma a organizar estes requisitos, foi criado um projeto utilizando metodologia Agile Scrum no JIRA. Este projeto está dividido em cinco épicos:

1. **Levantamento de requisitos:** Definição de requisitos (funcionais e não-funcionais) e organização dos mesmos num projeto Agile Scrum no JIRA. Para cada requisito foi criada uma história do utilizador (User Story) que especifica o essencial do requisito, quem vai usufruir desse requisito (um utilizador, companhia, associação, um utilizador com algum perfil específico) e como vai usufruir desse requisito. Para cada história foi também definida a sua prioridade tanto no Sprint como no Backlog do projeto e uma estimativa em horas de desenvolvimento da mesma. No final os requisitos foram ordenados por prioridade e foram divididos pelas duas fases de desenvolvimento.
2. **Desenho arquitetural e modelo de dados:** Definição do modelo de dados (passando de um modelo abstrato, para um modelo de Entidade/Relação e no final as otimizações feitas), arquitetura da aplicação (quantos e que tipo de módulos serão criados, qual a melhor organização dos mesmos), integrações e APIs.

3. **Desenvolvimento:** Desenvolvimento inicial da plataforma proposta no primeiro épico, ou seja, os requisitos mais prioritários. A prioridade dos requisitos foi definida pelo Product Owner (PO) do projeto.
4. **MVP:** Término dos desenvolvimentos necessários para subir a aplicação a ambiente produtivo (MVP). Na primeira fase do projeto foram definidos os requisitos que faziam parte do (MVP).
5. **Controlo de qualidade:** Ajustes necessários ao produto final, desenvolvimento de requisitos que não foram concluídos no épico anterior, correção de bugs, subida a ambiente produtivo e plano de ação futuro (atualização da documentação do projeto de modo a auxiliar desenvolvimentos futuros).

3.2 Definição de requisitos

Nesta secção é apresentada a definição dos requisitos não-funcionais, funcionais, os modelos de processos e as USs resultantes.

3.2.1 Requisitos funcionais

Requisitos funcionais definem funcionalidades de um sistema a nível de tarefas e serviços. "São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações." [31]

Para este sistema foram definidos os seguintes requisitos (cada requisito contém uma explicação, modelo de processos e as USs geradas a partir dos processos):

Registo e gestão de Utilizadores

Criação de utilizadores na aplicação (através de um ecrã de registo 3.1) e gestão dos mesmos (através de um ecrã de perfil 3.2).

Da avaliação do requisito foi criado um modelo de processos BPMN apresentado na figura 3.3.

A partir do modelo de processos, foi criada a US com identificador HN-44 e com a descrição "As an User I want to be able to register in the application".

Sistema Login

Ao sistema de login gerado pela plataforma Outsystems, decidiu-se adicionar camada extra de lógica e segurança. Esta camada extra é definida por:

- Possibilidade de fazer login através do sistema de Single Sign-on da Google.
- Possibilidade de adicionar uma autenticação de duas etapas a uma conta. Usou-se o Authenticato da Microsoft.

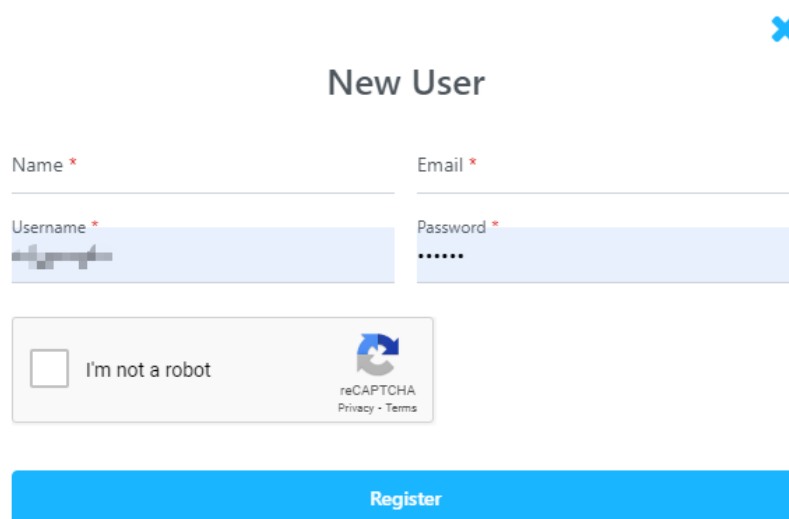


Figura 3.1: Ecrã de registo de utilizador e Sistema reCAPTCHA

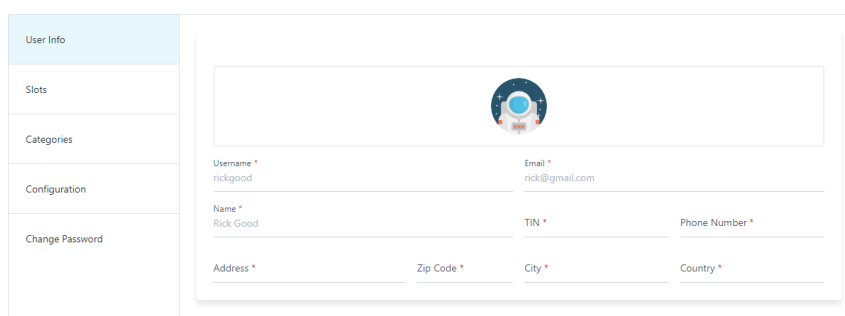


Figura 3.2: Ecrã de perfil de utilizador

Ecrã de login apresentado na figura 3.4.

Da avaliação do requisito foi criado um modelo de processos BPMN apresentado na figura 3.5.

A partir do modelo de processos, foi criada a US com identificador HN-2 e com a descrição “As a User I want to be able to Login in the application”.

Sistema reCaptcha

De forma a proteger a aplicação contra robos, foi adicionado ao sistema de registo e de login, o sistema reCaptcha (processos apresentados no modelo de login e registo 3.1).

A partir do modelo de processos, foi criada a seguinte US:

A partir do modelo de processos, foi criada a US com identificador HN-69 e com a descrição “As a Developer I want to guarantee that only real users register and login into the application”.

Gestão de Patrocinadores

Gestão de patrocinadores por parte de utilizadores com perfil de Admin ou Manager.

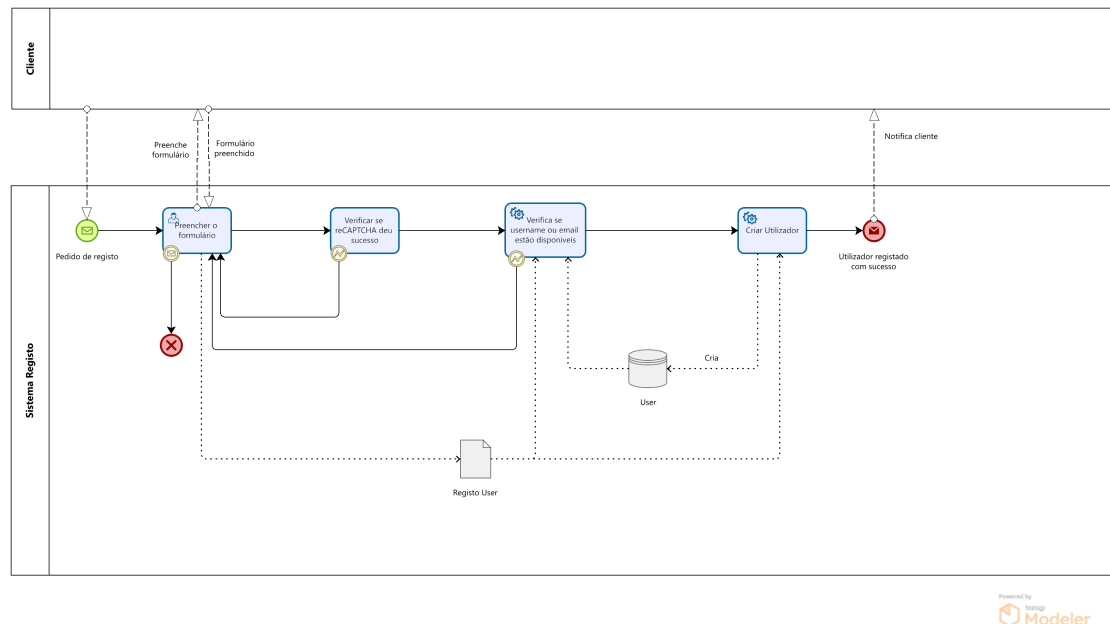


Figura 3.3: Modelo de Processos - Registo de utilizador

The screenshot shows the login interface for 'Human Network'. It features a logo at the top, a 'Sign in with Google' button, and input fields for 'Username' and 'Password'. There is a 'Remember me' checkbox and a 'Forgot password?' link. A blue 'Login' button is at the bottom.

Figura 3.4: Ecrã de Login

Esta funcionalidade é uma funcionalidade da aplicação Backoffice.

Da avaliação do requisito foi criado um modelo de processos BPMN apresentado na figura 3.6.

A partir do modelo de processos, foi criada a US com identificador HN-47 e com a descrição “As a User with Backoffice permissions I want to be able to create and manage Sponsors”.

3.2. DEFINIÇÃO DE REQUISITOS

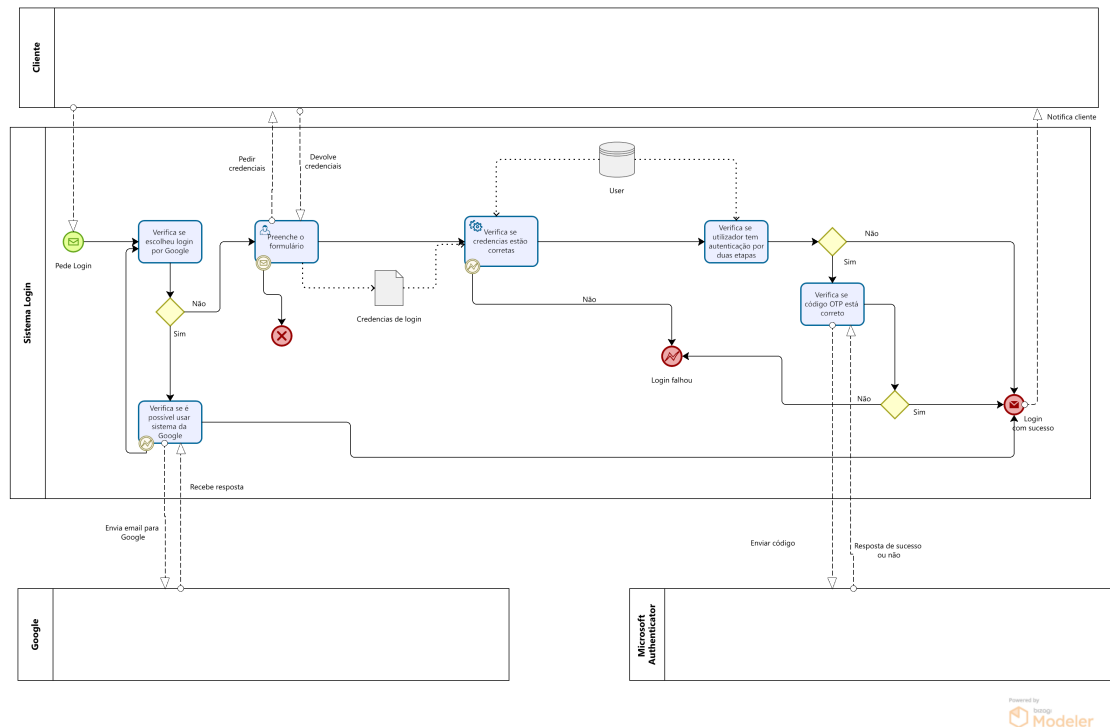


Figura 3.5: Modelo de Processos - Login

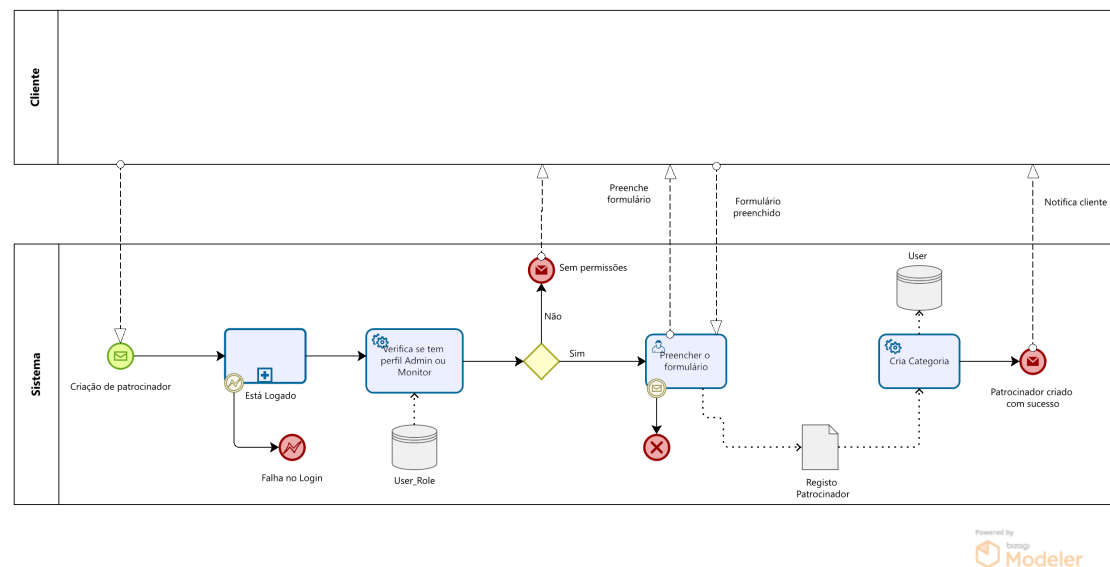


Figura 3.6: Modelo de Processos - Gestão de patrocinadores

Gestão de Categorias - Backoffice

Gestão de categorias por parte de utilizadores com perfil de **Admin** ou **Manager**. Esta

funcionalidade é uma funcionalidade da aplicação Backoffice.

Da avaliação do requisito foi criado um modelo de processos BPMN apresentado na figura 3.7.

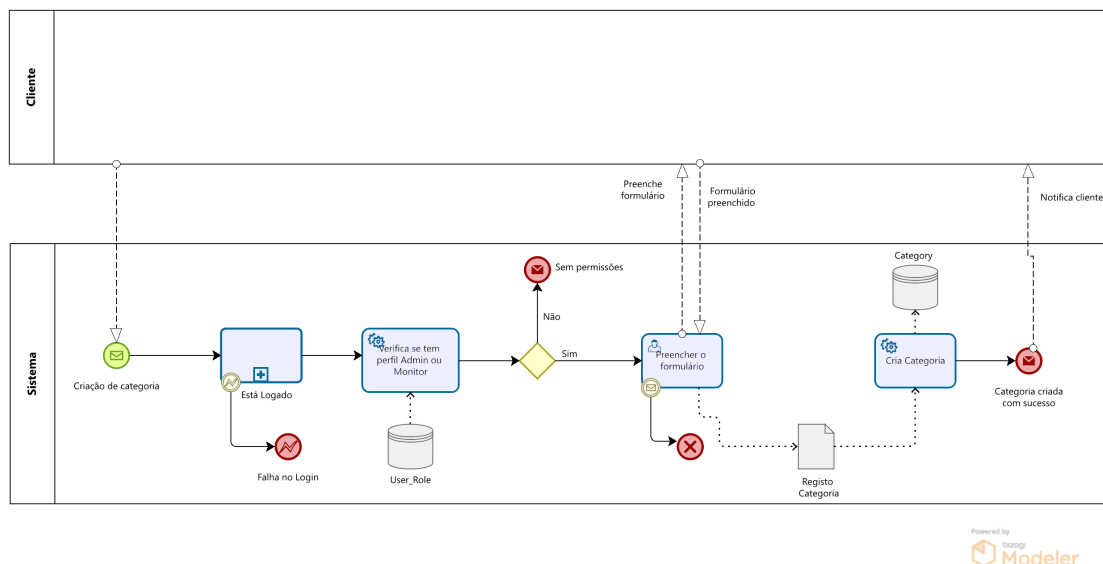


Figura 3.7: Modelo de Processos - Gestão de categorias

A partir do modelo de processos, foi criada a **US** com identificador **HN-3** e com a descrição “As a User with Backoffice permissions I want to be able to create and manage Categories”.

Gestão de Categorias do utilizador

Gestão das categorias escolhidas e favoritas de um utilizador. Esta funcionalidade serve para melhorar a pesquisa de projetos feita pelo o utilizador.

Da avaliação do requisito foi criado um modelo de processos BPMN apresentado na figura 3.8.

A partir do modelo de processos, foi criada a **US** com identificador **HN-53** e com a descrição “As a User I want to be able to choose my favorite categories so that the projects suggested to me, are accordingly with my preferences”.

Gestão de Redes

Gestão de redes por parte de utilizadores (através de um ecrã de registo de redes) e a sua gestão (através de um ecrã com todas as configurações, projetos e colaboradores da rede).

Redes são companhias ou associações. Sendo que a nível mais técnico existe pouca diferença entre companhias e associações, decidiu-se juntar-se ambos no mesmo conceito lógico e tabela na base de dados (**Network**)

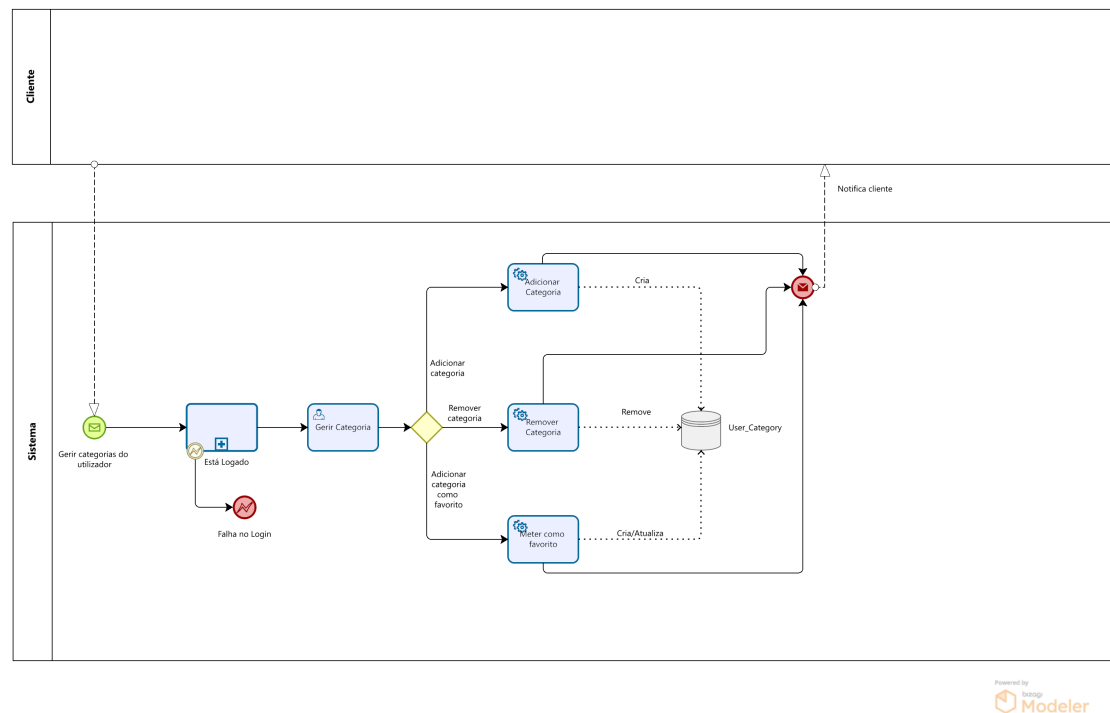


Figura 3.8: Modelo de Processos - Gestão de categorias de utilizadores

Da avaliação do requisito foi criado um modelo de processos BPMN apresentado na figura 3.9.

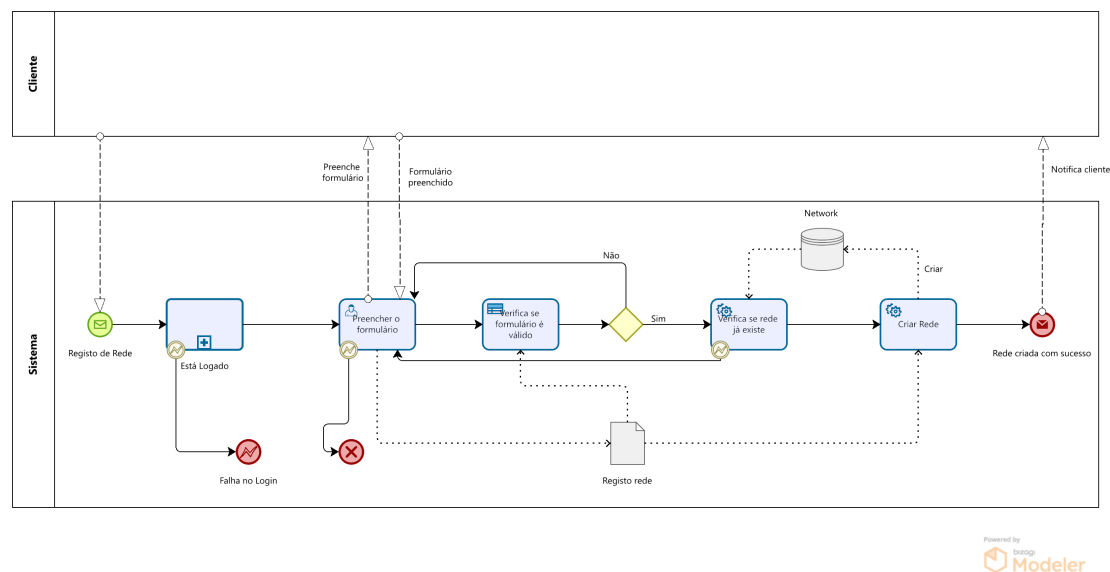


Figura 3.9: Modelo de Processos - Gestão de redes

A partir do modelo de processos, foi criada a **US** com identificador **HN-46** e com a descrição “As a User I want to be able to create a Network”.

Gestão de colaboradores de Redes

Ao se criar uma rede, é possível convidar utilizadores da aplicação ou pessoas que ainda não sejam utilizadores (é usado o email para esta gestão, caso exista um utilizador com aquele email, este é convidado pela aplicação, senão um email é enviado para a pessoa). É também possível definir o perfil do colaborador (Viewer ou Admin, como explicado em detalhe na sub-secção 4.1.1).

Apesar da plataforma **Human Network**, oferecer a gestão de colaboradores dentro de uma rede, fez sentido ter serviços que possibilitassem a gestão acelerada dos colaboradores de uma rede. Desta forma, decidiu-se criar vários serviços que possam ser facilmente integrados pelas redes, no seu processo de gestão de utilizadores. O serviços criados foram:

- Um serviço de criação de um colaborador, onde não só é criado um utilizador para o colaborador, mas como também é automaticamente associado à rede.
- Um serviço de remoção de um colaborador, onde é removido não só a associação entre o utilizador e a rede, mas também é apagado o utilizador da aplicação.
- Um serviço de importe total de utilizadores, onde são importados em massa colaboradores.

Disponibilizou-se também a possibilidade de importar os utilizadores a partir de um ficheiro excel. Para facilitar o importe através de excel, é possível extrair um ficheiro modelo (ver figura 3.10).

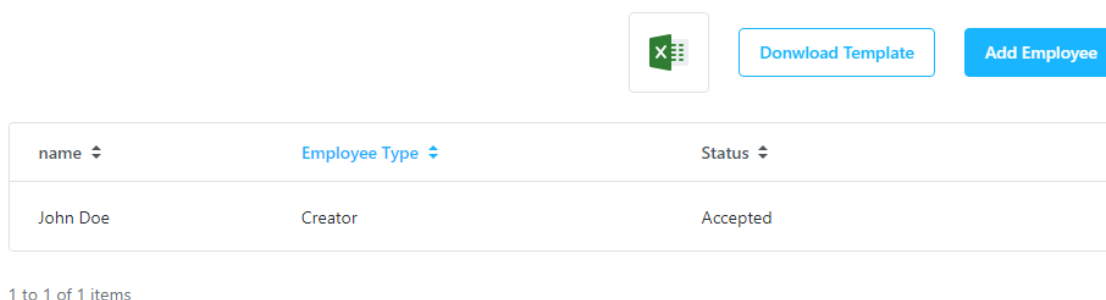


Figura 3.10: Gestão de Colaboradores

Da avaliação do requisito foram criados dois modelos de processos **BPMN** apresentados nas figuras 3.11 e 3.12.

A partir do modelo de processos, foi criada a **US** com identificador **HN-24** e com a descrição “As a Network’s admin I want to be able to invite an employees to the app”.

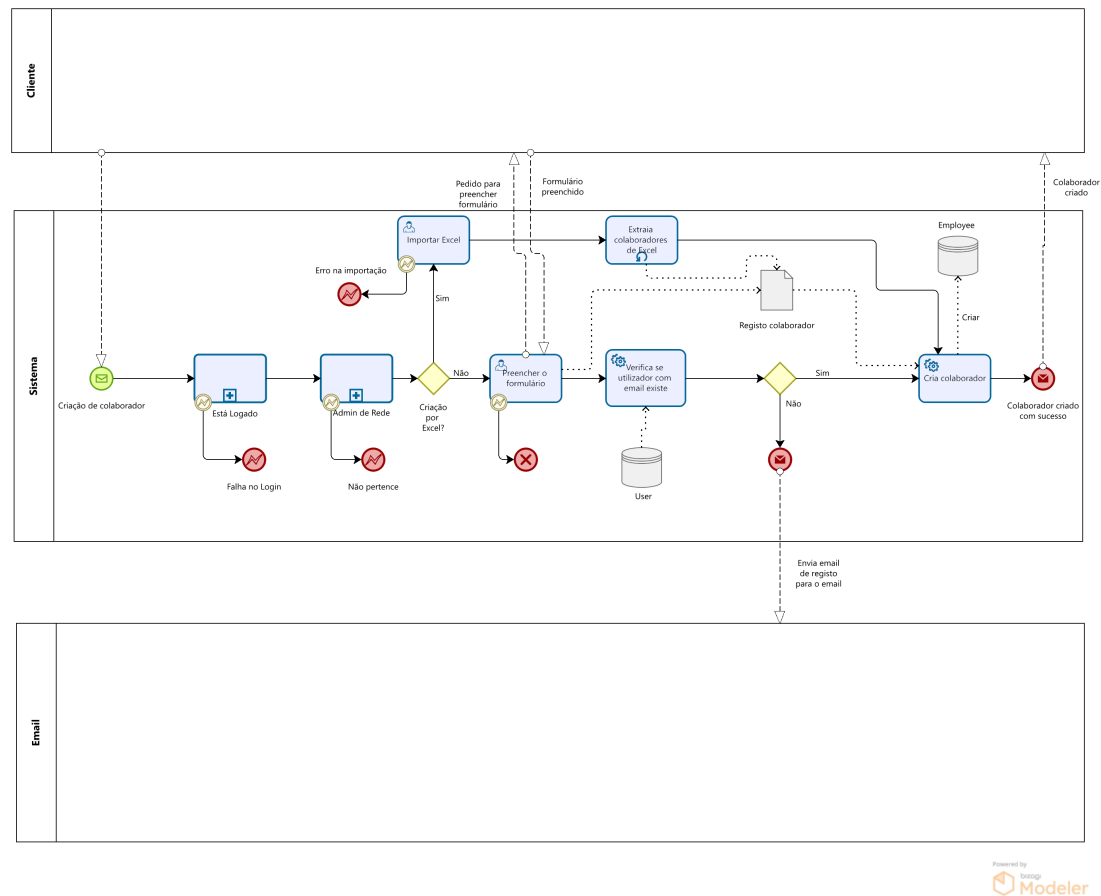


Figura 3.11: Modelo de Processos - Gestão de colaboradores

Gestão de Projetos

Gestão de projetos por parte de utilizadores com perfil de administrador de uma associação. Somente associações podem criar projetos.

Na altura de criação de um projeto é necessário escolher o tipo de projeto, projeto de um dia, projeto "one-shot" (projeto que ocorre todos os dias durante um certo período) ou um projeto com múltiplos eventos e é necessário definir a data de início e de fim do projeto. Também é definido se o registo por parte de utilizadores é automático ou se é necessário ser aceite por um administrador da associação (ver figura 3.13).

Depois de criado, o projeto pode ser editado e três novas abas aparecem:

- **Eventos:** Que contém a gestão dos eventos (ver figura 3.14).
- **Submissões:** Que contém todas as submissões pendentes feitas por utilizadores para os eventos (ver figura 3.15).
- **FAQs:** Que contém frequently asked questions (FAQ) (ver figura 3.16).

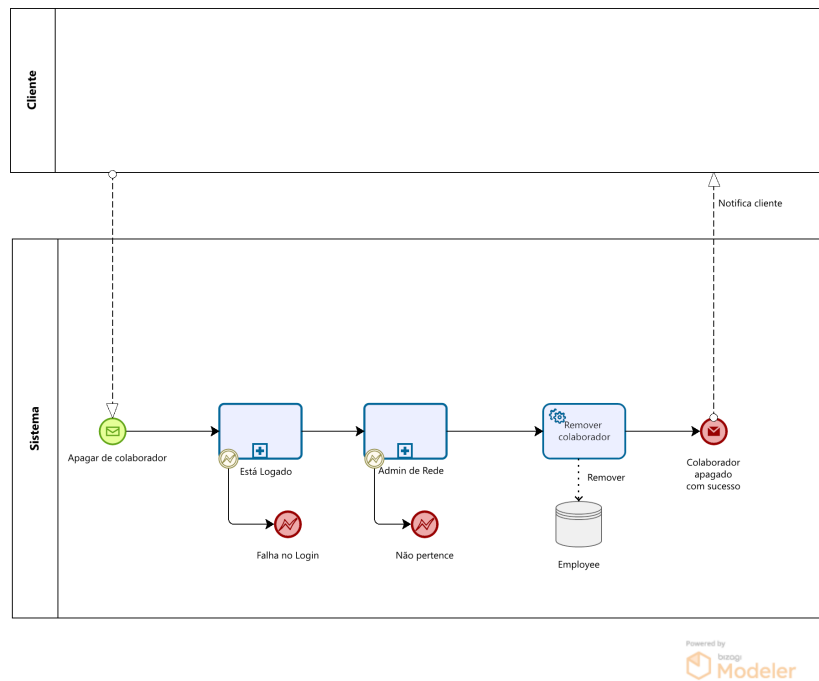


Figura 3.12: Modelo de Processos - Remover colaborador

Figura 3.13: Ecrã de edição de projetos

Da avaliação do requisito foram criados dois modelos de processos **BPMN** apresentados nas figuras 3.17 e 3.18.

A partir do modelo de processos, foi criada a **US** com identificador **HN-5** e com a descrição “As an Association’s admin I want to be able to create and manage Projects”.



Figura 3.14: Ecrã de edição de projeto - lista de eventos

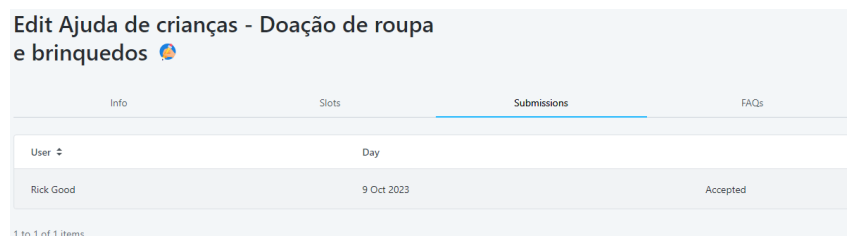


Figura 3.15: Ecrã de edição de projeto - lista de submissões

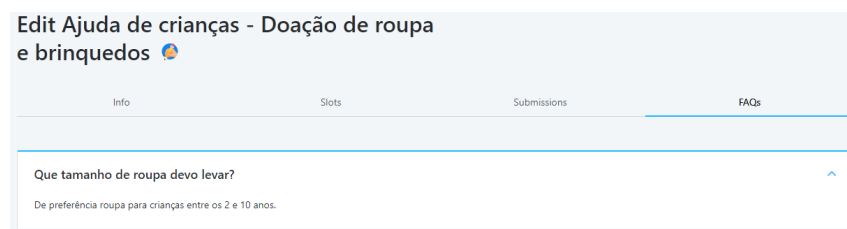


Figura 3.16: Ecrã de edição de projeto - lista de FAQs

Gestão de Categorias do projeto

Esta funcionalidade serve para melhorar a pesquisa de projetos feita por um utilizador. Um projeto pode ter várias categorias mas só pode ter uma como "Principal"(Categoria que aparece na listagem de projetos).

Da avaliação do requisito foi criado um modelo de processos BPMN apresentado na figura 3.19.

Sistema de procura de Projetos

Existem vários tipos de pesquisa sobre projetos, depende do propósito da pesquisa e que tipo de utilizador está a pesquisar (se é um colaborador de uma companhia, de uma associação ou não inscrita em nenhuma rede, o tipo de pesquisa poderá ser diferente).

Existem quatro tipos de pesquisa de projetos:

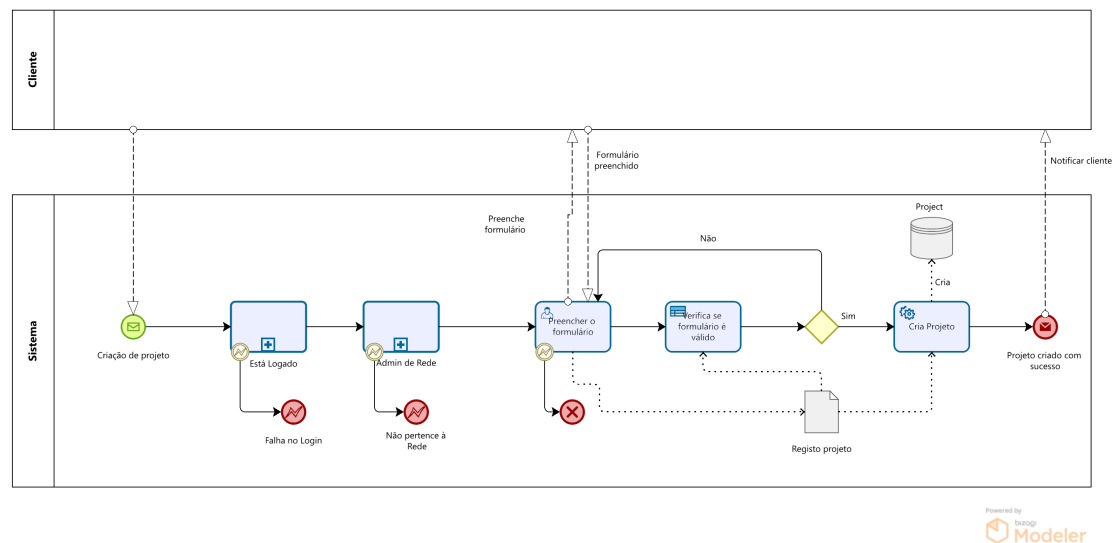


Figura 3.17: Modelo de Processos - Gestão de projetos

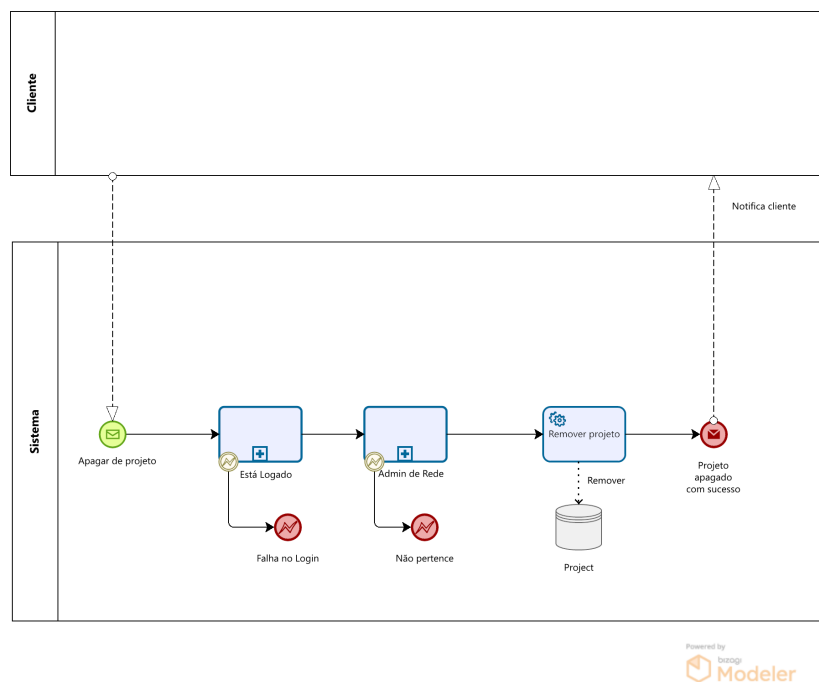


Figura 3.18: Modelo de Processos - Remover projeto

- No ecrã com a lista de todos os projetos (ver figura 3.20), existem vários filtros que afetam a pesquisa, tais como um filtro de pesquisa por texto, um por escolha de avaliação, que vai de uma a cinco estrelas e um filtro por categorias. Nesta pesquisa, todos os utilizadores vão obter a mesma lista de projetos (ver pergunta relacionada

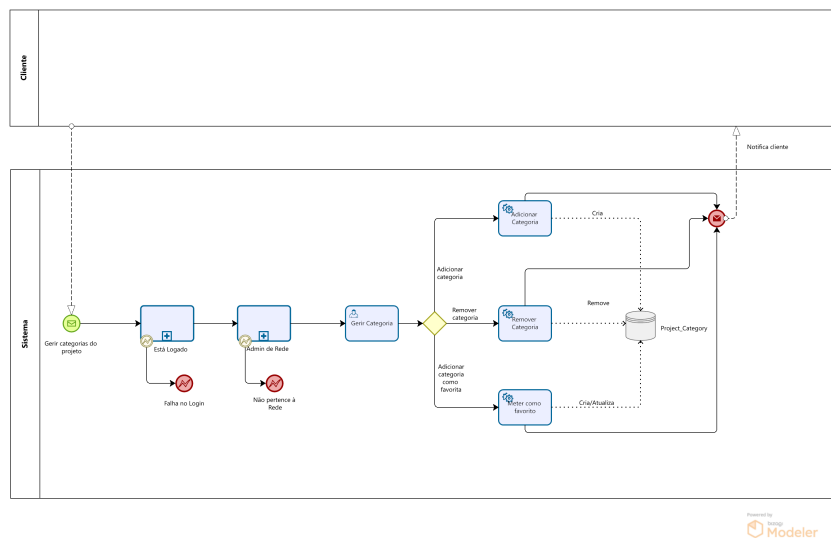


Figura 3.19: Modelo de Processos - Gestão de categorias de projeto

em anexo I).

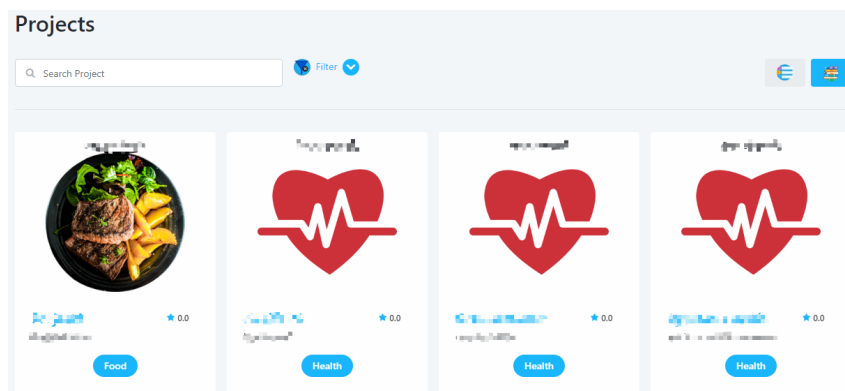


Figura 3.20: Ecrã de lista de projetos

- No ecrã com a lista de todos os projetos do utilizador (ver figura 3.21), existem vários filtros que afetam a pesquisa, tais como um filtro de pesquisa por texto e um filtro por categorias. Nesta pesquisa, são retornados os projetos em que a utilizador pertence (seja como membro da associação que criou o projeto, utilizador ou colaborador de uma companhia registado no projeto) e os projetos guardados como favoritos (o utilizador pode ou não estar registado no projeto) (ver pergunta relacionada em anexo I).
- No ecrã do detalhe de um projeto (ver figura 3.22) existe no final do ecrã uma lista de projetos sugeridos. Este sistema de sugestão de projetos têm em conta as categorias mais escolhidas do utilizador (projetos em que o utilizador mais se registou em eventos de uma certa categoria), tais como a categoria do projeto a ser visualizado.

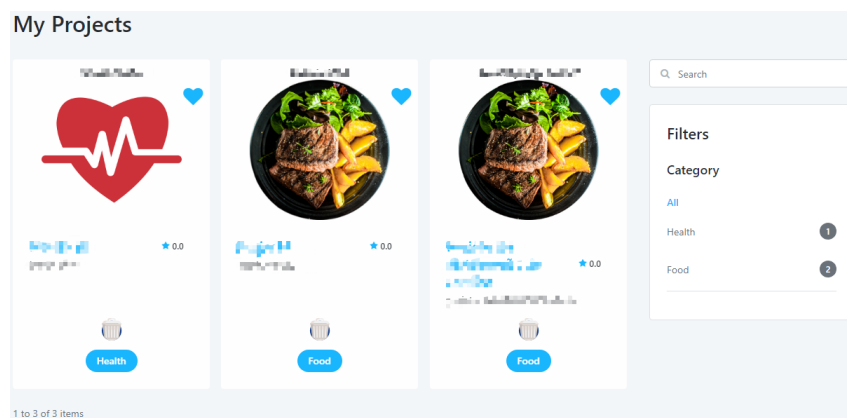


Figura 3.21: Ecrã de lista de projetos do utilizador

Também é usada a mesma lógica no ecrã de lista de projetos na aplicação mobile (ver figura 3.23).

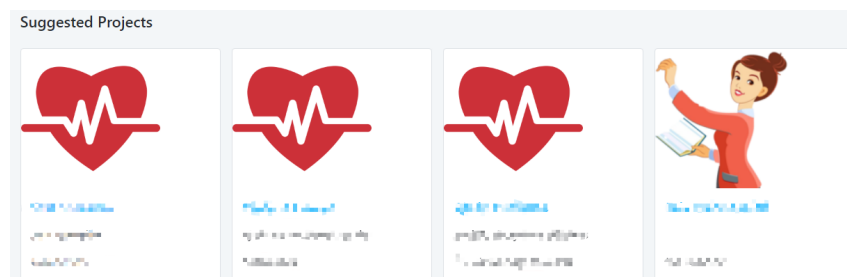


Figura 3.22: Ecrã de detalhe de projeto com uma lista de projetos sugeridos



Figura 3.23: Ecrã de lista de projetos sugeridos - aplicação mobile

As perguntas que populam estes ecrãs são as seguintes:

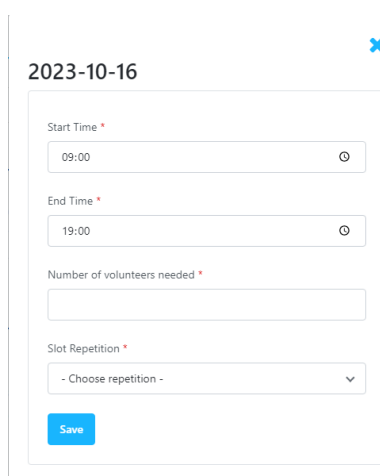
1. Se o ecrã for o de detalhe de projeto, então é executada uma pergunta focada na categoria do projeto (ver pergunta em anexo I).
2. Caso a pergunta apresentada em cima retorne uma lista vazia, poucos registos ou o ecrã seja o de lista de projetos na aplicação mobile, é executada a seguinte pergunta (ver pergunta em anexo I).
3. Caso a pergunta inicial seja executada, mas a segunda pergunta também tenha de ser executada, os valores de ambas perguntas são unidos num só resultado.

A partir do modelo de processos, foram criadas duas **USs** (ver tabela 3.1, HN-6 e HN-17).

Criação de eventos de Projetos

Criação e gestão de eventos num projetos por parte de utilizadores com perfil administrador numa associação. Estes eventos podem ser únicos ou ter repetição semanal, bi-semanal, mensal ou anual (projetos não únicos têm lógica de criação automática, estes eventos são criados até ao final do projeto).

Na aba de eventos, é possível criar um evento ao carregar numa data disponível no calendário (ver figura 3.24).



The screenshot shows a web form for creating an event on the date 2023-10-16. The form is titled '2023-10-16' and has a close button (X) in the top right corner. It contains the following fields:

- Start Time ***: A text input field with the value '09:00' and a clock icon on the right.
- End Time ***: A text input field with the value '19:00' and a clock icon on the right.
- Number of volunteers needed ***: A text input field.
- Slot Repetition ***: A dropdown menu with the option '- Choose repetition -'.
- Save**: A blue button at the bottom left of the form.

Figura 3.24: Ecrã de edição de projeto - criação de eventos

Da avaliação do requisito foram criados dois modelo de processos **BPMN** apresentados nas figuras 3.25 e 3.26.

A partir do modelo de processos, foi criada a **US** com identificador **HN-8** e com a descrição “As an Association’s admin I want to be able to create various slots for a Project”.

Inscrição em eventos por parte de Utilizadores

Um utilizador ao entrar no ecrã de detalhe de um projeto, consegue escolher a aba de eventos e candidatar-se a um evento dentro de uma lista (ver imagens 3.27).

Caso o projeto não seja de aceitação automática de registo de eventos, a lista de submissões é atualizada e fica possível um utilizador com perfil administrador aceitar ou rejeitar a submissão. Caso a submissão seja aceita, o evento aparece no calendário do utilizador, que se encontra no perfil do mesmo (ver figura 3.28).

Da avaliação do requisito foi criado um modelo de processos **BPMN** apresentado na figura 3.29.

A partir do modelo de processos, foi criada a **US** com identificador **HN-50** e com a descrição “As a User I want to be able to enroll a Slot”.

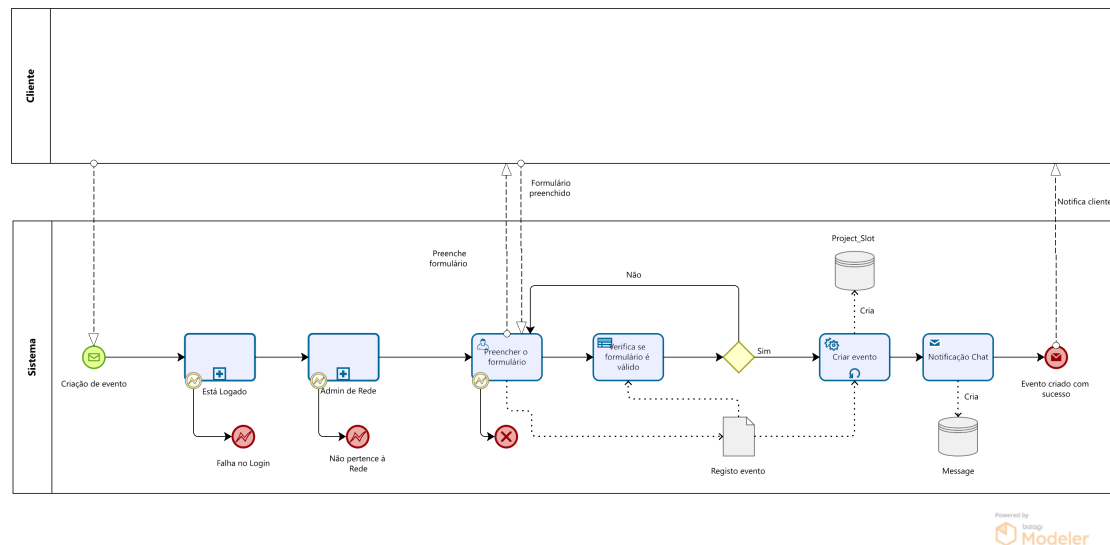


Figura 3.25: Modelo de Processos - Gestão de eventos de projetos

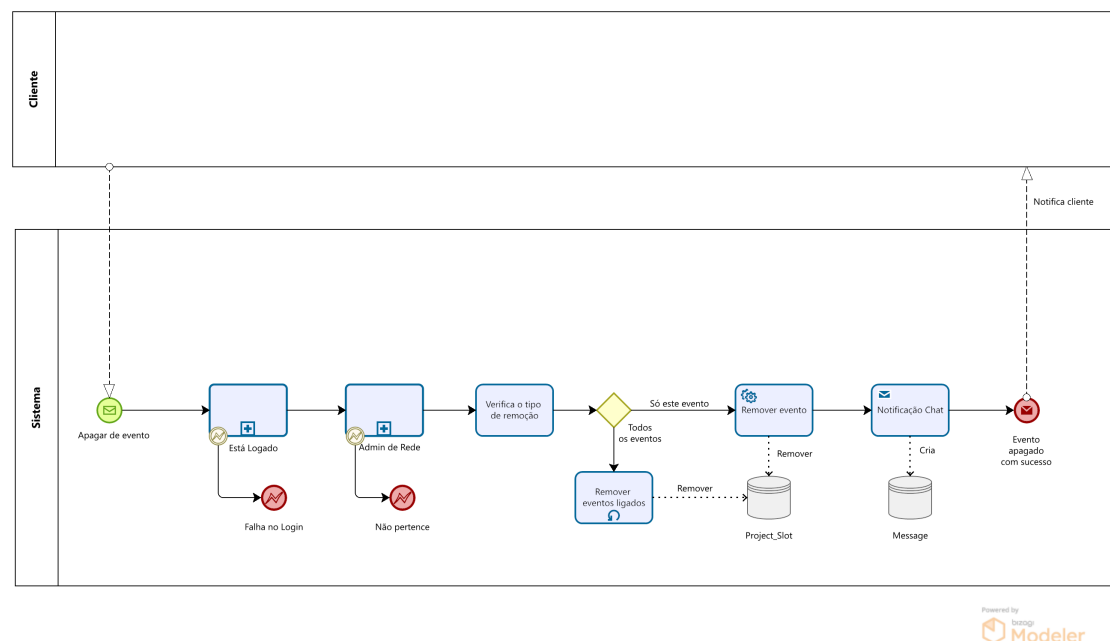


Figura 3.26: Modelo de Processos - Remover eventos de projetos

Prescrição em eventos por parte de Utilizadores

Um utilizador caso já não pretenda participar num evento, tem a possibilidade de prescrever a sua inscrição.

Da avaliação do requisito foi criado um modelo de processos BPMN apresentado na figura 3.30.

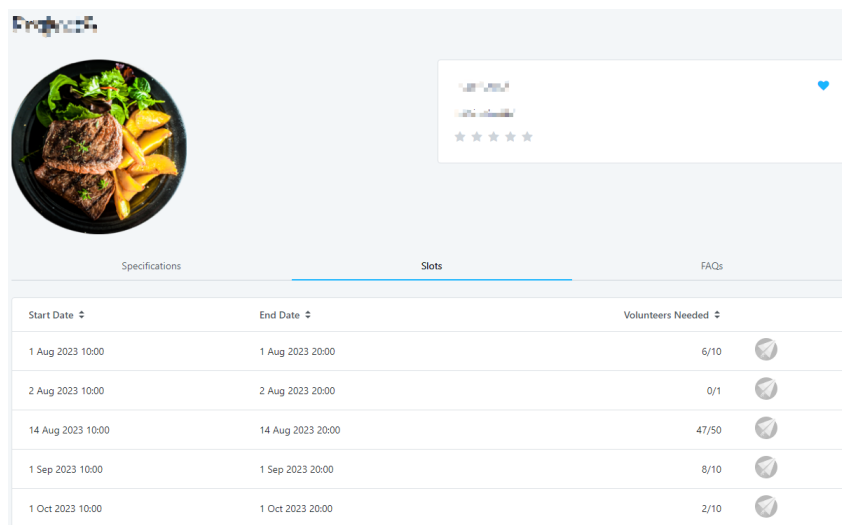


Figura 3.27: Ecrã de detalhe de projeto - inscrição num evento

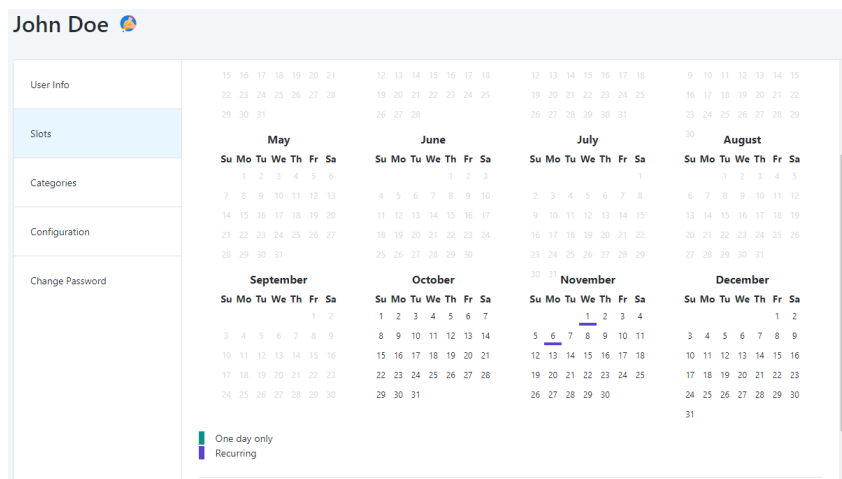


Figura 3.28: Ecrã de perfil de utilizador - calendário de eventos

Sistema de integração de Google Calendar

Apesar da facilidade de gestão de eventos oferecida pela aplicação **Human Network**, muitos utilizadores preferem ter a gestão dos seus eventos, num calendário, como o Google Calendar. Por este motivo, decidiu-se integrar o gestão de eventos com o Google Calendar, onde cada vez que uma submissão de evento por parte de um utilizador, é aceite ou o utilizador prescreve um evento, caso esteja definido no seu perfil, este evento é adicionado ou removido do Google Calendar do utilizador.

A partir do modelo de processos, foi criada a **US** com identificador **HN-42** e com a descrição “As a User I want to be able to login using Google’s Single Sign-On”.

Criação de FAQs em Projetos

Apesar da aplicação oferecer um sistema de mensagens, sentiu-se a necessidade de possibilitar as associações de adicionar respostas/informação sobre os tópicos mais comuns do projeto (informação adicional). Por isso, criou-se uma aba de **FAQs** (ver figura 3.16)

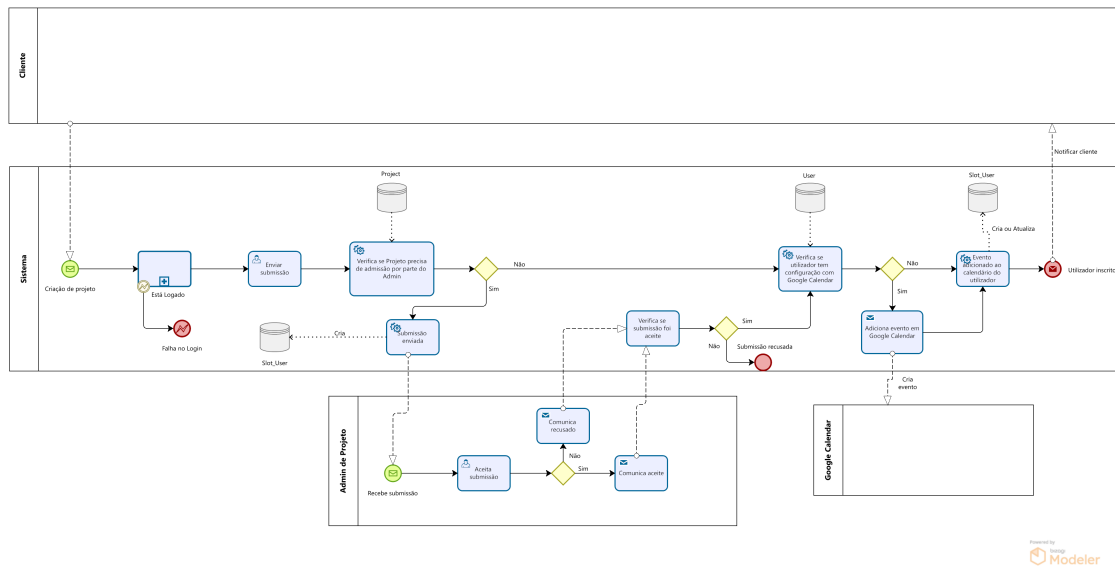


Figura 3.29: Modelo de Processos - Inscrição em eventos por utilizadores

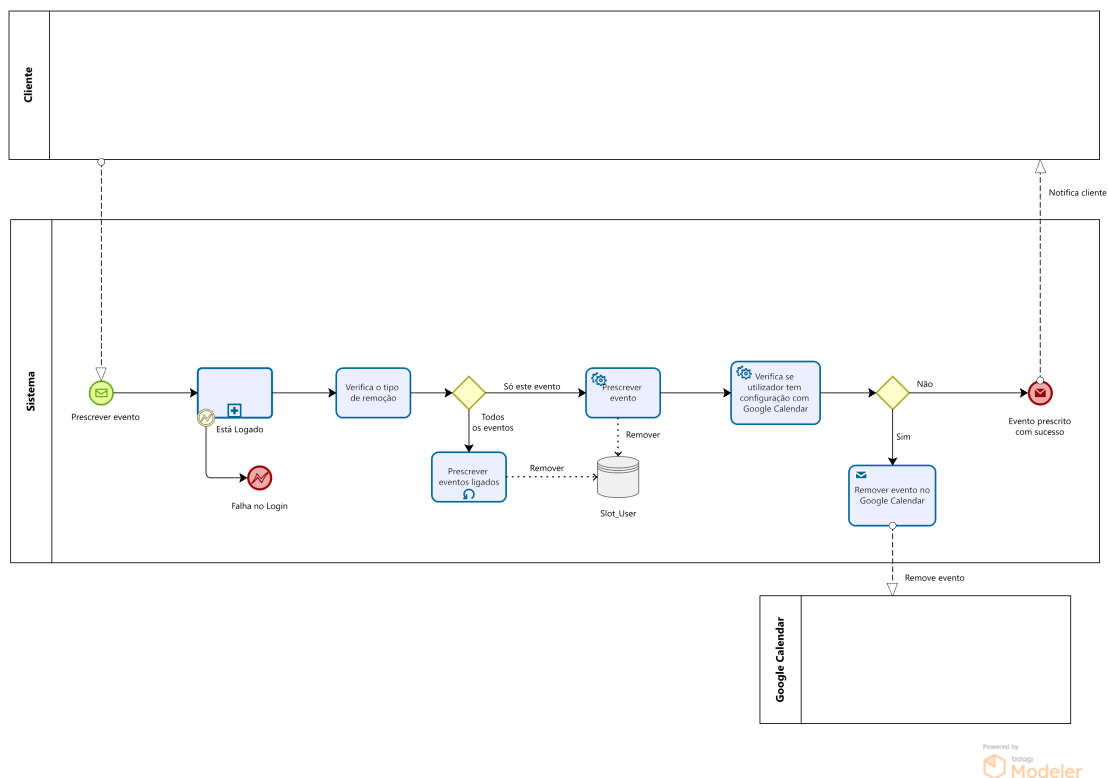


Figura 3.30: Modelo de Processos - Prescrição em eventos por utilizadores

gerida por utilizadores com perfil de administrador numa associação e de fácil pesquisa por parte de utilizadores registado no projeto.

Da avaliação do requisito foi criado um modelo de processos BPMN apresentado na figura 3.31.

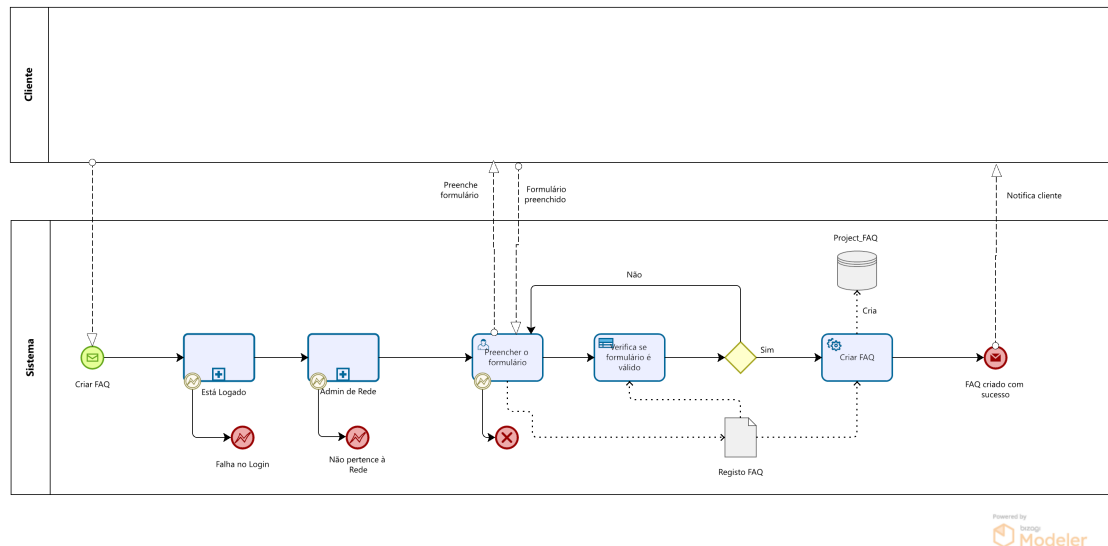


Figura 3.31: Modelo de Processos - Gestão de FAQs de projetos

A partir do modelo de processos, foi criada a US com identificador **HN-62** e com a descrição “As an Association’s admin I want to be able to create a FAQs section”.

Sistema de mensagens - Chat

Quando um utilizador fica registado num projeto (um dos eventos submetidos é aceite), é adicionado à conversa do projeto. Nesta conversa participam todos os utilizadores registados no projetos e todos os utilizadores, com perfil administrador, da associação que criou o projeto (ver figura 3.32).

Este sistema pode ser acedido a partir de qualquer ecrã, mas caso o utilizador esteja no ecrã de detalhe de um projeto, o sistema de mensagens abrirá nas mensagens desse projeto. não só são mostradas as mensagens trocadas por todos os membros desta conversa, mas também são registadas todas as grandes mudanças do projeto (criação, edição ou remoção de eventos, mudança de informação do projeto, etc).

De forma a melhorar o desempenho deste sistema e disponibilidade, permitindo que quando o utilizador abre o sistema de mensagens, não tenha de estar à espera que as mensagens carreguem, a query de pesquisa de mensagens é executada ao nível do layout e não do ecrã (querendo dizer que mesmo que o utilizador navegue para outro ecrã, a última conversa carregada não necessita de carregar outra vez, correr a query outra vez) e, em paralelo (não impacta o carregamento do ecrã ou das outras perguntas).

Da avaliação do requisito foi criado um modelo de processos BPMN apresentado na figura 3.33.

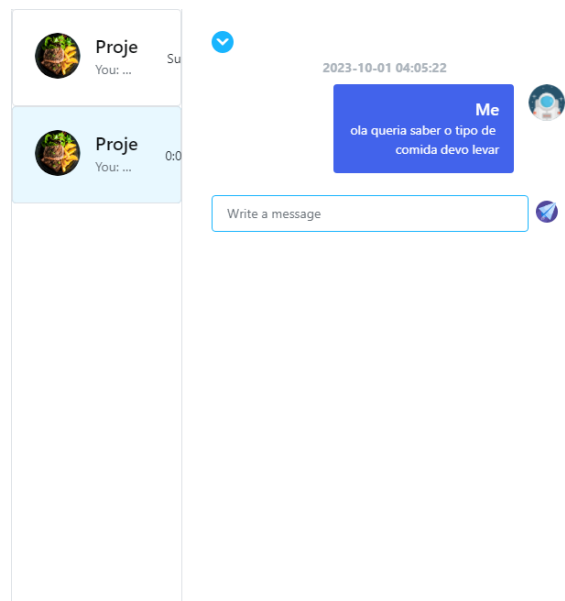


Figura 3.32: Sistema de mensagens - Chat

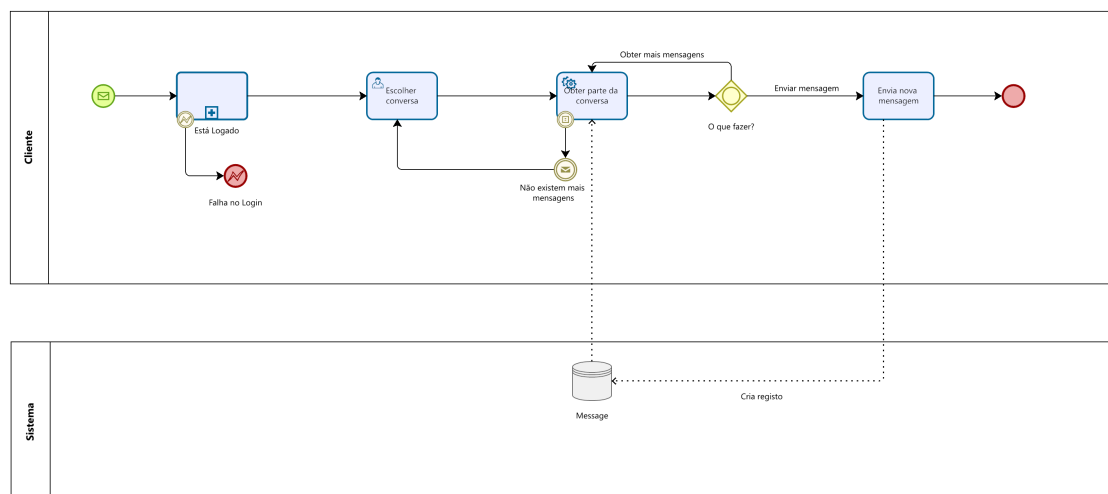


Figura 3.33: Modelo de Processos - Sistema de mensagens

A partir do modelo de processos, foi criada a **US** com identificador **HN-13** e com a descrição “As a User I want to have a chat with other Users, in order to check if the project meets my requirements and, for Associations, if the Users applying for the Project, has a good understanding of the Project at hands”.

Sistema de notificação de mensagens

Foi criado um sistema de atualização e notificação de mensagem através do método `setInterval()` de JavaScript (ver código 3.1). A cada 10 segundos, uma query que verifica o

identificador da última mensagem enviada na conversa, é executada. Caso o identificador devolvido seja diferente ao guardado localmente, a conversa é refrescada.

No início do projeto ponderou-se utilizar Google Firebase, mas depois de se analisar melhor o requisito decidiu-se seguir com esta abordagem.

Listagem 3.1: Sistema de atualização de mensagens

```
1 $parameters.Interval = window.setInterval(function(){actions.RefreshChat()}, 10000)
```

A partir do modelo de processos, foi criada a **US** com identificador **HN-54** e com a descrição “As a User I want to be notified everytime I receive a new message”.

Sistema de mudança de linguagem

Um dos requisitos mais importante para a plataforma **Human Network**, foi a possibilidade de mudar a linguagem de toda a aplicação, de forma a facilitar a sua utilização por pessoas de todas as nacionalidades. Por isso, foi criado um sistema de mudança de linguagem para várias linguas (ver figura 3.34), sistema este fácil de configurar por utilizadores com perfil de manager da aplicação, ou seja, trabalhadores da empresa KinetIT.

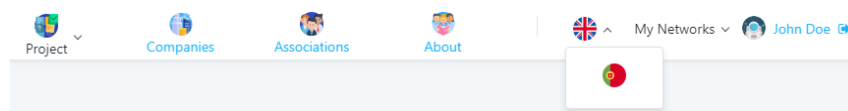


Figura 3.34: Sistema de mudança de linguagem

A partir do modelo de processos, foi criada a **US** com identificador **HN-12** e com a descrição “As a User I want to be able to change the overall language of the application”.

Crawler - Web Scraping

Criação de um sistema que extraía informação de vários websites. A funcionalidade e toda a gestão deste sistema é explicado na sub-secção 4.2.1.

Da avaliação do requisito foi criado um modelo de processos **BPMN** apresentado na figura 3.35.

A partir do modelo de processos, foi criada a **US** com identificador **HN-43** e com a descrição “As a Developer I want to create an web scaper, that searches in various Association’s websites”.

Aplicação Mobile

Foi requisitado a criação de uma aplicação mobile que mimicasse algumas das funcionalidades da aplicação web. A aplicação mobile contém as seguintes funcionalidades:

- Sistema de login de utilizadores (ver figura 3.36).
- Ecrã de procura de projetos (ver figura 3.37), onde são sugeridos projetos tendo em conta as suas preferências a nível de categorias (lógica explicada em 3.2.1).

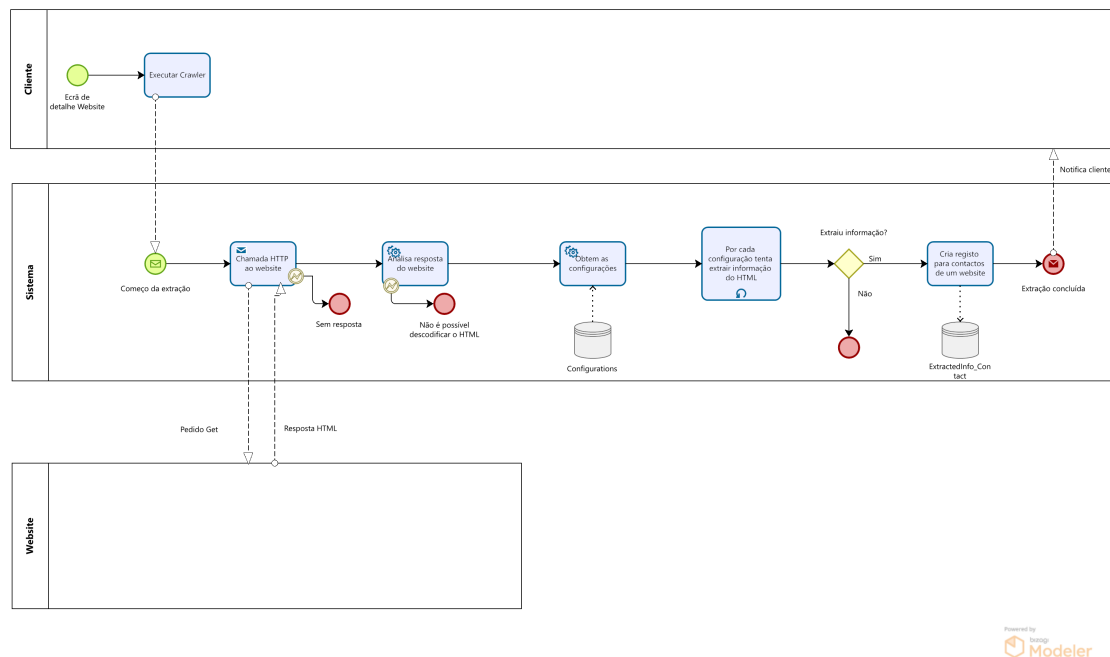


Figura 3.35: Modelo de Processos - Sistema de extração de informação (Crawler)

- Ecrã de perfil de utilizador (ver figura 3.38).
- Ecrã com um calendário de eventos (ver figura 3.39).
- Sistema de mensagens (ver figura 3.40).

As tabelas com todas as **USs** podem ser consultadas no final do capítulo, a seguir à secção **Sumário** (ver tabelas 3.1, 3.2 e 3.3).

3.2.2 Requisitos não-funcionais

Requisitos não-funcionais são requisitos que restringem os requisitos funcionais e estão relacionados com a utilização de uma aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, integridade e manutenção. "São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas." [31]

Todos estes requisitos foram tidos em conta no decorrer do desenvolvimento da aplicação **Human Network**.

- **Desempenho:** Foi garantido um bom desempenho da aplicação através da otimização, cache e teste de perguntas, de forma a garantir uma rápida resposta independente do número de dados existentes na base de dados, otimização dos ecrãs, garantindo que as boas práticas de React foram seguidas e que o ciclo de vida dos ecrãs foi

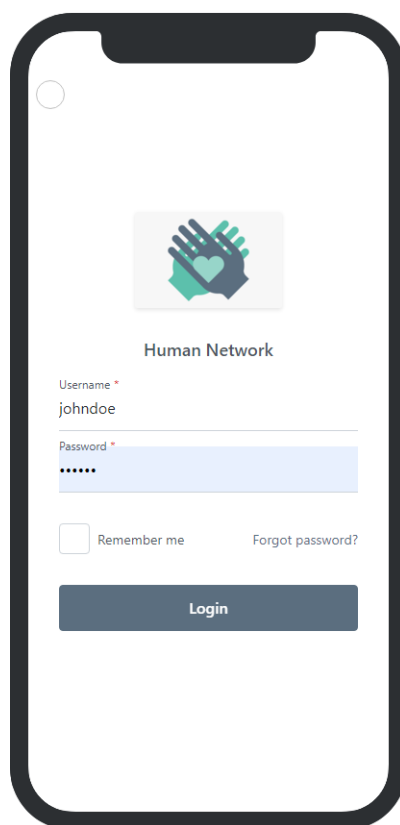


Figura 3.36: Aplicação mobile - ecrã de login

respeitado. Testes, problemas encontrados a nível de desempenho e as devidas resoluções, estão especificados em pormenor no capítulo 5.

- **Usabilidade:** De forma a garantir uma boa usabilidade, estudou-se as melhorias recomendadas por utilizadores à aplicação. Os utilizadores avaliaram a aplicação a nível de facilidade de aprendizagem, facilidade de encontrar e memorizar as funcionalidades (por exemplo, utilizando icons e imagens que sejam universalmente conhecidas, sem necessitar de recorrer a botões com rótulos), a redução de cliques e navegações desnecessárias entre ecrãs, garantindo que o utilizador não se cansa de listas infinitas e navegação constante entre ecrãs, entre outros.
- **Confiabilidade:** Foi garantido uma boa confiabilidade da aplicação através da criação de um sistema tolerante a erros, constante monitorização de erros e possíveis problemas de arquitetura, recorrendo a ferramentas da plataforma Outsystems, tais como AI Mentor Studio, Discovery e Service Center (mais detalhe sobre estas ferramentas em 2.2).
- **Segurança:** Um grande problema a nível de segurança nas aplicações, é a gestão dos seus utilizadores. Outsystems já garante esta segurança, sendo que a gestão de utilizadores, login e tokens é feita pela plataforma. Foi também feita a sanitização dos inputs de perguntas de forma a impossibilitar a injeção de perguntas, é sempre feita,

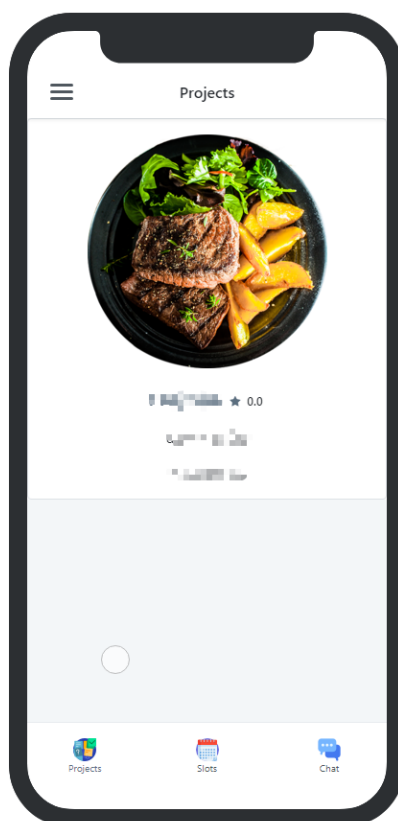


Figura 3.37: Aplicação mobile - ecrã com lista de projetos sugeridos

em qualquer funcionalidade, a validação de role e perfil, impedido que utilizadores não registados ou utilizadores sem perfil específico, possam aceder a ecrãs fora das suas permissões e que possam aceder e provocar funcionalidades sem as devidas permissões.

- **Disponibilidade:** Foi garantido uma boa disponibilidade da aplicação através da escalabilidade oferecida pela plataforma Outsystems, permitindo crescer horizontalmente (mais servidores) e verticalmente (melhores servidores) consoante o tráfego e a distribuição do mesmo através de um Load Balancer (sistema que distribui a carga entre várias servidores) [32].
- **Integridade:** Foi garantido uma boa integridade dos dados através de validações a nível de cliente e servidor para garantir uma maior segurança a nível de alteração ou destruição de dados.
- **Manutenção:** O desenvolvimento da aplicação foi feita de forma a ser escalável, organizada, fácil de atualizar para ferramentas/componentes mais recentes, e com constante monitorização de erros e possíveis problemas na arquitetura. Todas as semanas calculava-se a dívida técnica com o auxílio de uma ferramenta chamada AI Mentor e eram corrigidos os problemas de arquitetura.

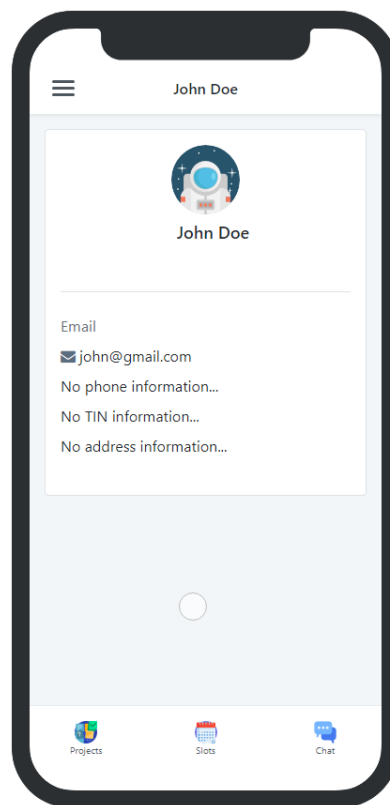


Figura 3.38: Aplicação mobile - ecrã de perfil de utilizador

3.3 Trabalho desenvolvido

O trabalho desenvolvido foi composto pelo fluxo de desenvolvimento de uma aplicação, **Human Network**. Através da metodologia escolhida (**Agile Scrum**), organizou-se o projeto a nível épicos e toda a estrutura dos mesmos, ou seja, divisão do Sprint, quanto tempo duraria um Sprint e dias de demos por Sprint para o cliente e orientador (ver 3.2).

Depois da definição da metodologia e a sua organização, deu-se o levantamento de requisitos, onde foram identificados os requisitos não funcionais e funcionais. Para os requisitos funcionais, desenvolveu-se os modelos de processo para ajudar no desenvolvimento dos mesmos, passagem dos requisitos para user stories e definidas as horas de desenvolvimento de cada uma. No final do levantamento dos requisitos, foi definido o backlog inicial e a distribuição das user stories por épicos.

Tendo os requisitos definidos, foi necessário desenhar a arquitetura que suportaria a aplicação. Foram definidos perfis de utilizadores, os ambiente aplicativos onde a aplicação seria utilizada, planeamento do modelo de dados e a sua otimização, que integrações e suas utilidades na aplicação e, APIs se iriam expor (ver 4.1).

Ao terminar o planeamento da arquitetura, começou-se por criar a arquitetura aplicacional (distribuição da aplicação em módulos), a criação dos módulos, das tabelas e as suas ações **CRUD** e, o começo do desenvolvimentos que garantissem os requisitos não funcionais, ou seja, garantir segurança na aplicação, criando perfis específicos, validações

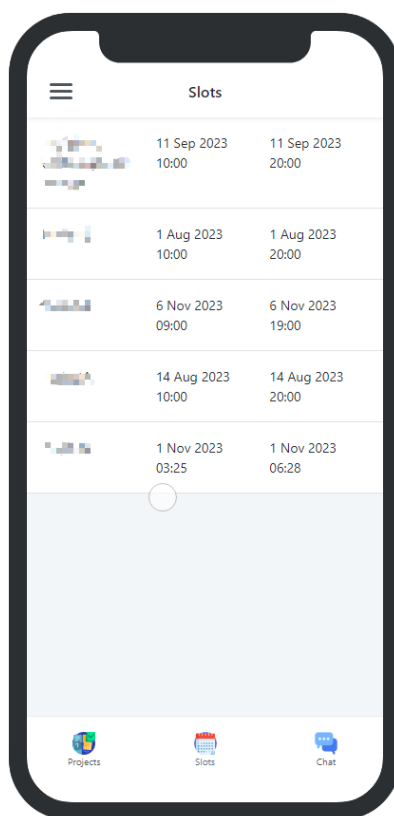


Figura 3.39: Aplicação mobile - ecrã calendário com lista de eventos

para esses perfis e melhoria no sistema de login e sessão de Outsystems (deixando pronto a possibilidade de login através de sistemas externos como Single-SignOn e autenticação de dois fatores para reforçar a segurança), e outros pequenos desenvolvimentos que garantissem uma boa escalabilidade da aplicação.

Tendo o modelo de dados definido e implementado, arquitetura e requisitos bem organizados, começou-se o desenvolvimento dos requisitos funcionais, dentro destes requisitos, estavam especificados as integrações, se necessárias, a utilizar (ver 3.4).

No final de cada desenvolvimento foram testadas as user stories, não só a nível de lógica de negócio (se estavam corretas em relação ao que foi requisitado), mas também a nível técnico (a nível de erros de arquitetura, se foram seguidas boas práticas ou se o desenvolvimento teria provocado algum erro na aplicação).

Depois de dado como terminado os desenvolvimentos, foram feitos testes a nível de desempenho da aplicação (ver capítulo 5). Também foi desenvolvida uma aplicação chamada **Crawler** que extrai dados de websites (ver 4.2.1).

3.4 Integrações

Nesta secção serão apresentadas todas as integrações usadas, uma pequena descrição das mesmas e onde foram utilizadas na aplicação **Human Network**.

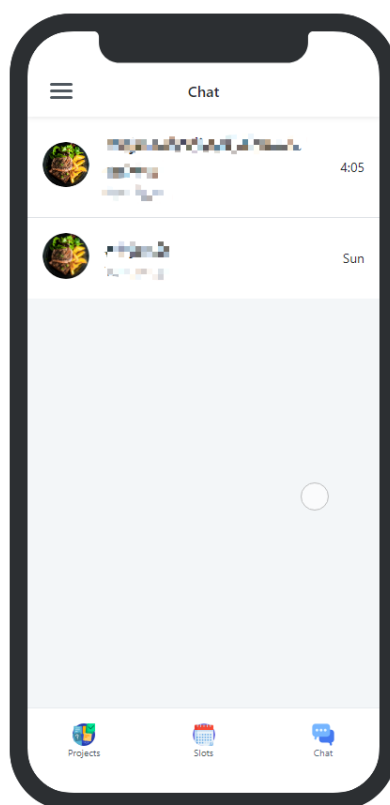


Figura 3.40: Aplicação mobile - sistema de mensagens

3.4.1 OTP - Two Factor Authenticator

Ferramenta de tokens que implementa serviços de verificação de duas etapas usando algoritmos de one-time password baseado em [HMAC](#) [33].

É uma das ferramentas mais utilizadas em aplicações web mais recentes graças à segurança adicional que traz.



Figura 3.41: Microsoft Authenticator

Esta ferramenta é usada para fortalecer a segurança das contas dos utilizadores que assim o queiram. A sua utilização não é obrigatório, mas estará disponível para todos os utilizadores. Basta configurar no perfil (ver figura [3.42](#)) e depois ao fazer login, aparecerá a seguinte figura [3.43](#).

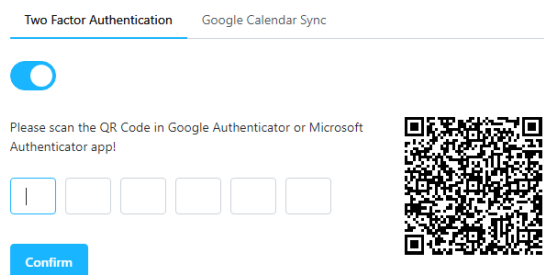


Figura 3.42: Configuração de autenticação de dois fatores na aplicação

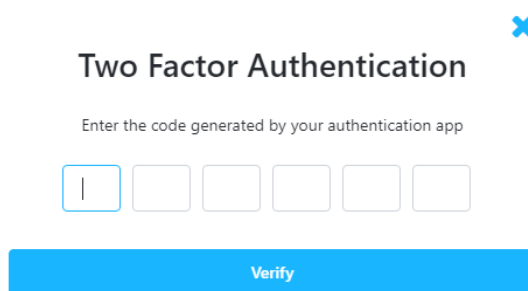


Figura 3.43: Estado da aplicação depois de Login

3.4.2 OAuth 2.0 - Google Authentication

OAuth 2.0 [34] é um protocolo de autorização que permite autenticação entre aplicações. Esta ferramenta é muito usada, principalmente em autenticação rápida da Google.



Figura 3.44: Google Authentication

Esta ferramenta é utilizada de forma a facilitar e a tornar mais seguro o Login.

3.4.3 Google Calendar

Google Calendar é um calendário desenvolvido pela Google.

Na **Human Network**, é possível utilizadores gerirem os seus eventos de projetos, através do Google Calendar. Foi desenvolvida uma sincronização destes eventos com o Google Calendar (criação, edição e remoção de eventos). Depois de configurado no perfil do utilizador, basta o mesmo inscrever-se num evento e aparecerá no seu calendário, como podemos ver na figura 3.47

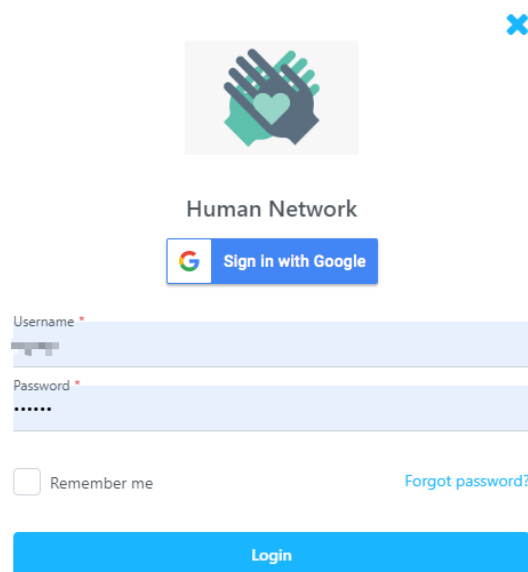


Figura 3.45: Bloco de Login com o botão para autenticação pela Google



Figura 3.46: Google Calendar

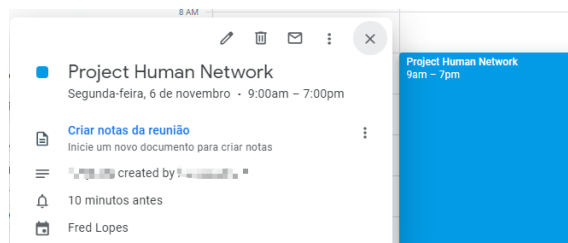


Figura 3.47: Google Calendar - Evento criado

3.4.4 reCAPTCHA

reCAPTCHA [35] é um serviço de proteção contra fraude e abusos desenvolvido na Universidade Carnegie Mellon e agora propriedade da Google.



Figura 3.48: reCAPTCHA

Este serviço foi utilizado para impedir robôs de acederem à aplicação (ver figura 3.49).

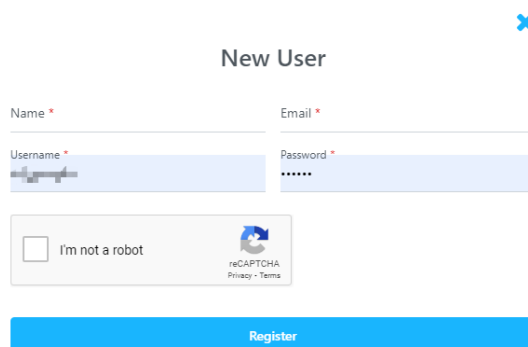
A registration form titled "New User" with a blue close button (X) in the top right corner. The form contains four input fields: "Name" and "Email" at the top, and "Username" and "Password" below them. The "Username" field has a small icon of a person. Below the input fields is a reCAPTCHA widget with a checkbox labeled "I'm not a robot" and a reCAPTCHA logo. At the bottom of the form is a blue "Register" button.

Figura 3.49: Registo de utilizador com reCAPTCHA

3.4.5 Image Optimizer

Sistema que otimiza imagens, reduzindo o tamanho, aumentando o tempo de carregamento de ecrã, mas mantendo a qualidade. [36]



Figura 3.50: Image Optimizer

Esta componente foi utilizada no carregamento de todas as imagens, desde carregamento de imagens de perfil de utilizadores, imagens de projetos, Redes, categorias e até mesmo de patrocinadores.

3.4.6 TouchID

TouchID [37] é um sensor biométrico que permite fazer login numa aplicação através do scan da impressão digital.



Figura 3.51: TouchID

3.5 Sumário

Neste capítulo organizou-se as várias fases do projeto, o levantamento dos requisitos e as suas especificações, a construção dos modelos de dados de modo a apoiar os desenvolvimentos e as respetivas [USs](#) que foram criadas a partir destes modelos e, as integrações

que foram usadas na aplicação.

Todo o estudo feito para este capítulo ajudou no desenvolvimento dos requisitos especificados. Esta organização prévia e o detalhe aprofundado de cada requisito, facilitou o desenvolvimento dos mesmos.

Nas seguintes tabelas estão detalhadas todas as **USs** desenvolvidas no âmbito deste projeto. As **USs** foram divididas em três tabelas, uma tabela com as **USs** definidas para qualquer tipo de utilizador, **USs** para utilizadores que têm como finalidade criar um projeto e **USs** referentes a *Backoffice* e medidas de segurança e integrações (onde o ator da **US** é o programador)

Identificador	Descrição
HN-1 Landing Page	As a User I want to be able to check the top Projects and Networks so that I can have an overall idea of the application before registering
HN-44 Create User	As a User I want to be able to register in the application
HN-2 Login	As a User I want to be able to Login in the application
HN-46 Create Network	As a User I want to be able to create a Network
HN-24 Create Employees	As a Network's admin I want to be able to invite an employees to the app
HN-6 Search for Projects	As a Company's admin or Individual I want to be able to search Projects created by Associations
HN-7 Join Projects	As a User I want to be able to enroll Projects created by Asosications
HN-50 Slot Submission	As a User I want to be able to enroll a Slot
HN-13 Chat Feature	As a User I want to have a chat with other Users, in order to check if the project meets my requirements and, for Associations, if the Users applying for the Project, has a good understanding of the Project at hands
HN-54 Notification system	As a User I want to be notified everytime I receive a new message
HN-12 Multi-language	As a User I want to be able to change the overall language of the application
HN-42 Google Calendar	As a User I want to be able to sync slots with my Google Calendar
HN-21 OAuth	As an User I want to be able to login using Google's Single Sign-On
HN-30 Two Factor Authentication	As a User I want to have an option to enable Two-factor authentication
HN-53 User Preferences	As a User I want to be able to choose my favorite categories so that the projects suggested to me, are accordingly with my preferences
HN-17 Suggested Projects	As a User I want to have a list of suggested projects based on my preferences
HN-16 Onboarding	As a User I want to have an Onboarding so that I can understand how the application works

Tabela 3.1: User Stories para utilizadores gerais criadas a partir dos modelos de processo

Identificador	Descrição
HN-5 Create Projects	As an Association's admin I want to be able to create and manage Projects
HN-8 Create Project's Slots	As an Association's admin I want to be able to create various slots for a Project
HN-62 Project's FAQ	As an Association's admin I want to be able to create a FAQs section

Tabela 3.2: User Stories referentes a utilizadores de associações criadas a partir dos modelos de processo

Identificador	Descrição
HN-41 Create Roles	As a Developer I want to create and manage roles and their permissions
HN-3 Create Categories	As a User with Backoffice permission I want to be able to create and manage categories
HN-47 Create Sponsors	As a User with Backoffice permissions I want to be able to create and manage Sponsors
HN-33 Email Service	As a User I want to receive an email everytime I register a Network, an Employee, get invited to a Network or my password is resetted
HN-69 reCAPTCHA	As a Developer I want to guarantee that only real users register and login into the application
HN-40 API to manage employees	As a Network's admin I want to be able to sync my employees within the application
HN-43 Crawler	As a Developer I want to create an web scaper, that searches in various Association's websites

Tabela 3.3: User Stories referentes ao programador e a utilizadores *Backoffice* criadas a partir dos modelos de processo

ARQUITETURA HUMAN NETWORK

Neste capítulo será apresentada a arquitetura proposta, o modelo de dados desenvolvido e a aplicação Crawler.

4.1 Proposta de arquitetura

Nesta secção é explicado todo o processo por de trás da modelagem da arquitetura, perfis usados na aplicação, aplicações criadas e os seus módulos, todo o processo de desenvolvimento do modelo de dados, a aplicação para extração de dados e as integrações consumidas e APIs expostas.

A aplicação foi desenvolvida em web e em mobile, sendo que as entidades e grande parte da lógica serão as mesmas.

Nesta secção serão apresentados os tipos de utilizadores, ambientes aplicacionais, arquitetura Outsystems, modelo de dados, integrações e serviços expostos pela aplicação.

4.1.1 Tipos de utilizadores de Human Network

Para se entender a arquitetura do sistema é necessário perceber que tipo de utilizadores irão utilizar a aplicação e quais as suas funções.

Human Network define dois sites, um destinado ao Backoffice (utilizadores da KinetIT que monitorizam os utilizadores da aplicação) e outro destinado aos seus consumidores e visitantes.

Backoffice

Os administradores têm como funções gerir os utilizadores, os seus perfis e projetos dos mesmos (quando se trata de consumidores) e gerir configurações, como a gestão de categorias de projetos. Este tipo de utilizadores têm também dois tipos de roles:

- **Manager:** Utilizadores que gerem configurações, empresas, associações, individuais, categorias e projetos.

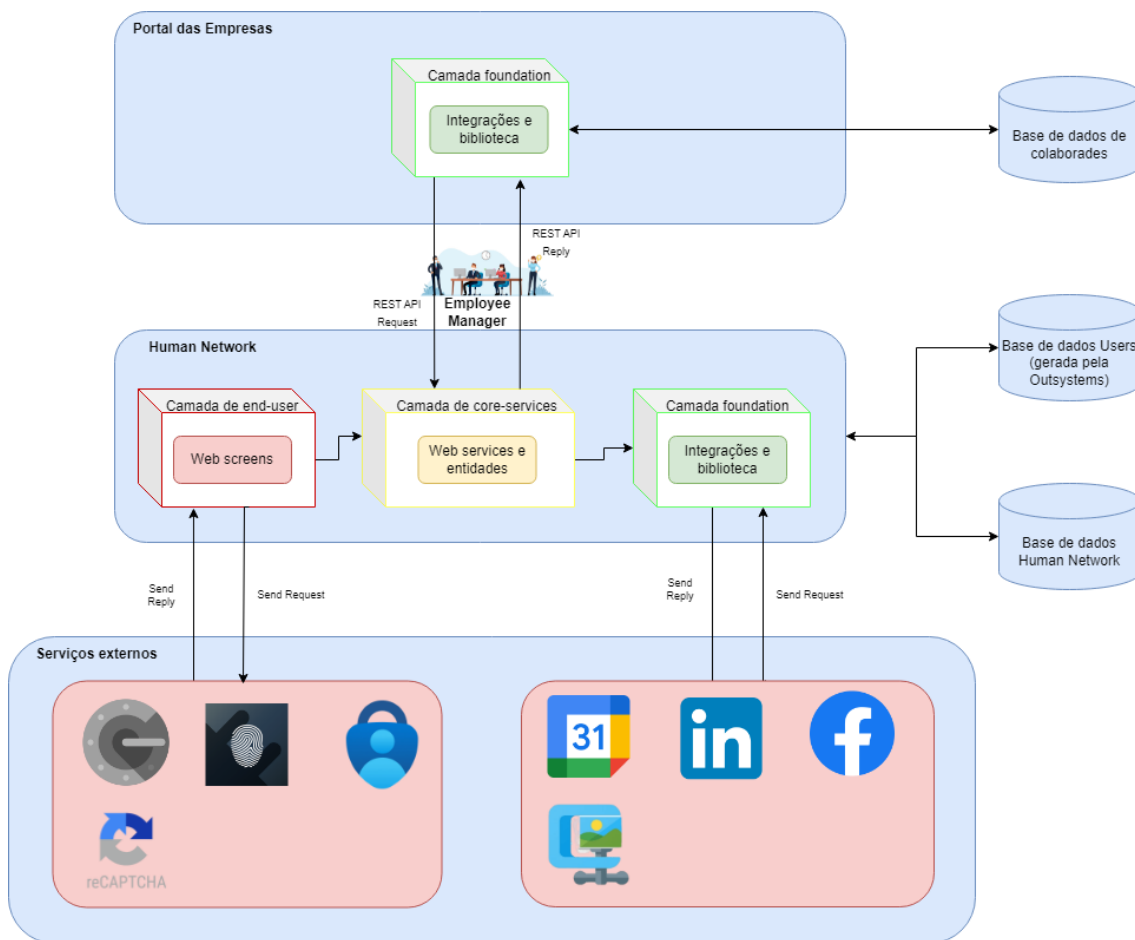


Figura 4.1: Arquitetura Human Network

- **Admin:** Utilizadores que aparte de terem as mesmas permissões que os managers, gerem todo o tipo de utilizadores da plataforma, tal como as suas roles. Estes têm o poder de decidir que trabalhadores da KinetIT têm acesso à plataforma.

Frontoffice

Utilizadores consumidores que estão divididos em três tipos:

- **Associações:** Utilizadores representantes de associações que criam e gerem projetos.
- **Companhias:** Utilizadores representantes de empresas que enviam propostas a associações para contribuírem em projetos.
- **Indivíduos:** Utilizadores que enviam propostas a associações para contribuírem em projetos.

Este tipo de utilizadores têm também dois tipos de roles:

- **Viewer:** Utilizadores que não têm permissões de gestão, só de visualização.

- **Admin:** Utilizadores que têm permissões de gestão. Estes conseguem registar utilizadores da sua empresa na plataforma, tal como decidir que tipo de role estes terão e também gerir as Networks(Associações e Companhias) e os seus projetos.

4.1.2 Ambientes

Serão utilizados três tipos de ambiente:

- **Desenvolvimento:** Ambiente onde o código é desenvolvido.
- **Qualidade:** Ambiente onde o código é testado.
- **Produção:** Ambiente final utilizado pelos consumidores da aplicação.

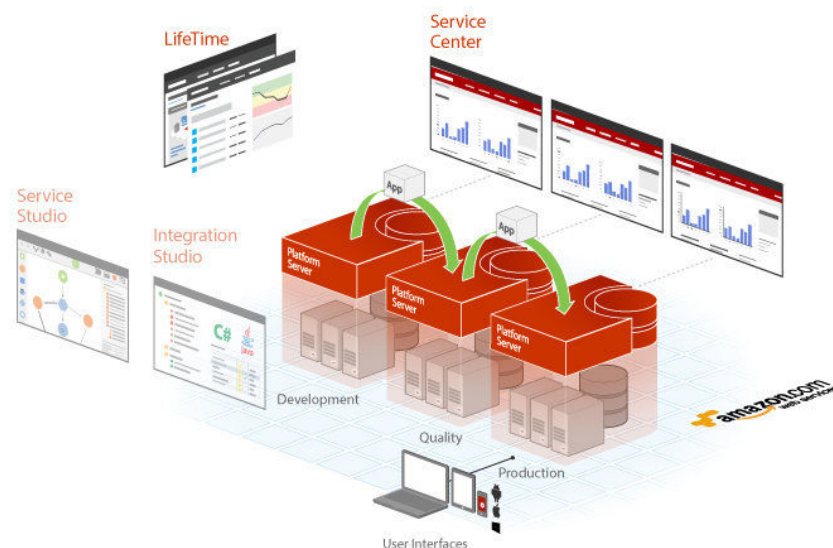


Figura 4.2: Diagrama de ambientes e ferramentas Outsystems

4.1.3 Arquitetura Outsystems

A arquitetura Outsystems está dividida em três camadas:

- **Camada de end-user:** Camada que fornece funcionalidades ao utilizador final, ecrãs, blocos e funções do lado do cliente.
- **Camada de core services:** Camada que contém entidades, exposição serviços, temporizadores, processos e lógica de negócio.
- **Camada de foundation:** Camada que contém integração de serviços externos e bibliotecas.

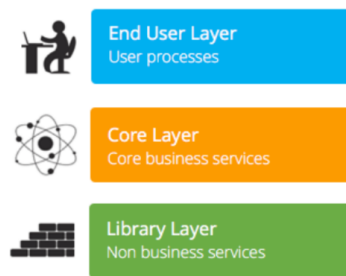


Figura 4.3: Three layer canvas - Divisão de módulos Outsystems

É possível ver a arquitetura das aplicações (web e mobile) acedendo ao Discovery. As seguintes figuras (4.4, 4.6 e 4.5) mostram os módulos criados nas aplicações, tanto como as referências com outros módulos, número de tabelas, ações e blocos UI públicos (utilizadas noutros módulos) e ainda serviços Rest ou SOAP expostos.

A aplicação web está dividida nos seguintes módulos:

- **HumanNetwork_UI**: Módulo com os ecrãs principais da aplicação.
- **HumanNetwork_CW**: Módulo com os componentes/blocos de negócio.
- **HumanNetwork_BO**: Módulo com os ecrãs de configuração/Backoffice.
- **HumanNetwork_Email**: Módulo com os ecrãs e lógica de emails.
- **HumanNetowrk_Pat**: Módulo com os componentes/blocos gerais. Diferença entre este módulo e o _CW é que este módulo contém blocos sem qualquer lógica de negócio como bloco de rating de projetos e Chat.
- **HumanNetwork_Th**: Módulo com o tema e ecrãs gerais (usados no Backoffice e Frontoffice) como Login, Menu e Layouts.
- **HumanNetwork_Lib**: Módulo com a lógica de utilidade, usada por todos os módulos.
- **HumanNetwork_FeatureControl_Lib**: Módulo com todas as **Site Propeties** (variáveis globais constantes) e configurações da aplicação.

A aplicação mobile contém apenas um módulo:

- **HumanNetwork_Mobile**: Módulo com os ecrãs de mobile.

Existe ainda uma aplicação Outsystems, que contém todos os módulos de lógica de negócio, serviços e integrações e APIs:

- **HumanNetwork_API**: Módulo com serviços expostos como Gestão de colaboradores por parte de empresas que utilizem a aplicação (este serviço é explicado em pormenor na secção 4.2).



Figura 4.4: Aplicação Web - Ferramenta Discovery

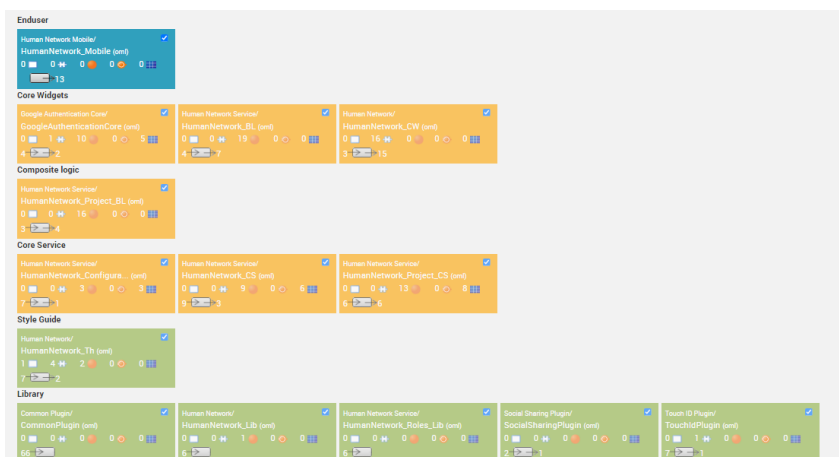


Figura 4.5: Aplicação Mobile - Ferramenta Discovery

- **HumanNetwork_BL:** Módulo com a lógica de negócio de utilizadores e empresas.
- **HumanNetwork_CS:** Módulo com as tabelas e ações **CRUD** de utilizadores e empresas.
- **HumanNetwork_IS:** Módulo com as integrações.
- **HumanNetwork_Project_BL:** Módulo com a lógica de negócio de projetos.
- **HumanNetwork_Project_CS:** Módulo com as tabelas e ações **CRUD** de projetos.
- **HumanNetwork_Configuration_CS:** Módulo com as tabelas e ações **CRUD** de configurações como categorias, estados de projetos, estados de utilizadores, lista de sponsors da aplicação entre outras configurações..
- **HumanNetwork_Roles:** Módulo com a lógica de Roles e Perfis de utilizadores.

4.1.4 Modelo de dados

De forma a chegar a um modelo de dados robusto e otimizado, foi necessário fazer um estudo dos requisitos e um planeamento do modelo, de forma a garantir que não só este

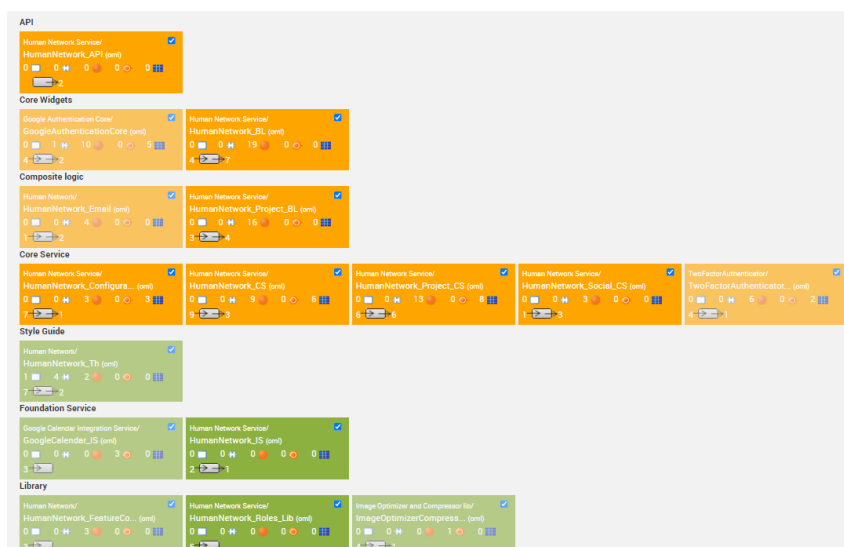


Figura 4.6: Serviços Core da aplicação (Web/Mobile) - Ferramenta Discovery

modelo responde aos requisitos propostos, mas também que está otimizado e pronto para escalar com a aplicação.

O planeamento dos modelos, relações e otimizações foi baseado numa análise prévia dos requisitos através de uma metodologia Agile Scrum.

Entidades e atributos

Do primeiro estudo dos requisitos é possível identificar quais as entidades e atributos que o sistema precisa de guardar e gerir e, deste estudo podemos criar um modelo de dados de entidades. Este modelo representa o modelo de dados no estado inicial, com a simples passagem de requisitos para as entidades e atributos necessários para representar os requisitos.

Deste primeiro estudo as seguintes entidades foram criadas, com os respetivos atributos (ver figura 4.7).

Definição de Relações

A seguir a definir o modelo de dados de entidades, foi necessário definir as relações entre as mesmas. Nesta fase são definidas as relações entre as entidades, o tipo de relação (um-para-um, um-para-muitos ou muitos-para-muitos), identificar as chaves estrangeiras e a integridade referencial.

• Entidades

- **Category** (colunas: **Id** e **Name**).
- **Employee** (colunas: **Id**, **UserId**, **NetworkId**, **EmployeeTypeId** e **StatusId**).
- **EmployeeRange** (colunas: **Id**, **Label** e **Order**).
- **EmployeeType** (colunas: **Id**, **Label**, **Order** e **Is_Active**).

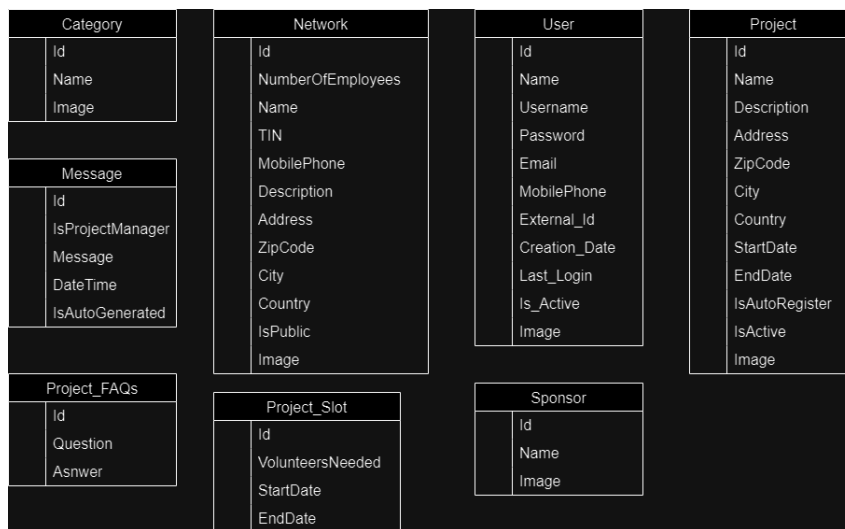


Figura 4.7: Modelo de Entidades e os seus atributos

- **Message** (colunas: **Id**, **ProjectId**, **UserId**, **IsProjectManager**, **Message** e **IsAutoGenerated**).
 - **Network** (colunas: **Id**, **NetworkTypeId**, **StatusId**, **NumberOfEmployeesId**, **Name**, **TIN**, **MobilePhone**, **Description**, **Address**, **ZipCode**, **City**, **Country** e **IsPublic**).
 - **NetworkType** (colunas: **NetworkId**, **Label**, **Order**, **Is_Active** e **IsToShowInRegister**).
 - **Project** (colunas: **Id**, **AssociationId**, **ProjectStatusId**, **ProjectTypeId**, **Name**, **Description**, **Address**, **ZipCode**, **City**, **Country**, **StartDate**, **EndDate**, **IsAutoRegister** e **IsActive**).
 - **Project_FAQs** (colunas: **Id**, **ProjectId**, **Question** e **Answer**).
 - **Project_Slot** (colunas: **Id**, **ProjectId**, **SlotTypeId**, **SlotGenerator**, **VolunteersNeeded**, **StartDate** e **EndDate**).
 - **ProjectStatus** (colunas: **Id**, **Label**, **Order** e **Is_Active**).
 - **ProjectType** (colunas: **Id**, **Label** e **Order**).
 - **SlotType** (colunas: **Id**, **Label** e **Order**).
 - **Sponsor** (colunas: **Id**, **Name** e **Image**).
 - **Status** (colunas: **Id**, **Label**, **Order** e **Is_Active**).
 - **User** (colunas: **Id**, **Name**, **Username**, **Password**, **Email**, **MobilePhone**, **External_Id**, **Creation_Date**, **Last_Login** e **Is_Active**).
- **Relações**
 - A tabela **Employee** tem uma chave estrangeira **UserId** que referencia a tabela **User**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.

- A tabela **Employee** tem uma chave estrangeira **NetworkId** que referencia a tabela **Network**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Employee** tem uma chave estrangeira **EmployeeTypeId** que referencia a tabela **EmployeeType**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Protect (Protect Rule)**.
- A tabela **Employee** tem uma chave estrangeira **StatusId** que referencia a tabela **Status**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Protect (Protect Rule)**.
- A tabela **Message** tem uma chave estrangeira **ProjectId** que referencia a tabela **Project**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Network** tem uma chave estrangeira **NetworkTypeId** que referencia a tabela **NetworkType**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Protect (Protect Rule)**.
- A tabela **Project** tem uma chave estrangeira **AssociationId** que referencia a tabela **Network**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Project** tem uma chave estrangeira **ProjectStatusId** que referencia a tabela **ProjectStatus**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Protect (Protect Rule)**.
- A tabela **Project** tem uma chave estrangeira **ProjectTypeId** que referencia a tabela **ProjectType**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Protect (Protect Rule)**.
- A tabela **Project_FAQs** tem uma chave estrangeira **ProjectId** que referencia a tabela **Project**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Project_Slot** tem uma chave estrangeira **ProjectId** que referencia a tabela **Project**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Project_Slot** tem uma chave estrangeira **SlotTypeId** que referencia a tabela **SlotType**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Protect (Protect Rule)**.
- A tabela **Project_Slot** tem uma chave estrangeira **SlotGenerator** que referencia a tabela **Project_Slot**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)** (esta relação foi feita para situações de repetição de eventos, ao apagar o evento/slot que gerou as repetições de eventos, todos os eventos gerados são também apagados).

Modelo abstrato

Depois de definir o modelo relacional, é necessário desenvolver um modelo de dados conceituais de alto nível que representa as entidades e relações usando, por exemplo, um diagrama de ER (Entidade-Relação). Este modelo é focado na estrutura e nas conexões entre as entidades.

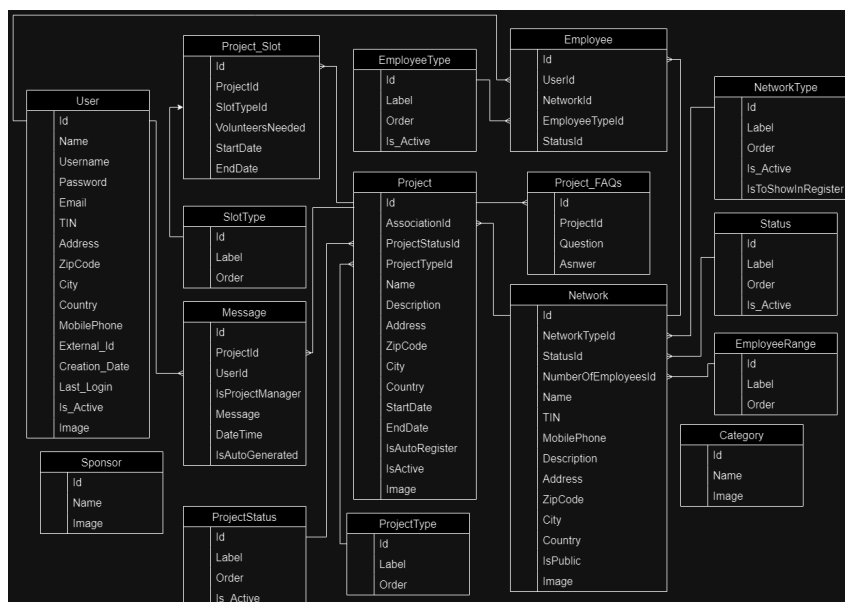


Figura 4.8: Modelo relacional

Antes de se passar à próxima fase, o modelo foi apresentado ao cliente (CEO da KinetIT) e ao orientador para eventuais refinamentos ao modelo. Desse refinamento chegou-se à conclusão que certas melhorias teriam de ser feitas ao modelo apresentado na figura 4.8. Ao modelo anterior foram adicionadas novas tabelas, colunas e desdobrou-se algumas tabelas de forma a separar informação mais pesada e/ou sensível, como imagens ou NIF.

- **Entidades**

- **CategoryEXT** (colunas: **CategoryId**, **Image** e **Filename**).
- **Network_Category** (colunas: **Id**, **NetworkId**, **CategoryId** e **IsFavorite**).
- **NetworkEXT** (colunas: **Id**, **Image** e **Filename**).
- **Project_Category** (colunas: **Id**, **ProjectId**, **CategoryId** e **IsPrincipal**).
- **ProjectEXT** (colunas: **ProjectId**, **Image** e **Filename**).
- **Project_User** (colunas: **Id**, **ProjectId**, **UserId**, **CompanyId**, **StatusId**, **IsFavority**, **Rating** e **Comment**).
- **Slot_User** (colunas: **Id**, **ProjectSlotId**, **UserId**, **StatusId** e **GoogleEventId**).
- **User_Category** (colunas: **Id**, **UserId**, **CategoryId** e **IsFavorite**).

- **UserEXT** (colunas: **UserId**, **TIN**, **Address**, **ZipCode**, **City**, **Country**, **TwoFactorAuth**, **TwoFactorSetup**, **OTPSecret**, **HasCalendarSync**, **CurrentLanguage**, **Image** e **Filename**).

- **Colunas**

- Foram adicionadas às tabelas, **Category**, **Employee**, **Network**, **Project**, **Project_FAQs**, **Slot_User** e **Sponsor** as colunas **CreatedOn** e **CreatedBy**, para se poder monitorizar os utilizadores que criaram e a que horas foram criados os registos.
- Foi adicionado às tabelas, **Category**, **Network**, **Project**, **Project_FAQs** e **Sponsor** as colunas **UpdatedOn** e **UpdatedBy**, para se poder monitorizar os utilizadores que atualizar e a que horas foram atualizados os registos.

- **Relações**

- A tabela **CategoryEXT** tem uma chave estrangeira **CategoryId** que referencia a tabela **Category**, com uma relação de um-para-um entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Network_Category** tem uma chave estrangeira **NetworkId** que referencia a tabela **Network**, com uma relação de muitos-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Network_Category** tem uma chave estrangeira **CategoryId** que referencia a tabela **Category**, com uma relação de muitos-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **NetworkEXT** tem uma chave estrangeira **NetworkId** que referencia a tabela **Network**, com uma relação de um-para-um entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Project_Category** tem uma chave estrangeira **ProjectId** que referencia a tabela **Project**, com uma relação de muitos-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Project_Category** tem uma chave estrangeira **CategoryId** que referencia a tabela **Category**, com uma relação de muitos-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **ProjectEXT** tem uma chave estrangeira **ProjectId** que referencia a tabela **Project**, com uma relação de um-para-um entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Project_User** tem uma chave estrangeira **ProjectId** que referencia a tabela **Project**, com uma relação de muitos-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.

- A tabela **Project_User** tem uma chave estrangeira **UserId** que referencia a tabela **User**, com uma relação de muitos-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Project_User** tem uma chave estrangeira **CompanyId** que referencia a tabela **Network**, com uma relação de muitos-para-muitos entre as tabelas e integridade referencial **Ignore (Ignore Rule)**.
- A tabela **Project_User** tem uma chave estrangeira **StatusId** que referencia a tabela **Status**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Protect (Protect Rule)**.
- A tabela **Slot_User** tem uma chave estrangeira **ProjectSlotId** que referencia a tabela **Project_Slot**, com uma relação de muitos-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Slot_User** tem uma chave estrangeira **UserId** que referencia a tabela **User**, com uma relação de muitos-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **Slot_User** tem uma chave estrangeira **StatusId** que referencia a tabela **Status**, com uma relação de um-para-muitos entre as tabelas e integridade referencial **Protect (Protect Rule)**.
- A tabela **User_Category** tem uma chave estrangeira **UserId** que referencia a tabela **User**, com uma relação de muitos-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **User_Category** tem uma chave estrangeira **CategoryId** que referencia a tabela **Category**, com uma relação de muitos-para-muitos entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- A tabela **UserEXT** tem uma chave estrangeira **UserId** que referencia a tabela **User**, com uma relação de um-para-um entre as tabelas e integridade referencial **Delete (Delete Rule)**.
- Maior parte das tabelas têm uma coluna chamada **CreatedBy** e uma coluna chamado **UpdatedBy**, estas colunas são chaves estrangeiras da tabela **User** e integridade referencial **Ignore (Ignore Rule)**.

Na figura seguinte (4.9) as relações foram pintadas com cor vermelha, verde ou cinzenta, simbolizando a integridade referencial, sendo que vermelho simboliza a *Delete*, verde simboliza a *Protect* e cinzento *Ignore*, explicado em detalhe no paragrafo 4.1.4. Não foram adicionadas ligações entre as tabelas e a tabela **User** referente a gestão das tabelas, para simplificar o modelo, mas estas relações estão especificadas no paragrafo anterior.

Modelo de dados físico

Depois da definição de entidades, as suas relações e colunas, há que definir os tipos de

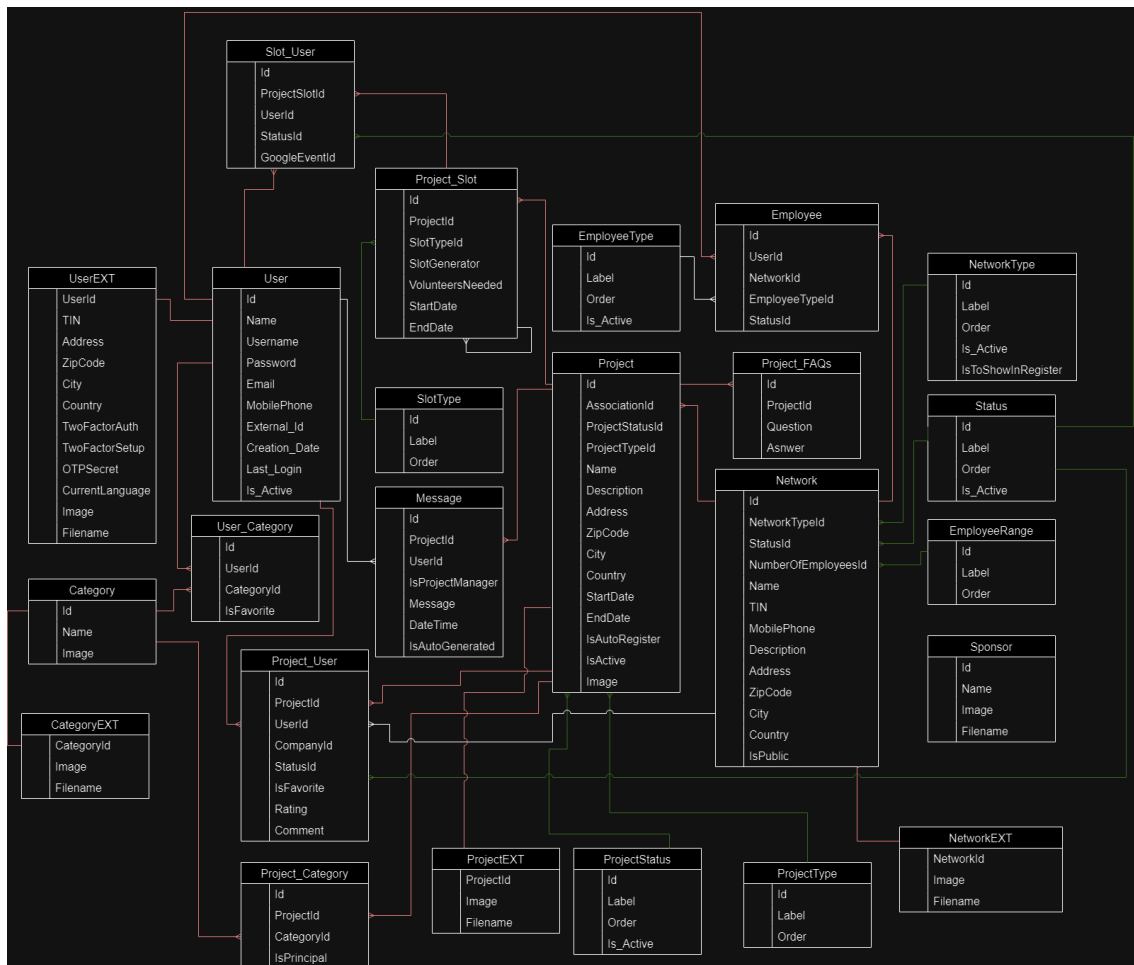


Figura 4.9: Modelo abstrato

dados dos atributos. Este modelo foca-se meramente nos tipos de dados a guardar (ver figura 4.10)

Sistema de Gestão de Base de Dados

Uma parte muito importante na passagem de modelos abstratos para tabelas físicas é a escolha de um sistema de gestão da base de dados. O sistema escolhido foi o SQL Server, simplesmente pela facilidade de integrar com a plataforma Outsystems.

Esquema de base de dados

A última fase de um modelo é a passagem para o esquema de base de dados, onde as tabelas, colunas, chaves primárias e estrangeiras, índices e restrições são criadas.

De forma a não tornar a aplicação monolítica, decidiu-se dividir o modelo de dados em quatro camadas:

- Camada de gestão de utilizadores e as suas roles (já feita pela Outsystems).
- Camada de gestão de empresas e os seus trabalhadores (ver figura 4.12).

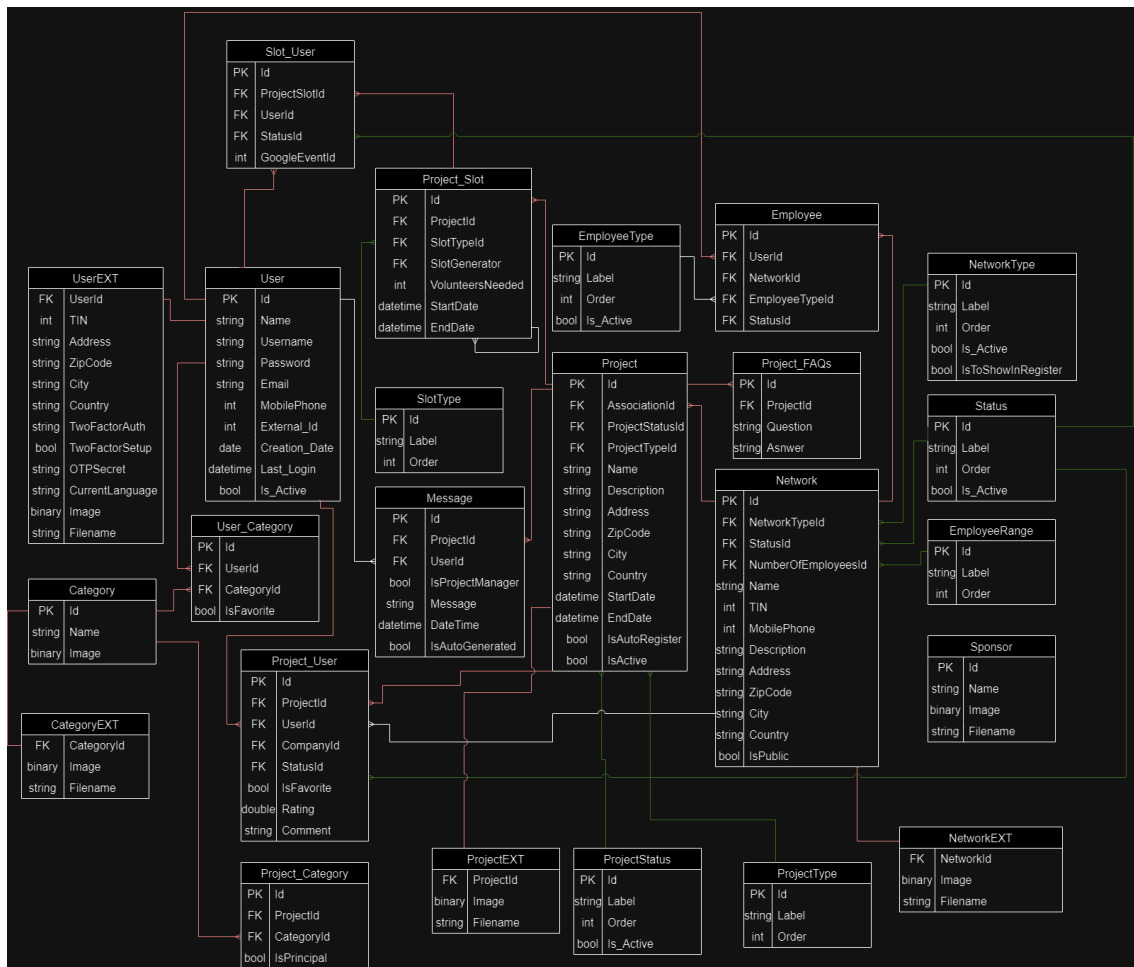


Figura 4.10: Modelo físico

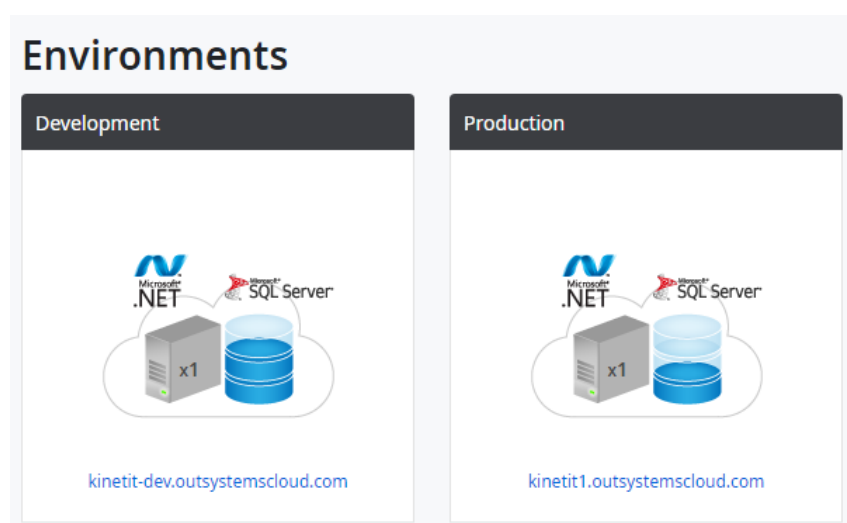


Figura 4.11: SQL Server em ambiente Outsystems

- Camada de gestão de projetos (ver figura 4.13).

- Camada de gestão de configurações (ver figura 4.14).

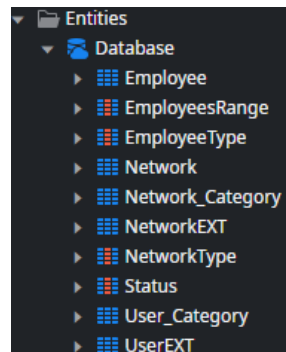


Figura 4.12: Camada de gestão de empresas e trabalhadores

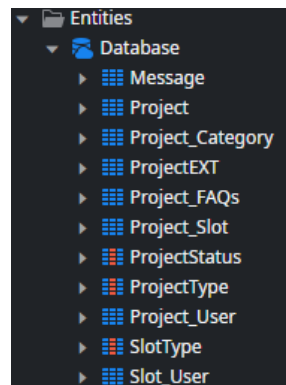


Figura 4.13: Camada de gestão de projetos

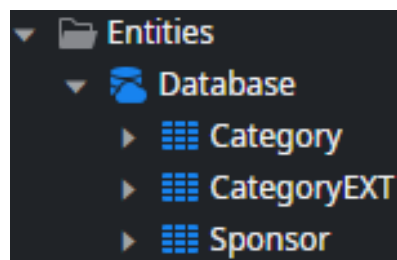


Figura 4.14: Camada de gestão de configurações

No final foram feitas otimizações de forma a tornar as perguntas mais rápidas, através de criação de índices, garantir que as tabelas não têm dados obsoletos apagando registos antigos (através da criação de temporizadores, explicado em detalhe na sub-secção 4.1.4), garantir integridade entre entidades, através de integridade referencial (explicado em detalhe no paragrafo 4.1.4) e segurança e permissões aplicadas à base de dados (explicado em detalhe no paragrafo 4.1.4).

Índices

Uma parte muito importante no desempenho de perguntas, é a escolha de índices. À

medida que uma base de dados vai crescendo, as perguntas tendem a demorar mais. A boa escolha de índices previne esta lentidão, fornecendo uma maneira de localizar rapidamente dados relevantes para as perguntas. Um índice é uma estrutura de dados que guarda um subconjunto de colunas de uma tabela juntamente com apontadores para a linha da tabela.

Índices são uma boa forma de garantir bom desempenho em perguntas, mas para isso as tabelas têm de ser bem analisadas, pois a escolha errada de índices pode piorar o desempenho de perguntas. Por não falar que diminuem a rapidez de operações de escrita, sendo que é necessário fazer-se um estudo metódico. Por isso, as tabelas e os seus atributos foram analisados em detalhes para se perceber que índices (atributos das tabelas) e tipos de índices (coluna única, composto, *unique*, clusterizado ou não-clusterizado) fariam sentido ser criados para cada tabela.

Os índices que foram criados, tal como os testes feitos, estão detalhados na secção 5.1.

Manutenção de Entidades

A manutenção de entidades é uma parte muito importante na escalabilidade e rápida resposta de perguntas. Quanto maior for uma entidade, mais tempo irá demorar a processar uma query. Apesar da otimização através de índices melhorar o desempenho, por vezes não suficiente para esta otimização. Por vezes entidades contêm dados obsoletos, e a gestão destes dados é muito importante, seja através de a criação de regras de exclusão 4.1.4 ou pela limpeza de dados antigos e desnecessários das entidades através de temporizadores.

Foram criados vários temporizadores que correm em paralelo e todas as semanas, que apagam dados antigos, pois nem todas as entidades necessitam histórico.

Integridade Referencial

Ao criar uma relação entre entidades, é necessário definir qual tipo de integridade referencial usar ao excluir registos. A integridade referencial especifica o que acontece com um registo de uma entidade B que referencia um registo da entidade A, quando o registo da Entidade A é apagado.

Existem três tipos de integridade referenciais:

- **Delete** - Garante que quando um registo de uma entidade principal é apagado, todos os registos de outras entidades relacionadas, também são apagados.
- **Protect** - Impede a exclusão de registos da entidade principal enquanto houver registos associados na entidade relacionada.
- **Ignore** - Não garante integridade entre registos, sendo que um registo que seja apagado da entidade principal, não tem nenhum efeito sobre entidades relacionadas.

A Integridade Referencial não só ajuda a manter integridade entre entidades, mas também ajuda a limpar dados obsoletos, por exemplo, se uma associação for apagada,

não faz sentido manter os projetos que esta criou, pelo o que os seus projetos e toda a informação referente aos projetos e à associação, serão apagados.

Segurança e Permissões

Uma das preocupações a nível de segurança numa aplicação, é o acesso a tabelas por parte de utilizadores sem permissões, seja por tentativa de corrupção ou roubo de dados. De forma a garantir esta segurança, a plataforma Outsystems tem um mecanismo que impede a edição de dados fora do escopo da aplicação/módulo, tornando estas tabelas *read-only* (só de leitura, ver figura 4.15), garantindo a segurança contra a corrupção de dados, sendo que só os utilizadores com permissões é que conseguem fazer edição de dados através das ações de [CRUD](#). Apesar destas tabelas ainda poderem ser lidas, o utilizador necessita de permissões e de estar registado na aplicação. Os dados mais sensíveis como informação de um utilizador, Rede ou Projeto, só podem ser acedidos por utilizadores específicos. Só o utilizador e administradores da plataforma têm acesso aos seus dados, dados relativos a Projetos e Redes, só os seus colaboradores e administradores. É de notar que as três tabelas contêm uma tabela de extensão onde os dados mais sensíveis são guardados, não só melhorando a segurança das mesmas, mas também otimizando perguntas feitas a estas tabelas, visto que a informação mais pesada, tal como imagem de perfil, é guardada nestas tabelas de extensão.

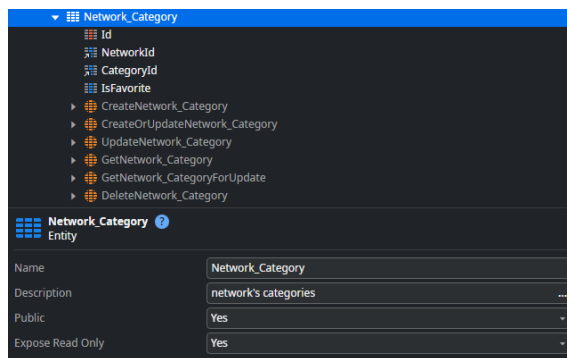


Figura 4.15: Definições de entidades

4.2 Gestão de colaboradores

De modo a facilitar a criação e gestão de colaboradores de uma empresa, foi criada uma [API](#) que pode ser consumida pelas empresas de modo a sincronizar o estado dos seus colaboradores, a criação automática e a remoção de colaboradores que deixem a empresa.

4.2.1 Crawler - Web Scraping

Web Scraping [38] é uma forma de extração de informação de sites tal como texto, imagens, links, tabelas (maior parte da informação está em formato HTML), convertendo a



Figura 4.16: Gestão de colaboradores

informação extraída em estruturas para análise posterior como **JSON** ou **CSV**.

Uma parte bastante importante na publicação de uma aplicação B2B (Business-to-business) é conseguir chegar às empresas de forma direta. Desta forma, uma das estratégias para alcançar as associações e empresas será contactar as mesmas diretamente através de email e telefone. Estes dados serão obtidos através de um **Crawler** que irá extrair contactos email e/ou telefone e guardará esta informação num Backoffice, de modo a que um utilizador com permissões possa contactar estas empresas/associações com o intuito de as convidar a registarem-se na plataforma **Human Network**.

Apesar de, normalmente, web scraping ser uma funcionalidade automática, de forma a haver um foco mais específico em certas associações, optou-se por possibilitar uma gestão mais manual por parte dos trabalhadores da KinetIT, onde os mesmos têm um Backoffice para gerir os websites de onde querem que se extraia informação.

Um dos desafios de web scraping, é a possibilidade de existir mudança na estrutura de um website, mecanismo contra scraping (impedimento de robôs por parte dos websites usando um ficheiro chamado robots.txt) e limite de chamadas, pelo o que este tipo de ferramenta necessita de uma constante manutenção. Existem formas de contornar alguns destes desafios como, por exemplo, usar proxies ou mudar o 'user-agent' (um atributo enviado no cabeçalho do pedido) para contornar estes mecanismos de defesa mas, por questões legais e éticas, estas soluções não foram implementadas.

Existem várias ferramentas e bibliotecas que aceleram o desenvolvimento de Crawlers tais como BeautifulSoup, Soup, Scrapy ou Puppeteer, mas optou-se por utilizar bibliotecas Outsystems para desenvolver esta componente, de forma a manter esta aplicação no mesmo sistema que **Human Network**, Outsystems.

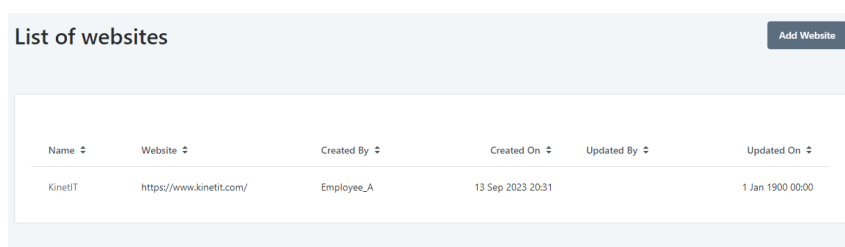


Figura 4.17: Ciclo de vida Web Scraping

Foi criada uma aplicação para gerir o Crawler. Esta aplicação contém:

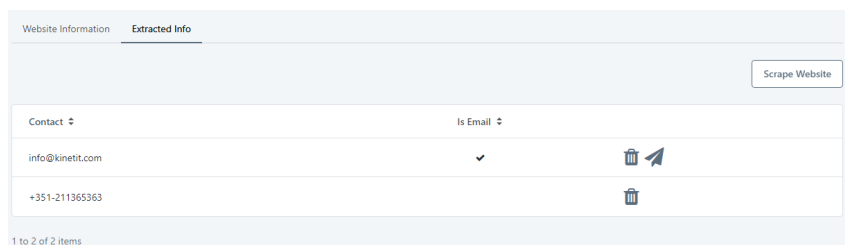
- **HumanNetwork_Crawler_IS**: Módulo de integração, que envia uma chamada REST a um site que retorna o conteúdo HTML.
- **HumanNetwork_Crawler_CS**: Módulo de core services, que analisa o HTML e extrai os contactos e grava-os na base de dados.
- **HumanNetwork_Crawler_UI**: Módulo end-user, que gere os sites que o Crawler deve extrair informação, um ecrã com a informação extraída para cada site e um ecrã que possibilita o contacto via email.

Entrando no módulo end-user, somos deparados com o seguinte ecrã inicial 4.18. Neste ecrã é possível gerir todos os websites, como verificar os dados extraídos, como forçar um envio de email caso algum dos contactos extraídos seja email (ver figura 4.19).



Name	Website	Created By	Created On	Updated By	Updated On
KinetIT	https://www.kinetit.com/	Employee_A	13 Sep 2023 20:31		1 Jan 1900 00:00

Figura 4.18: Aplicação Crawler - ecrã inicial



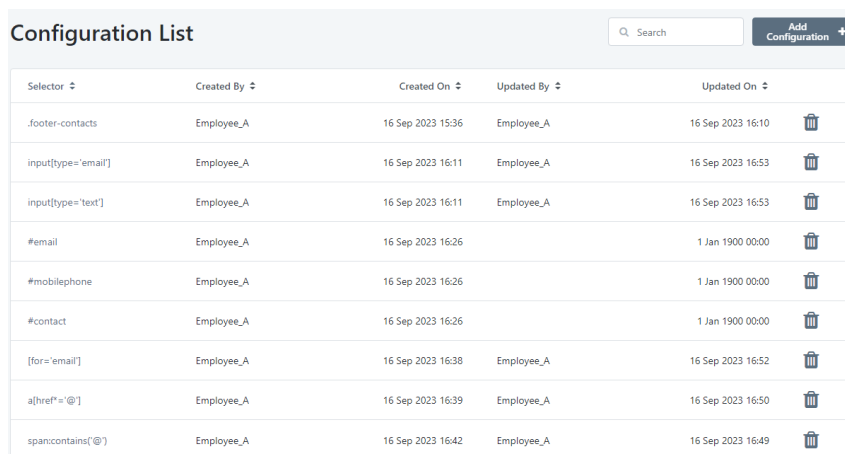
Contact	Is Email
info@kinetit.com	✓
+351-211365363	

Figura 4.19: Aplicação Crawler - ecrã de detalhe de websites

Para configurar as palavras-chave (usando seletores CSS) a serem procuradas num website, existe um ecrã de configuração (ver figura 4.20).

Depois de configuradas as palavras-chave, podemos forçar a extração de informação carregando no botão "Scrape Website" (ver figura 4.19), inicializando o processo de extração.

O processo de extração começa por fazer um pedido ao URL definido, que retorna o conteúdo do website em formato texto HTML. Com o texto recebido, é feito o *parse* para um objeto HTML. Em seguida é feita uma query à base de dados que retorna todas as palavras-chave a pesquisar no ficheiro HTML. A pesquisa é feita através de seletores de CSS, ou seja, por cada palavra-chave, é corrido todo o ficheiro à procura de classes, atributos ou elementos com aquela palavra-chave (ver figura 4.21 com a parte essencial do código de extração de informação). No final, são guardados qualquer tipo de contactos, seja email ou telefone, e caso seja encontrado algum contacto email é enviado um email de promoção da aplicação **Human Network**.



Selector	Created By	Created On	Updated By	Updated On	
.footer-contacts	Employee_A	16 Sep 2023 15:36	Employee_A	16 Sep 2023 16:10	
input[type='email']	Employee_A	16 Sep 2023 16:11	Employee_A	16 Sep 2023 16:53	
input[type='text']	Employee_A	16 Sep 2023 16:11	Employee_A	16 Sep 2023 16:53	
#email	Employee_A	16 Sep 2023 16:26		1 Jan 1900 00:00	
#mobilephone	Employee_A	16 Sep 2023 16:26		1 Jan 1900 00:00	
#contact	Employee_A	16 Sep 2023 16:26		1 Jan 1900 00:00	
[for='email']	Employee_A	16 Sep 2023 16:38	Employee_A	16 Sep 2023 16:52	
a[href*='@']	Employee_A	16 Sep 2023 16:39	Employee_A	16 Sep 2023 16:50	
span:contains('@')	Employee_A	16 Sep 2023 16:42	Employee_A	16 Sep 2023 16:49	

Figura 4.20: Aplicação Crawler - ecrã de configuração de palavras-chave

4.3 Sumário

Nesta capítulo foi definida a arquitetura da aplicação, não só a nível da pilha tecnológica, mas também a nível das aplicações e módulos Outsystems criados, todo o processo de desenvolvimento do modelo de dados, a [API](#) que foi criada para permitir que empresas possam consumir os serviços de gestão de utilizadores disponibilizados pela aplicação e, a explicação da aplicação de extração de dados, **Crawler**.

Todo o estudo feito para este capítulo ajudou na conceção de uma arquitetura e um modelo de dados robusto.

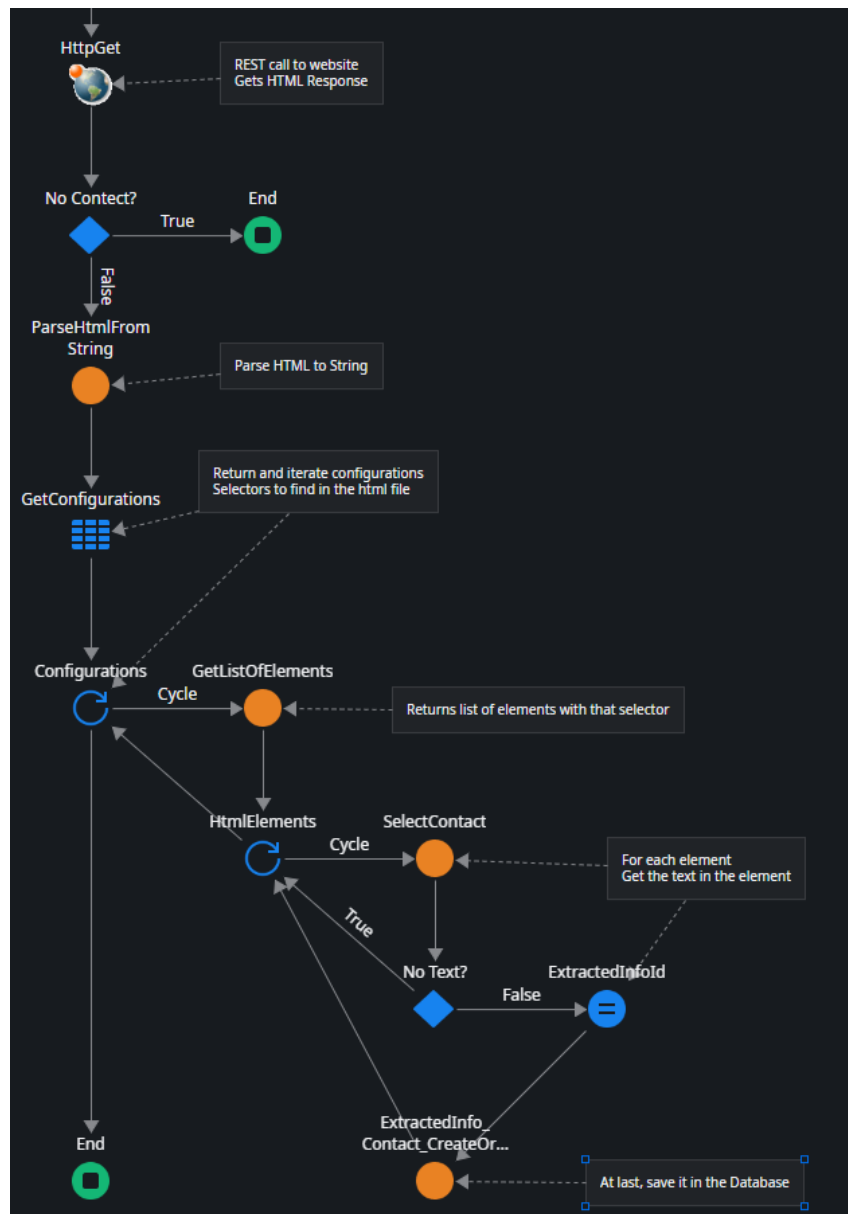


Figura 4.21: Parte essencial do código do Crawler

OTIMIZAÇÃO

Neste capítulo são definidos os índices criados para testes de otimização de perguntas, testou-se os ecrãs mais usados da aplicação a nível de desempenho, testes de segurança e testes de carga.

“O teste de aplicações web é uma prática padrão de teste de software para testar sites e aplicações para identificar possíveis erros antes que estejam acessíveis aos utilizadores. São testadas questões como segurança, desempenho, funcionalidade e capacidade de lidar com tráfego e utilização em tempo real.” [39]

5.1 Otimização de perguntas

Nesta secção são apresentados os índices de forma a otimizar as perguntas.

Como explicado na secção 4.1.4, os índices são um fator muito importante num modelo de dados, porque, se bem utilizados, podem contribuir para diminuir o tempo de resposta das perguntas.

Apesar da contribuição dos índices para diminuir o tempo de resposta das perguntas, em muitos projetos a criação e a gestão de índices é ignorada. Por este motivo, a plataforma Outsystems cria automaticamente alguns tipos de índices ao se criar tabelas, como índice na chave primária para garantir que cada registo na tabela tem um identificador único e que pode ser acedido rapidamente e índices para chaves estrangeiras de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial. A ferramenta AI Mentor Studio da Outsystems também analisa as perguntas criadas na aplicação e recomenda a criação de índices em tabelas.

Para o modelo de dados apresentado na secção 4.1.4, foram criados os seguintes índices (são apresentadas as tabelas, a lista de índices e uma breve explicação da escolha dos mesmos):

- **CategoryEXT:**

1. **CategoryId:** índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter

a integridade referencial.

- **Employee:**

1. EmployeeTypeId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
2. NetworkId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
3. StatusId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
4. UserId e Network: índice composto e único, criado de forma impor a exclusividade dos valores nas duas colunas, ou seja, um utilizador (UserId) não pode estar inscrito como colaborador mais que uma vez numa rede (NetworkId).

- **Message:**

1. ProjectId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
2. UserId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.

- **Network:**

1. NetworkTypeId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
2. NumberOfEmployees: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
3. StatusId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
4. TIN: índice único, criado de forma impor a exclusividade dos valores na coluna, ou seja, impedir que duas redes diferentes tenham o mesmo **TIN** (Taxpayer Identification Number).

- **Network_Category:**

1. **CategoryId**: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 2. **NetworkId**: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 3. **NetworkId e CategoryId**: índice composto único, criado de forma impor a exclusividade dos valores nas duas colunas, ou seja, uma rede (**NetworkId**) só tem um registo para a mesma categoria (**CategoryId**).
- **NetworkEXT**:
 1. **NetworkId**: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 - **Project**:
 1. **ProjectTypeId**: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 2. **AssociationId**: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 - **Project_Category**:
 1. **CategoryId**: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 2. **ProjectId**: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 3. **ProjectId e CategoryId**: índice composto único, criado de forma impor a exclusividade dos valores nas duas colunas, ou seja, um projeto (**ProjectId**) só tem um registo para a mesma categoria (**CategoryId**).
 - **ProjectEXT**:
 1. **ProjectId**: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 - **Project_FAQs**:

1. ProjectId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
- **Project_Slot:**
 1. ProjectId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 2. SlotTypeId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 - **Project_User:**
 1. ProjectId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 2. UserId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 3. ProjectId e UserId: índice composto único, criado de forma impor a exclusividade dos valores nas duas colunas, ou seja, um utilizador (UserId) só se pode inscrever num projeto uma vez (ProjectId).
 - **Slot_User:**
 1. ProjectSlotId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 2. StatusId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 3. UserId: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 4. ProjectSlotId e UserId: índice composto único, criado de forma impor a exclusividade dos valores nas duas colunas, ou seja, um utilizador (UserId) só se pode inscrever num evento uma vez (ProjectSlotId).
 - **User_Category:**

1. **CategoryId**: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 2. **UserId**: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 3. **UserId e CategoryId**: índice composto único, criado de forma impor a exclusividade dos valores nas duas colunas, ou seja, um utilizador (**NetworkId**) só tem um registo para a mesma categoria (**CategoryId**).
- **UserEXT**:
 1. **UserId**: índice simples, criado automaticamente pela a ferramenta Service Studio, de forma a melhorar a performance em operações *JOIN* e para manter a integridade referencial.
 2. **TIN**: índice único, criado de forma impor a exclusividade dos valores na coluna, ou seja, impedir que dois utilizadores diferentes tenham o mesmo **TIN**.

5.2 Otimização de ecrãs

Nesta secção são feitos testes de tempo de resposta dos ecrãs mais usados da aplicação.

O tempo de resposta de ecrãs é um ponto muito importante no desenvolvimento de uma aplicação web. Não só impacta a experiência que um utilizador terá na aplicação, mas também tem impacto a nível funcional. Deste modo, houve um cuidado especial em garantir que todos os ecrãs (principalmente os mais visitados), seguiam as boas práticas de desenvolvimento, tais como:

- **Experiência do utilizado**: A escolha de imagens, icons, cores e o seu contraste, são muito importantes para garantir uma boa experiência. A plataforma Outsystems, através de **IA** (Inteligência Artificial) gera a paleta de cores da aplicação. Esta paleta é baseada na imagem usada como logo. Esta paleta pode ser modificada, mas a gerada pela plataforma garante que as boas práticas de padrões de cores são seguidas.
- **Clareza visual**: Uma parte muito importante de aplicações web é garantir que todo o texto, imagens e icons são visíveis e ao mesmo tempo não são cansativos para o olho humano. Apesar de cores como amarelo e vermelho serem muito usados para mensagens de erro ou alerta, o uso excessivo destas cores, pode tornar-se cansativo para o olho humano. Por isso, a escolha de cores é muito importante
- **Compatibilidade**: Um grande problema com a utilização exagerada de componentes e bibliotecas externas, é a falta de compatibilidade de algumas componentes com todos os navegadores. Por vezes é necessário usar componentes diferentes para se

conseguir a mesma funcionalidade em todos os navegadores. As componentes oferecidas pela Outsystems já fazem esta gestão, pelo o que problemas de compatibilidade, não é um problema na aplicação.

- **Acessibilidade:** Acessibilidade é um dos pontos mais importantes no desenvolvimento de aplicações. Não só promove inclusividade de indivíduos com deficiências visuais, auditivas, motoras e cognitivas possam, independentemente das suas habilidades, possam interagir com o conteúdo da aplicação, garantindo um compromisso ético com os seus utilizadores, trazendo uma maior aderência e audiência, mas também por todos os benefícios a nível de [SEO](#) (Search Engine Optimization). Aplicação com maior acessibilidade são melhor cotadas a nível de motores de busca, potencialmente levando a maior visibilidade e tráfego.

Para garantir todos estes fatores, foram escolhidas as imagens e icons com muito cuidado, escolhendo só icons e imagens que tivessem cores dentro da paleta de cores da aplicação e que fossem fáceis de se perceber o seu propósito. Por exemplo, em vez de se usar um botão a dizer "Apagar" para apagar um registo, usou-se um icon de um caixote do lixo. Mesmo com o cuidado que se teve para escolher estas imagens e icons, foi adicionado o atributo *alt* a todas as imagens e icons, para garantir que pessoas que dependem de leitores de ecrãs, não percam o conteúdo ou contexto das imagens e/ou icons. De forma a garantir também um menor tempo de resposta na obtenção das imagens que estão em base de dados, antes de serem guardadas, foram comprimidas de maneira a diminuir o seu tamanho, mas com a menor perda de qualidade possível.

Foram também criadas *meta tags* que são etiquetas [HTML](#) que fornecem informação adicional sobre uma página aos clientes e motores de busca. Também foram criadas etiquetas que impedem a extração de informação por parte de robôs.

De forma a medir estes parâmetros, usou-se a ferramenta da Google, LightHouse.

5.2.1 LightHouse

LightHouse é uma ferramenta, desenvolvida pela Google, que testa a qualidade de páginas web. [23] Esta ferramenta testa os ecrãs seguindo os seguintes critérios:

- **Tempo de resposta:** Onde é testado o tempo de resposta de perguntas, do primeiro conteúdo a ser carregando no ecrã e do mais pesado (conteúdo visual), se existe código [JS](#) e/ou [CSS](#) que não esteja a ser utilizado, [CSS](#) em linha e se as imagens têm tamanho correto e o melhor formato.
- **Acessibilidade:** Verifica se a paleta de cores está correta, se existe contraste suficiente entre a cor do texto e o fundo da página, se os links e botões contém as variáveis *aria-label* e *title* (variáveis que dão informação extra sobre o que é que os botões ou links fazem, que é um aspeto muito importante para desbloquear utilizador e para pessoas com deficiências visuais) e se as imagens têm uma variável *alt* (que identifica

a imagem, uma variável importante caso haja algum problema no carregamento da imagem e também para pessoas com deficiências visuais).

- **Boas práticas:** Verifica se o ecrã utiliza bibliotecas ou [APIs](#) que estão obsoletas, que as imagens são carregadas com a proporção original, que não existem erros logados na consola, que nenhum serviço falhou e se utiliza [HTTPS](#) (Hyper Text Transfer Protocol Secure).
- **Regras SEO:** Verifica se o ecrã tem título, se o documento tem etiquetas meta e que está aberta a extração de dados por *crawlers* (informação sobre extração de dados na sub-seção [4.2.1](#)), de forma a melhorar a pesquisa em motores de busca.

Estes critérios são avaliados por cores:

- 0 a 49 (vermelho): Pobre
- 50 a 89 (laranja): Precisa de melhorias
- 90 a 100 (verde): Muito bom

Nos próximos parágrafos são apresentados os testes feitos aos ecrãs através do uso da ferramenta LightHouse.

5.2.2 Ecrã Home

O ecrã *Home* é dos ecrãs mais importantes da aplicação, sendo este o ecrã de entrada de novos utilizadores. É um ecrã que, acima de todos os aspetos importantes que um ecrã deve ter, tem de ser rápido a mostrar o primeiro conteúdo carregado (*First Contentful Paint*) e tem de ser acessível para qualquer pessoa (*Acessibility*).

Como se pode verificar na figura [5.1](#), este ecrã tem uma boa cotação, principalmente a nível de acessibilidade e regras [SEO](#). Apesar de a nível de tempo de resposta, o ecrã não estar perfeito, os dois pontos mais importante desta métrica estão com muito bons valores, como podemos ver na figura [5.2](#), onde o primeiro conteúdo carregado (*First Contentful Paint*) está abaixo de um segundo e o tempo de bloqueio (*Total Blocking Time*) é nulo.

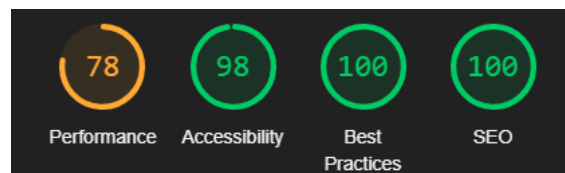
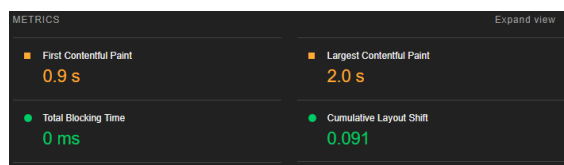


Figura 5.1: LightHouse- Métricas gerais do ecrã *Home*

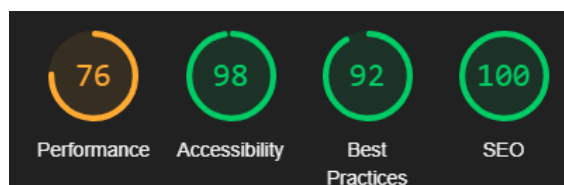
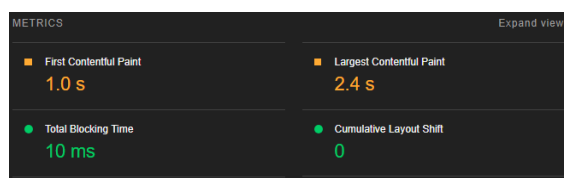
Apesar do tempo de resposta não estar perfeito e precisar de alguns ajustes, o motivo por esta métrica não estar perfeita, tem a ver com a escolha de icons da aplicação, icons mais pesados que os icons convencionais, mas mais apelativos e informativos. Todos os outros fatores estão com valores perfeitos.

Figura 5.2: LightHouse - Métricas de tempo de resposta do ecrã *Home*

5.2.3 Ecrã *Projects*

O ecrã *Projects* é um dos ecrãs mais importante da aplicação. É o ecrã onde são listados os projetos criados por redes e onde utilizadores individuais ou por conta de companhias, conseguem escolher os projetos para se inscreverem.

Como se pode verificar na figura 5.3, este ecrã tem uma boa cotação, principalmente a nível de acessibilidade e regras SEO. Apesar de a nível de tempo de resposta, o ecrã não estar perfeito, os dois pontos mais importante desta métrica estão com muito bons valores, como podemos ver na figura 5.4, onde o primeiro conteúdo carregado (*First Contentful Paint*) está abaixo de um segundo e o tempo de bloqueio (*Total Blocking Time*) está com apenas 10 milissegundos.

Figura 5.3: LightHouse- Métricas gerais do ecrã *Projects*Figura 5.4: LightHouse - Métricas de tempo de resposta do ecrã *Projects*

5.2.4 Ecrã *Network_Details*

O ecrã *Network_Details* é um ecrã muito importante da aplicação. É o ecrã onde as redes podem gerir a sua informação, os seus projetos e os seus colaboradores. Apesar da informação da rede, colaboradores e projetos estarem separados por abas, decidiu-se executar todas as perguntas que obtém estas informações no início do carregamento do ecrã(em paralelo), invés de quando se abre a aba específica, pois desta maneira as perguntas são só executadas uma vez, no carregar do ecrã. Este fator foi muito importante não só a nível de tempo de resposta do ecrã, mas também na navegação entre abas por parte de utilizadores de redes, pois garante que quando um utilizador carrega numa aba,

este não necessita de esperar que o conteúdo fique disponível, pois já foi previamente carregado.

Como se pode verificar na figura 5.5, este ecrã tem uma boa cotação, principalmente a nível de acessibilidade e regras SEO. Apesar de a nível de tempo de resposta, o ecrã não estar perfeito, os dois pontos mais importante desta métrica estão com muito bons valores, como podemos ver na figura 5.6, onde o primeiro conteúdo carregado (*First Contentful Paint*) está abaixo de um segundo e o tempo de bloqueio (*Total Blocking Time*) é nulo. A nível de regras SEO, o único motivo por esta métrica não estar cotada a 100%, é porque foi adicionada uma etiqueta meta que impede a extração de dados do ecrã e reforça a proteção contra DDoS (Distributed denial-of-service), no entanto dificulta os motores de busca.

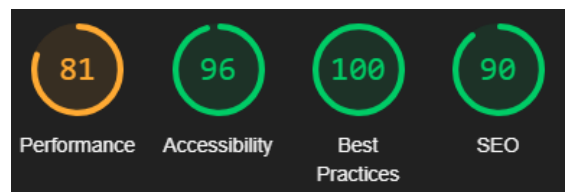


Figura 5.5: LightHouse- Métricas gerais do ecrã *Network_Details*

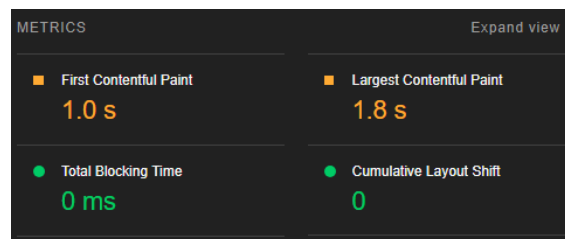


Figura 5.6: LightHouse - Métricas de tempo de resposta do ecrã *Network_Details*

5.2.5 Ecrã *Profile*

O ecrã *Profile* é o ecrã onde os utilizadores podem gerir a sua informação, os eventos onde estão inscritos, as suas configurações e os categorias favoritas. Apesar da informação do utilizador, eventos inscritos, configurações e categorias favoritas estarem separados por abas, decidiu-se executar todas as perguntas que obtém estas informações no início do carregamento do ecrã (em paralelo), invés de quando se abre a aba especifica, pois desta maneira as perguntas são só executadas uma vez, no carregar do ecrã. Este fator foi muito importante não só a nível de tempo de resposta do ecrã, mas também na navegação entre abas por parte de utilizadores de redes, pois garante que quando um utilizador carrega numa aba, este não necessita de esperar que o conteúdo fique disponível, pois já foi previamente carregado.

Como se pode verificar na figura 5.7, este ecrã tem uma boa cotação, principalmente a nível de acessibilidade e regras SEO. Apesar de a nível de tempo de resposta, o ecrã não estar perfeito, os dois pontos mais importante desta métrica estão com muito bons valores,

como podemos ver na figura 5.8, onde o primeiro conteúdo carregado (*First Contentful Paint*) está abaixo de um segundo e o tempo de bloqueio (*Total Blocking Time*) é nulo. A nível de regras [SEO](#), o único motivo por esta métrica não estar cotada a 100%, é porque foi adicionada uma etiqueta meta que impede a extração de dados do ecrã e reforça a proteção contra [DDoS](#), no entanto dificulta os motores de busca.

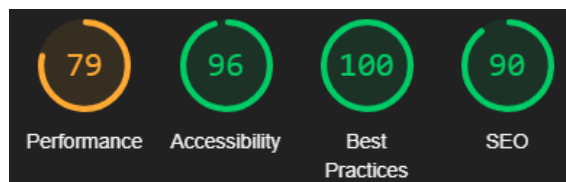


Figura 5.7: LightHouse- Métricas gerais do ecrã *Profile*

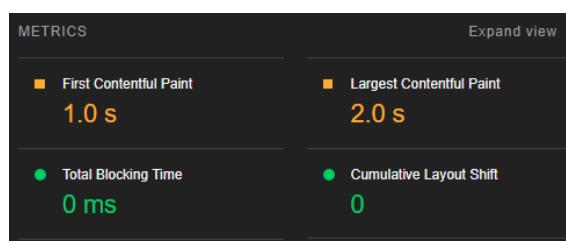


Figura 5.8: LightHouse - Métricas de tempo de resposta do ecrã *Profile*

5.3 Segurança

Nesta secção é explicada a importância de garantir uma boa segurança numa aplicação. Foram feitos testes de segurança à aplicação desde proteção contra injeção de [SQL](#) a proteção contra extração de dados.

A segurança de uma aplicação é um fator bastante importante no desenvolvimento de uma aplicação. Assegurar que os utilizadores têm de estar devidamente autenticados, que só conseguem executar certas funcionalidades com o devido perfil, que existe uma proteção dos seus dados, dentro de outras medidas.

No desenvolvimento da **Human Network**, este fator foi assegurado através de vários mecanismos de segurança diferentes:

- **Autenticação de duas etapas:** Que fortaleceu a autenticação do utilizador.
- **Autorização:** Uma boa gestão de perfis, limitando certas funcionalidades da aplicação para utilizadores sem devida autorização.
- **Encriptação de dados:** Garantir que os dados sensíveis, tais como a palavra-passe são encriptados antes de serem guardados na base de dados e que são usados protocolos de transmissão de dados seguros ([HTTPS](#)).

- **Segurança nas duas frentes:** Garantir não só segurança a nível de Frontend, mas também no Backend, ou seja, validações a nível de ecrã têm de estar seguido de validações a nível do servidor, garantindo uma camada extra de segurança contra ataques de replicação de informação, injeção de dados, [XSS](#) (Cross-Site Scripting) e [CSRF](#) (Cross-Site Request Forgery).
- **Gestão de sessão:** Através da utilização de *cookies* e *tokens* com vida útil curta e que são recalculados depois de login.
- **Monitorização do sistema:** A plataforma Outsystems oferece uma monitorização total da sua fábrica através da ferramenta **Service Center**.
- **Testes contra extração de dados:** Utilizou-se a aplicação **Crawler** para testar esta medida (teste demonstrado na sub-secção [5.3.2](#)).
- **Restaurar e Recuperar:** A plataforma Outsystems garante esta medida ao guardar cópias de segurança da base de dados.

5.3.1 Proteção contra injeção SQL

A prevenção de injeção de [SQL](#) é das partes mais importantes de segurança numa aplicação. Não só protege a aplicação de ser corrompida, mas também proteger os seus utilizadores. [40]

De forma a garantir proteção contra este tipos de ataques, foi necessário garantir que todas as variáveis de entrada de código [SQL](#), eram *sanitized* (processo de limpeza e validação de variáveis, de forma a prevenir ataques de injeção [SQL](#)) antes de serem usadas e que todas as funções contém uma camada de autenticação, de forma a garantir que é um utilizador real e com permissões a fazer o pedido.

Os testes que garantem esta segurança, estão sinalizados nas seguintes figuras [5.9](#) e [5.10](#) onde a primeira figura mostra uma tentativa de criação de um projeto usando o Postman e a segunda imagem mostra o erro devolvido pela plataforma ao tentar injetar código [SQL](#). Nesta segunda figura podemos verificar um problema com a sanitização de código [SQL](#), certas operações não são sanitizadas para variáveis que estão com o atributo “Expandir em linha” ativo. De forma a contornar esta limitação garantiu-se que as tabelas, fora do módulo de onde foram criadas, são expostas apenas para leitura.

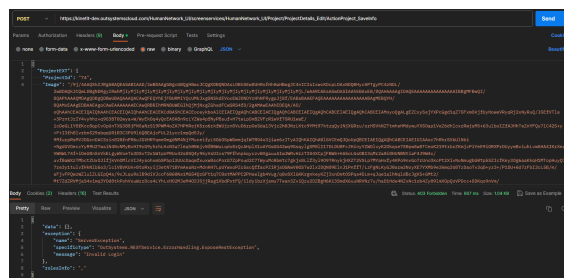


Figura 5.9: Tentativa de criação de projeto através do Postman

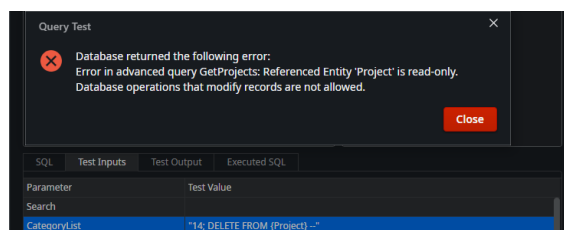


Figura 5.10: Teste de injeção de código SQL

5.3.2 Regras de proteção de dados

Para garantir que aplicações estão totalmente protegidas contra extração de dados, há que garantir que os ecrãs que contêm dados dos utilizadores tem as regras [SEO](#) que impedem esta extração. Um problema com a restrição desta extração é que depois torna mais difícil motores de buscar exporem a aplicação. De forma a garantir a segurança de dados dos utilizadores criou-se um ficheiro "robots-txt" que impede que robôs extraíam informação da aplicação/utilizadores. De modo a garantir que o *scraping* dos motores de busca é feito, criou-se um ficheiro "sitemap" que permite que certos automatismo extraíam informação da aplicação. O conteúdo destes documentos está disponível em anexo (ver [V](#) e [VI](#)).

De forma a testar esta funcionalidade, usou-se a aplicação **Crawler** (ver sub-seccção [4.2.1](#)), que tentará extrair informação da aplicação.

Como podemos ver pelo resultado devolvido pela ação de extração de dados na figura [5.11](#), a aplicação **Human Network**, está protegida contra este tipo de ataques.

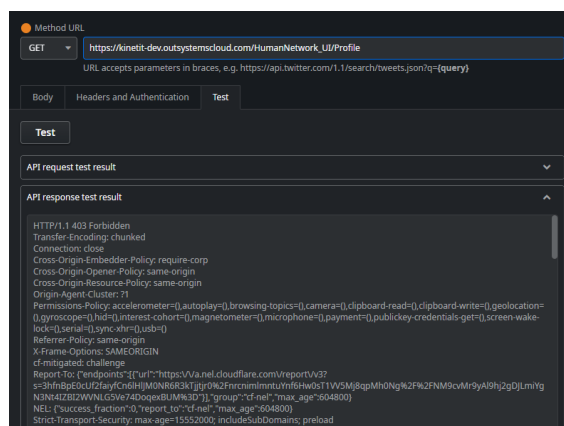


Figura 5.11: Tentativa de extração de dados da aplicação Human Network

5.4 Testes de carga

Nesta secção foram feitos testes à aplicação em relação à disponibilidade, escalabilidade e tempo de resposta da aplicação.

Os testes de carga são testes muito importantes para perceber a disponibilidade e o quão escalável uma aplicação é. Avalia o comportamento da aplicação ao lidar com um número vasto de utilizadores a fazerem pedidos à aplicação, ao mesmo tempo. De

forma a testar este comportamento, usou-se a ferramenta JMeter da Apache para avaliar como a aplicação se comporta quando 100 e 1000 utilizadores, utilizam a aplicação ao mesmo tempo (os ecrãs testados foram a página principal “Home” e a página de projetos “Projects”).

Para ambos os testes foram utilizadas as seguintes configurações para as threads do JMeter:

- **Number of Threads (users):** 100 e 1000
- **Ramp-up period (seconds):** 100
- **Loop Count:** 10

Os fatores a ter em atenção nos grafos disponibilizados nas sub-seções seguintes são:

- **Throughput:** A capacidade (número de pedidos) que o servidor suporta por minuto.
- **Deviation:** O desvio de pedidos, ou seja, a variação do número de pedidos por minuto. Quanto menor o número melhor.
- **Nº of Samples:** O número de amostras enviadas para o servidor.
- **Average:** A média de amostras enviadas para o servidor.

5.4.1 Ecrã Home

Ao correr o teste no ecrã “Home”, obtiveram-se os resultados apresentados nas figuras 5.12 e 5.13.

Da figura 5.12, ao analisar o grafo podemos concluir que foram feitos 525 pedidos por minuto, o desvio foi de 58 pedidos, que é um número bastante bom tendo em conta o volume de pedidos feito, que o número de amostras enviadas foram 1035 e a média de amostras foi de 395 por 100 utilizadores, tudo no espaço de 1 minuto.

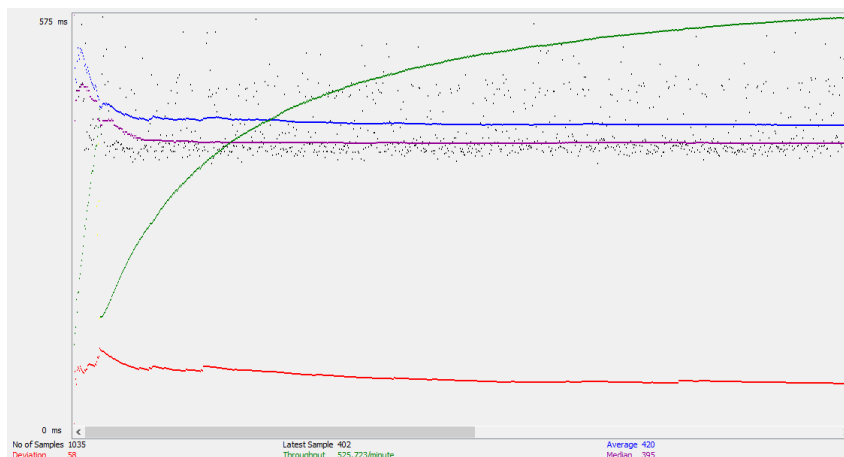


Figura 5.12: Teste JMeter - Ecrã Home com 100 utilizadores

Da figura 5.13, ao analisar o grafo podemos concluir que foram feitos 5749 pedidos por minuto, o desvio foi de 210 pedidos, que piorou em relação ao teste feito a 100 utilizadores, mas que mesmo assim é um valor baixo, que o número de amostras enviadas foram 10000 e a média de amostras foi de 435 por 1000 utilizadores, tudo no espaço de 1 minuto.

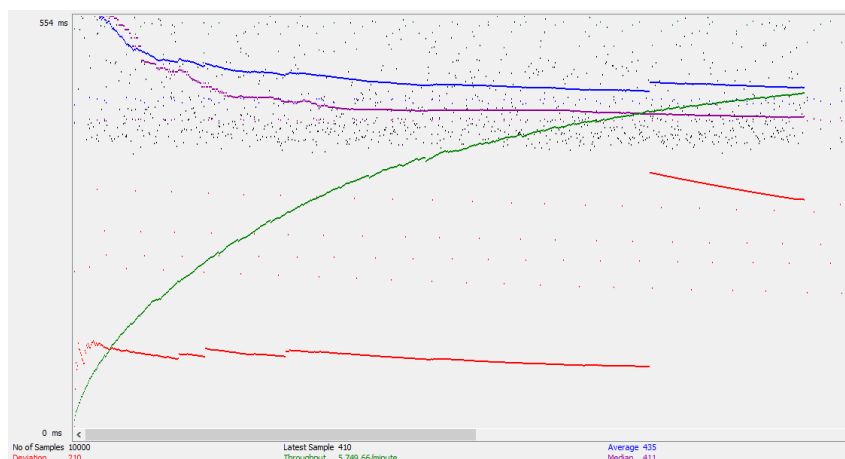


Figura 5.13: Teste JMeter - Ecrã Home com 1000 utilizadores

5.4.2 Ecrã Projects

Ao correr o teste no ecrã “Projects”, obtiveram-se os resultados apresentados nas figuras 5.14 e 5.15.

Da figura 5.14, ao analisar o grafo podemos concluir que foram feitos 582 pedidos por minuto, o desvio foi de 53 pedidos, que é um número muito bom, que o número de amostras enviadas foram 1000 e a média de amostras foi de 430 por 100 utilizadores, tudo no espaço de 1 minuto.

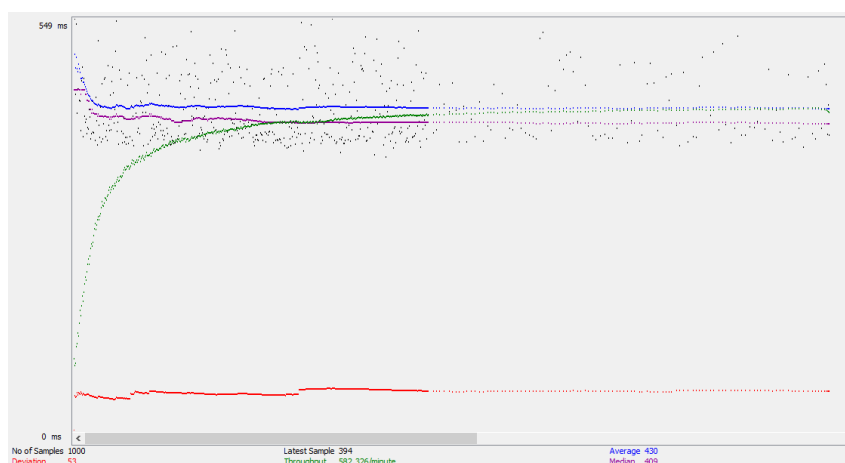


Figura 5.14: Teste JMeter - Ecrã Projects com 100 utilizadores

Da figura 5.15, ao analisar o grafo podemos concluir que foram feitos 5751 pedidos por minuto, o desvio foi de 55 pedidos, que é um número muito bom, que o número de

amostras enviadas foram 10000 e a média de amostras foi de 423 por 1000 utilizadores, tudo no espaço de 1 minuto.

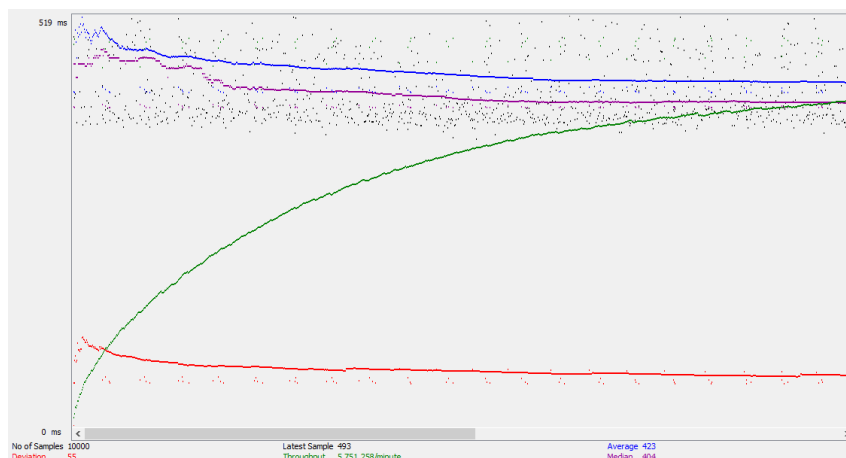


Figura 5.15: Teste JMeter - Ecrã Projects com 1000 utilizadores

5.5 Sumário

Neste capítulo foram definidos os índices criados (os testes a nível de perguntas com e sem índices, ficou para trabalho futuro) e foram feitos testes à aplicação ao tempo de resposta dos ecrãs com o auxílio da ferramenta LightHouse, testes de segurança com o auxílio da ferramenta Postman e a aplicação **Crawler**, testes de segurança contra a injeção de [SQL](#) e testes de carga, que avaliam a disponibilidade, escalabilidade e tempo de resposta da aplicação.

6.1 Cenários de uso

Nesta secção serão apresentados cenários de uso da aplicação **Human Network**.

Existem dois fluxos de utilização da aplicação:

- **Criador de Projetos** - Em que a ideia é criar e gerir uma associação e gerir os seus projetos. Os criadores de projetos são utilizadores que pertencem a uma associação.
- **Inscrito em Projetos** - Em que a ideia é um utilizador inscrever-se em projetos (seja por parte de uma companhia ou por conta própria). Podem-se registar em projetos utilizadores que pertençam a uma companhia ou utilizadores por conta própria.

Para cada fluxo foram criados manuais de utilização para guiar os utilizadores. Estes manuais foram utilizados durante os testes feitos por utilizadores reais. Testes estes detalhados na secção 7.

6.1.1 Criador de Projetos

Um utilizador que se regista na **Human Network** com a finalidade criar e gerir projetos, tem de primeiro criar uma associação (ver figura 6.1). Depois de criada a associação, o utilizador pode gerir os seus colaboradores, informação e projetos.

Ao criar um projeto (ver figura 6.2) o utilizador necessita de preencher a informação referente a esse projeto, tal como o nome, descrição, data de inicio e fim, o tipo de projeto, as categorias do projeto e se o projeto aceitará submissões automaticamente, ou se essa função fica do lado de um utilizador, com as permissões necessárias, da associação que está a gerir o projeto (ver figura 6.3).

Após criar o projeto, é possível gerir todos os eventos do projeto, [FAQs](#) e submissões.

Todo este fluxo é descrito no manual de utilização para criadores de projetos (ver anexo II).



Figura 6.1: Manual de utilização - Criação de Projetos (passo 1)

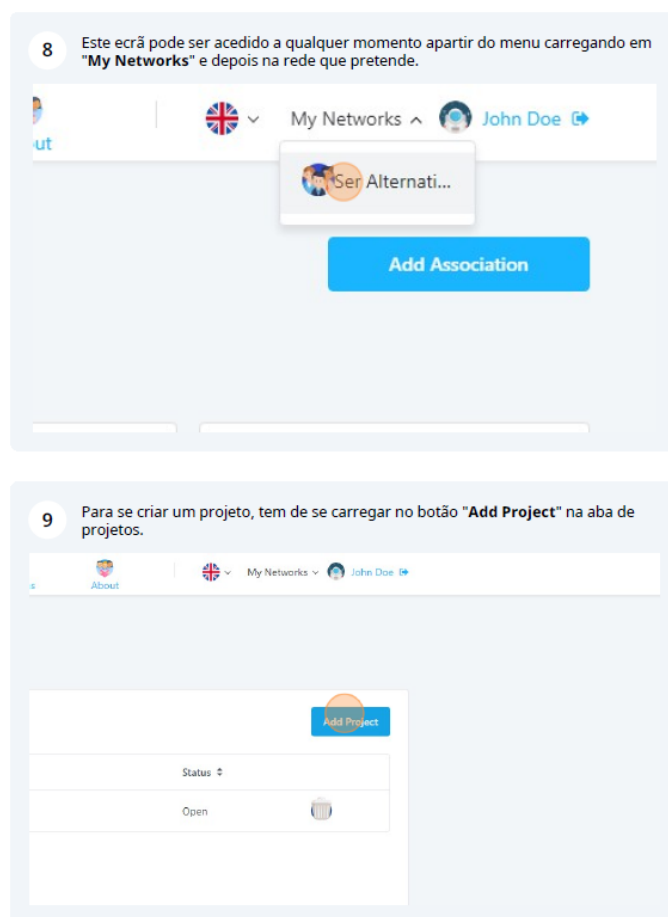
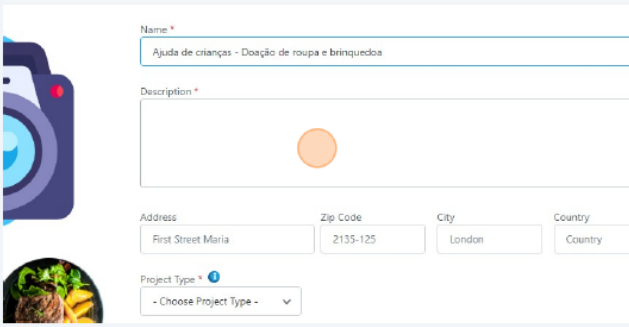


Figura 6.2: Manual de utilização - Criação de Projetos (passos 8 e 9)

10 O utilizador será redirecionado para um ecrã onde pode criar o projeto.



11 Depois de criado o projeto, é possível gerir os seus eventos, submissões enviadas por utilizadores (tentativa de inscrição) e os FAQs.

Para se criar um evento, é necessário escolher a aba "Slots"

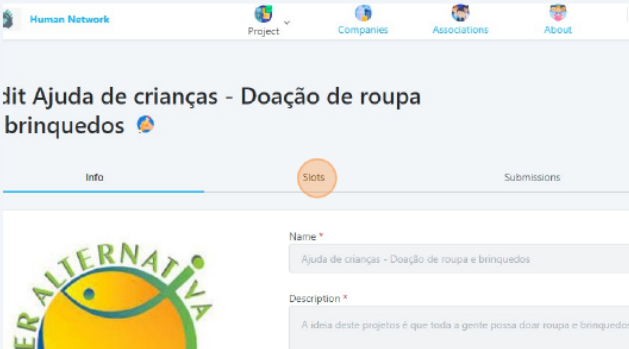


Figura 6.3: Manual de utilização - Criação de Projetos (passos 10 e 11)

6.1.2 Inscrito em Projetos

Um utilizador que se regista na **Human Network** com a finalidade de participar em projetos, tem de primeiro se inscrever num projeto. **Human Network** tem não só um ecrã específico de pesquisa de projetos (ver figura 6.4), mas também na página inicial, a lista dos projetos com mais utilizadores inscritos.

De forma a um utilizador se inscrever num projeto, este necessita de submeter um pedido de participação num evento e ser aceite (ver figura 6.5). Estes eventos estão disponível na sua página de perfil, não só do projeto que se registou, mas de todos os projetos (ver figura 6.6). De forma a facilitar a gestão de eventos, também é possível o utilizador, configurar o seu utilizador de modo a que todos os eventos onde está inscrito, sejam sincronizados para o *Google Calendar*.

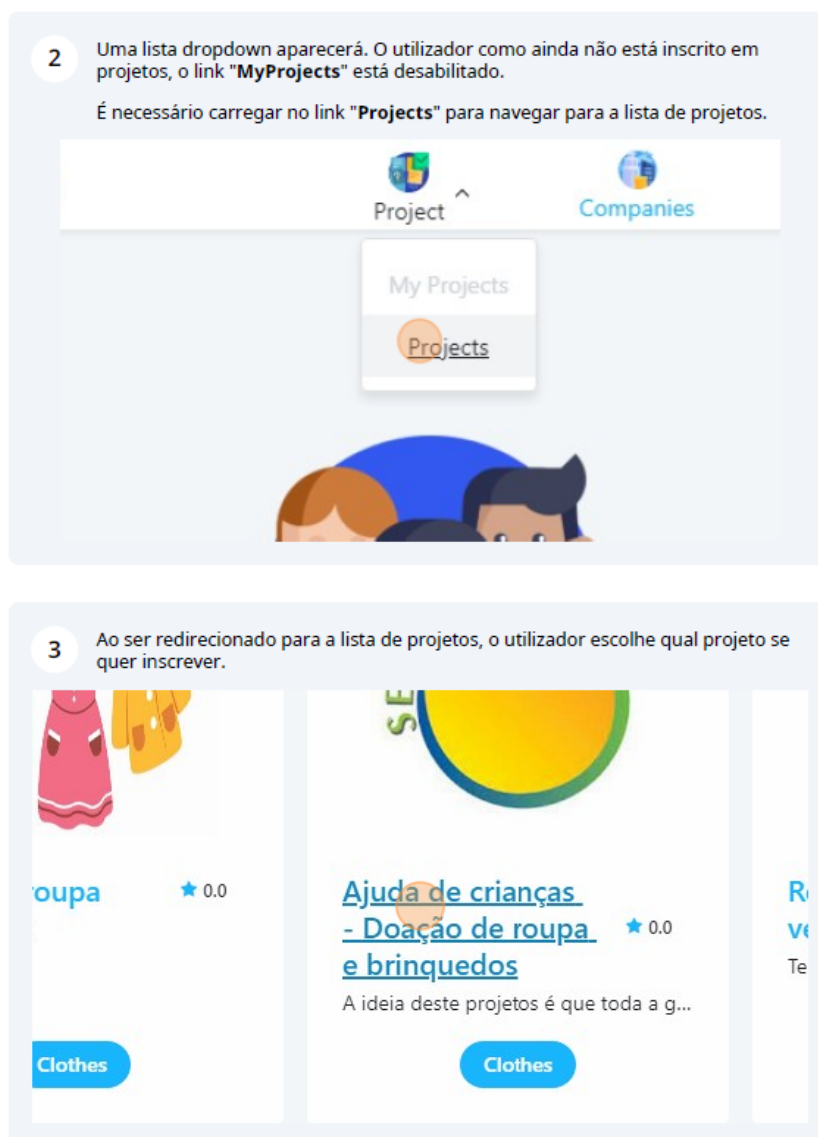


Figura 6.4: Manual de utilização - Inscrição em Projetos (passos 2 e 3)

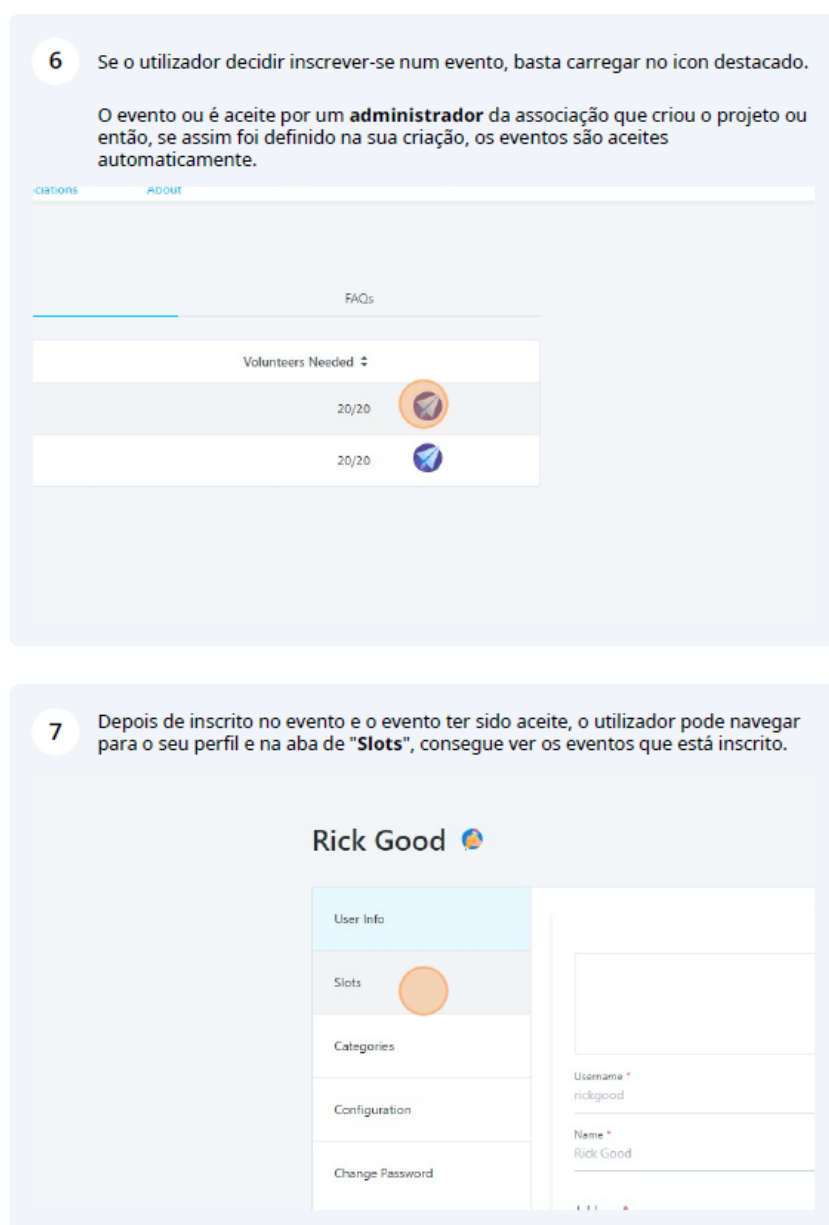


Figura 6.5: Manual de utilização - Inscrição em Projetos (passos 6 e 7)

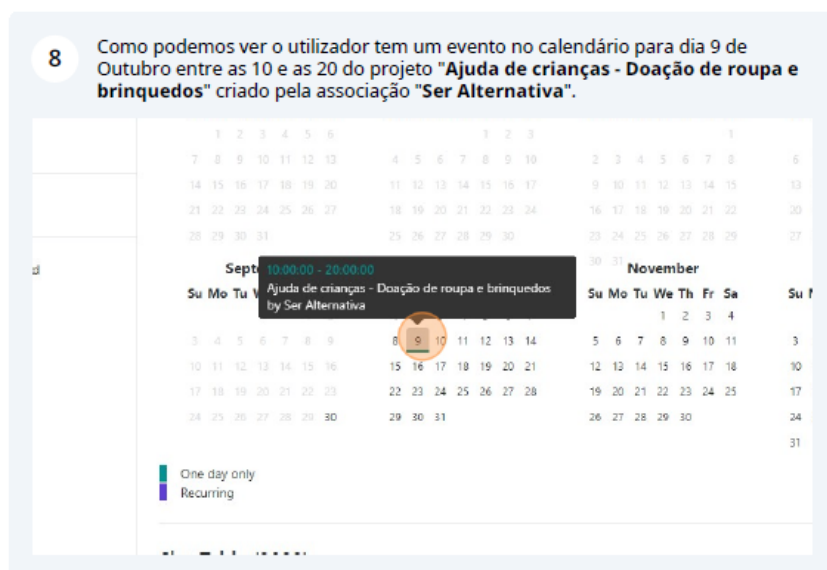


Figura 6.6: Manual de utilização - Inscrição em Projetos (passos 8 e 9)

Todo este fluxo é descrito no manual de utilização para inscrito em projetos (ver anexo [III](#)).

AVALIAÇÃO

Neste capítulo é feita a análise dos questionários feitos a vinte utilizadores da aplicação. Estes utilizadores testaram a aplicação a nível de usabilidade e funcionalidades. O questionário em causa está disponível em anexo [A](#).

Os utilizadores que participaram neste questionário estão entre os 18 e 60 anos de idade, metade é do sexo masculino e a outra metade do sexo feminino e com formação em vários setores, duas pessoas de gestão, um de finanças, uma em enfermagem, dois de design gráfico e o cinco dos participantes em informática.

7.1 Testes de Usabilidade

Nesta secção é estudada a avaliação feita pelos utilizadores a nível de usabilidade da aplicação.

A nível de usabilidade, a opinião dos utilizadores está dividida. Alguns utilizadores acharam fácil de utilizar a aplicação, outros acharam que era difícil de perceber como percorrer certos fluxos (criação de projetos, criação de redes e inscrição em projetos). Alguns destes utilizadores sentiram a constante necessidade de utilizar o manual de utilização que lhes foi fornecido (existem dois manuais, ambos em anexo, um para utilizadores que se inscreveram em projetos [III](#) e outro para quem criou projetos [II](#)).

Os utilizadores no final do questionário, deram a opinião geral sobre a usabilidade da aplicação, e tal como para todos os outros fatores, as suas recomendações foram guardadas como melhorias do projeto.

7.2 Testes Funcionais

Nesta secção é estudada a avaliação feita pelos utilizadores a nível de funcionalidades da aplicação.

A nível funcional, tirando um erro que foi encontrado por dois utilizadores, é da opinião geral dos utilizadores que a aplicação **Human Network** está funcional. Mesmo

tendo sido encontrado um erro, este erro não era bloqueante para o fluxo que estava a ser percorrido pelos utilizadores e foi rapidamente corrigido e retestado com sucesso.

7.3 Resumo do questionário

Nesta secção é feito um resumo da avaliação feita pelos utilizadores, onde se focou mais nas perguntas de resposta livre.

Apesar de ter havido boas opiniões face à utilização da aplicação, principalmente no que tem a ver com funcionalidade e tempo de resposta da aplicação, houve algumas opiniões menos boas e maior parte delas focadas na usabilidade e interface da aplicação.

No final do questionário foi perguntado aos utilizadores o que menos gostaram, dificuldades que sentiram ao usar a aplicação e que requisitos achavam que fazia sentido serem desenvolvidos para a aplicação. Todas estas opiniões, foram analisadas e algumas delas foram adicionadas ao trabalho futuro da aplicação. As opiniões mais relevantes foram:

- Melhorar a gestão de projetos, a sua criação estava um pouco confusa.
- Mais notificações e mensagens de *feedback* a informar que uma ação foi concluída com sucesso
- A secção de categorias não deveria ser drag and drop, ou então deveria ser mais fácil de perceber a funcionalidade. Muitos utilizadores sentiram que era pouco intuitiva.
- Retirar o icon "lápiz" para permitir editar um projeto/rede. Esta funcionalidade de edição deveria ser funcionalidade padrão.
- Tal como na criação, a inscrição de eventos deveria estar em formato calendário.
- Questionário o utilizador, depois do seu registo, o que pretende fazer na aplicação (criar projetos ou inscrever-se em projetos) de modo a encaminhar o utilizador para o fluxo escolhido, facilitando a navegação na aplicação.
- Na página inicial, permitir ver a informação mais detalhada dos projetos e redes em destaque.
- Adicionar integração com *Google Maps* para facilitar o encontro de eventos de projetos.

7.4 Visão geral da avaliação

Nesta secção são apresentados gráficos com as respostas ao questionário e é discutido o peso desta avaliação na aplicação, a razão dos valores obtidos e as recomendações dos utilizadores.

Apesar de maior parte dos utilizadores ter achado fácil de encontrar onde criar redes (ver figura 7.1), ainda uma boa percentagem de utilizadores teve dificuldade em encontrar

e, mesmo depois de consultar o manual de utilização, achou confuso e recomendou adicionar um botão de criação de rede (companhia ou associação) no dropdown no menu “MyNetworks”.

Foi fácil encontrar onde criar redes?

13 respostas

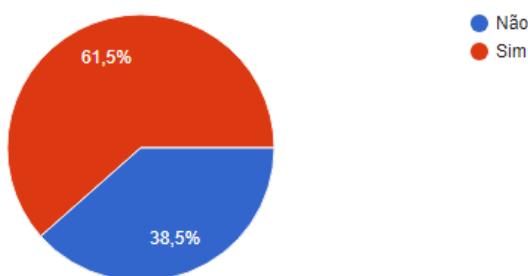


Figura 7.1: Questionário - Foi fácil encontrar onde criar redes?

Muitos utilizadores acharam complicado a gestão de redes (ver figura 7.2). Não que a gestão de informação, colaboradores ou projetos fosse complexa, mas os utilizadores sentiram que a parte de edição estava confusa, pois para editar esta informação, o utilizador precisava de carregar num icon de modo a habilitar este modo. A recomendação foi retirar este icon e permitir que todos os utilizadores com permissões possam, sem qualquer passo intermédio, editar a informação.

Achaste fácil a gestão de redes?

13 respostas

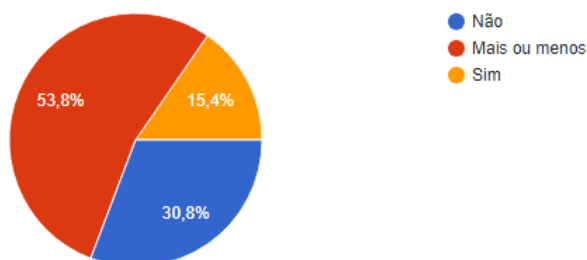


Figura 7.2: Questionário - Foi fácil gerir redes?

Alguns utilizadores acharam complicado a criação de projetos (ver figura 7.3). Os utilizadores sentiram que a parte de criação estava confusa, pois para criar um o projeto, o utilizador precisava de carregar num icon de modo a habilitar o botão “Add Project”. A recomendação foi retirar este icon e permitir que todos os utilizadores com permissões possam, sem qualquer passo intermédio, criar novos projetos.

(Esta pergunta é só para quem criou um Associação)

Foi fácil perceber onde se criavam novos projetos?

9 respostas

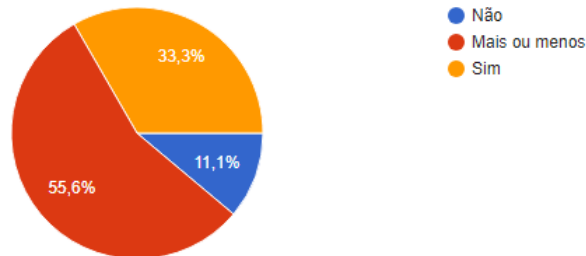


Figura 7.3: Questionário - Foi fácil encontrar onde criar projetos?

Alguns utilizadores acharam complicado a gestão de projetos (ver figura 7.4). Não que a gestão de informação fosse complexa, mas os utilizadores sentiram que a parte de edição estava confusa, pois para editar o projeto, o utilizador precisava de carregar num icon de modo a habilitar este modo. A recomendação foi retirar este icon e permitir que todos os utilizadores com permissões possam, sem qualquer passo intermédio, editar a informação do projeto.

(Esta pergunta é só para quem criou um Associação)

Foi fácil encontrar onde gerir projetos (sem a utilização do manual de utilização)?

9 respostas

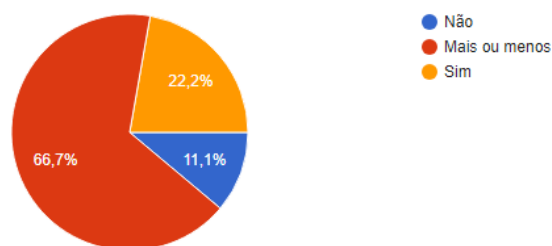


Figura 7.4: Questionário - Foi fácil encontrar onde gerir os projetos?

Muitos utilizadores acharam complicado a gestão de projetos (ver figura 7.5). Não que a gestão de informação de projetos fosse complexa, mas os utilizadores sentiram que a parte de edição estava confusa, pois para editar esta informação, o utilizador precisava de carregar num icon de modo a habilitar este modo. A recomendação foi retirar este icon e permitir que todos os utilizadores com permissões possam, sem qualquer passo intermédio, editar a informação.

(Esta pergunta é só para quem criou um Associação)

Achaste fácil gerir os projetos?

9 respostas

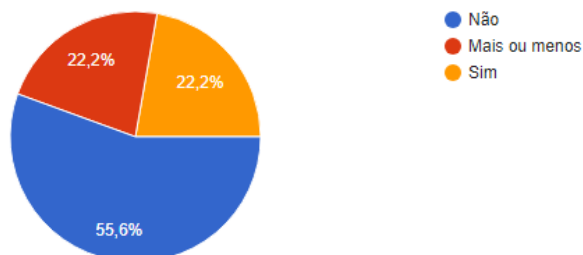


Figura 7.5: Questionário - Foi fácil gerir projetos?

Muitos utilizadores acharam complicado a criação de eventos (ver figura 7.6). Os utilizadores sentiram que a parte de criação estava confusa, pois para criar eventos, o utilizador precisava de carregar num icon de modo a habilitar este modo, o que torna o processo confuso para um utilizador que se depara com um calendário que não faz nada quando clicado. A recomendação foi retirar este icon e permitir que todos os utilizadores com permissões possam, sem qualquer passo intermédio, criar eventos.

(Esta pergunta é só para quem criou um Associação)

Foi fácil criar eventos?

9 respostas

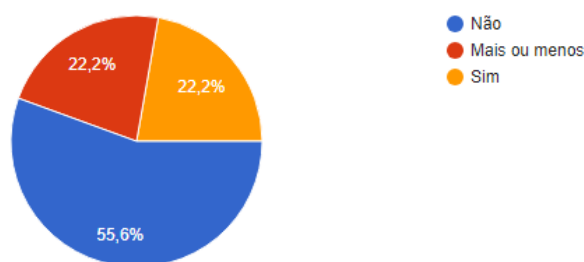


Figura 7.6: Questionário - Foi fácil criar eventos?

Maior parte dos utilizadores não percebeu o conceito de inscrição em projetos (ver figura 7.7). De forma a inscrever-se num projeto, um utilizador tem que ser aceite num evento. Foi recomendação dos utilizadores, que existem duas fases neste fluxo. Primeiro a inscrição no projeto, que faz com que este apareça no ecrã "MyProjects" e só depois a inscrição em eventos deste projeto.

Achaste fácil inscrever em projetos?

12 respostas

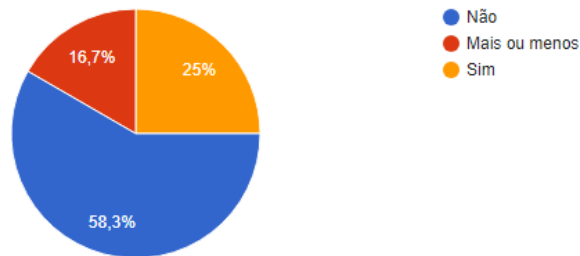


Figura 7.7: Questionário - Foi fácil inscrever em projetos?

Maior parte dos utilizadores não percebeu o conceito de inscrição em eventos (ver figura 7.8). Os utilizadores não conseguiram perceber o que era um evento. Foi recomendação dos utilizadores, adicionar na aba de “Slots”, uma explicação breve sobre o que são eventos.

Achaste fácil inscrever em eventos?

12 respostas

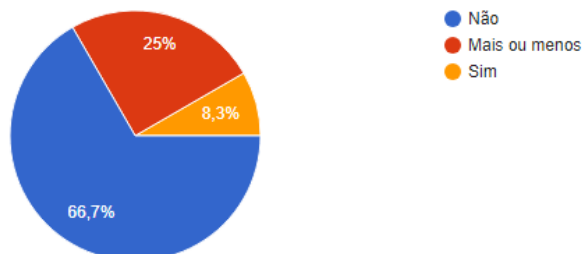


Figura 7.8: Questionário - Foi fácil inscrever em eventos?

Grande parte dos utilizadores concordou que os projetos sugeridos, que apareciam no final do detalhe de um projeto, faziam sentido (ver figura 7.9). Alguns utilizadores acharam que neste ecrã os projetos sugeridos serem baseados na categoria do projeto que se está a visualizar, não faziam sentido. Depois de explicar a estes utilizadores que existem dois tipo de algoritmos para projetos sugeridos, um baseado na categoria do projeto a visualizar e outro nas preferências dos utilizadores, estes concordaram que a abordagem foi bem feita.

Achaste que os projetos sugeridos (no final do ecrã de detalhe de um projeto) fazia sentido face às categorias que escolheste?

13 respostas

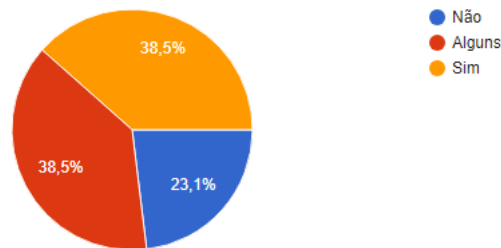


Figura 7.9: Questionário - Projetos sugeridos faziam sentido?

Visto que alguns temas como por exemplo a gestão e criação de projetos, teve uma avaliação mais negativa por causa da escolha de um icon para mudar o modo de visualização para edição, os resultados apresentados na figura 7.10 foram bastante bons, tendo em conta este fator.

Foi necessário a constante utilização do manual de utilização?

14 respostas

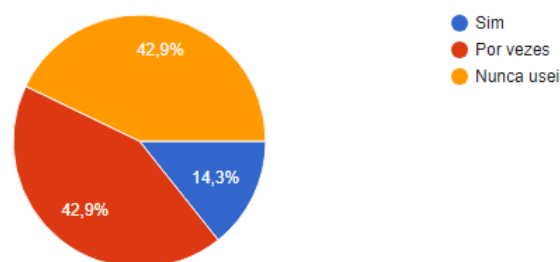


Figura 7.10: Questionário - Foi necessário o constante uso do Manual de utilização?

Apesar dos testes terem sido feitos num ambiente de desenvolvimento, onde os servidores são mais lentos, os resultados apresentados na figura 7.11 foram bastante bons.

De 1 a 5 (sendo 1 muito lenta e 5 muito rápida) a nível de desempenho em que nível achas que aplicação está?

14 respostas

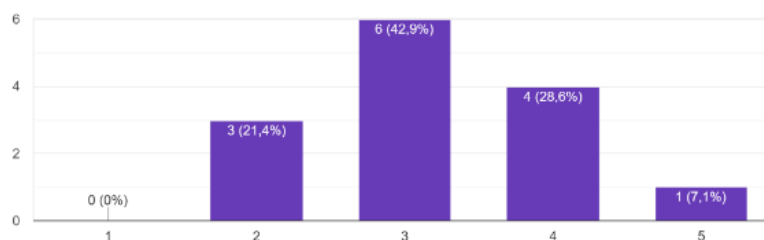


Figura 7.11: Questionário - Avaliação do tempo de resposta da aplicação

Como já foi discutido, um grande problema na aplicação é a usabilidade (ver figura

7.12). Todas as recomendações foram adicionadas ao *Backlog* do projeto para no futuro serem endereçados.

De 1 a 5 (sendo 1 muito difícil e 5 muito fácil) com que facilidade conseguiste utilizar a aplicação (sem recorrer ao manual de utilização).

14 respostas

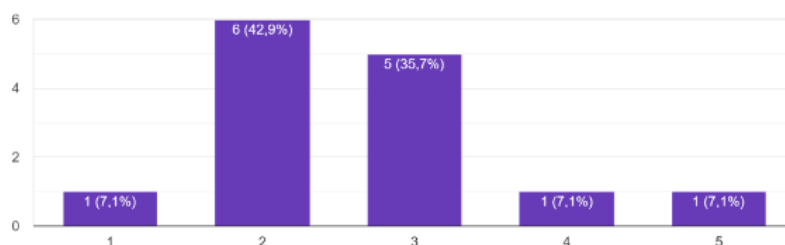


Figura 7.12: Questionário - Avaliação da facilidade de utilização da aplicação

7.5 Sumário

Neste capítulo foi feita a avaliação da aplicação a nível de funcionalidades e usabilidade. Foram disponibilizados dois manuais de utilização para dois diferentes fluxos de utilização, para utilizadores que querem criar projetos e para utilizadores que querem se inscrever em projetos. Estes manuais foram usados na avaliação da aplicação por parte de utilizadores reais. Esta avaliação por parte de utilizadores reais, ajudou a perceber as limitações da aplicação e o trabalho futuro (melhorias e novas funcionalidades que podem vir a ser desenvolvidas).

CONCLUSÕES

8.1 Contribuições

O objetivo principal desta tese, era a conceção de uma aplicação que promovesse causas sociais e pode-se declarar que este objetivo foi cumprido. Outro objetivo muito importante, era garantir um sistema com a maior qualidade possível a nível de robustez, escalabilidade, tempo de resposta, segurança, usabilidade e que seguisse as boas práticas de programação. Todos estes objetivos foram cumpridos, mesmo havendo margem de melhoria nalguns temas, foi feito o trabalho e estudo de forma a garantir uma aplicação com qualidade. Mas o objetivo principal da conceção desta plataforma, era ajudar pessoas necessitadas e oferecer, às associações e utilizadores e companhias que querem ajudar pessoas em necessidade, uma plataforma de fácil acesso, gestão de projetos de cariz social e de fácil procura e participação nestes projetos. **Human Network** vem desta forma facilitar este processo e oferecer funcionalidades que outras plataformas do mesmo tipo não oferecem.

Foi feito o trabalho total de organização e desenvolvimento de um projeto. Desde da escolha da metodologia do projeto, o levantamento dos requisitos, a definição da arquitetura e do modelo de dados, tal como o estudo aprofundado destes dois temas, a escolha de ferramentas de auxílio ao desenvolvimento do projeto tais como a tecnologia de desenvolvimento (Outsystems), escolha de integrações que garantissem a finalidade de certos requisitos, escolha de ferramentas de monitorização de erros, testar a aplicação através de ferramentas como LightHouse, Selenium, JMeter e avaliação da aplicação e, posterior questionário, da aplicação desenvolvida a utilizadores reais.

Aos objetivos definidos no começo do desenvolvimento da tese, foram adicionados novos objetivos, como por exemplo, a construção de uma aplicação que extraísse dados de um website, em que o objetivo desta aplicação é extrair contactos de associações, de forma a convidá-las a registarem-se na aplicação **Human Network**.

Foi criada também uma documentação do projeto, onde é explicada a arquitetura do projeto e a lógica de negócio.

A ideia foi garantir um bom *hand-over* do que foi feito. Focado não só no código desenvolvido, mas também garantir um código limpo, que segue boas práticas, fácil de

perceber e documentado, uma boa organização do projeto, desde documentação lógica e detalhe dos requisitos, à garantia de que existe uma passagem fácil de informação para novos membros do projeto.

8.2 Limitações

Apesar dos objetivos definidos no início da tese terem sido alcançados, há sempre espaço para melhorar. Um dos pontos mais importantes a melhorar é a usabilidade da aplicação. Foi apontado por muitos utilizadores que a aplicação necessita de estar mais simples e tem de ser mais fácil de perceber como executar certos fluxos, como por exemplo a criação de projetos. Muitos utilizadores não sentiram necessidade de utilizar o chat, visto que os projetos têm uma secção de FAQs. Um ponto muito importante de melhoria é a garantir que a aplicação está otimizada para uso em computador, telemóvel e tablet. Esta é uma limitação atual da aplicação, um ponto que não foi muito focado por existir uma versão **Mobile** da aplicação, mas mesmo assim, esta otimização é uma necessidade futura do projeto.

8.3 Trabalho futuro

No começo do levantamento de requisitos, foi analisado o tempo que se demoraria a desenvolver a aplicação. Visto que os requisitos levantados excediam o tempo de projeto, ordenou-se por relevância os requisitos, onde alguns dos requisitos foram marcados como não prioritários e que inicialmente não entraram no âmbito do projeto, apesar de que alguns destes requisitos acabaram por ser desenvolvidos. Foram criadas **USs** para estes requisitos. Estas **USs** estão definidas na tabela 8.1.

Apesar de não entrarem no âmbito do projeto tese, estes requisitos tiveram de ser bem detalhados para permitir que quem for continuar o desenvolvimento do projeto, tenha noção do que se tem de fazer para cada requisito.

Aos requisitos que já existiam foram adicionados novos requisitos que foram levantados depois da análise dos questionários feitos aos utilizadores (ver secção 7). Ficou também como trabalho futuro, a especificação mais detalhada destes requisitos, igual a todos os outros que foram detalhados na secção 3, ou seja, a definição detalhada do requisito, a criação de um ou mais modelos de processos e uma ou mais **USs**.

Ficou para trabalho futuro, testar a aplicação a nível de otimização de perguntas.

Identificador	Descrição
HN-14 Social Media Share Feature	As an User I want to be able to share my events in Facebook, Twitter and LinkedIn
HN-70 Attendance QRCode	As an Association's employee I want to be able to register users' attendance by generating an QRCode that the users can scan at the end of the event
HN-38 Notifications (Mobile)	As an User I want to be notified when a big change happens (for example a project that I'm enrolled creates new slots or deletes existing one)
HN-9 Rating Projects	As an User I want to be able to rate a Project
HN-24 Reward System	As an User I want to be able to win rewards by completing an event, by rating a Project and many more accomplishments that will be defined in the future

Tabela 8.1: User Stories para trabalho futuro

BIBLIOGRAFIA

- [1] J. M. Lourenço. *The NOVAtesis L^AT_EX Template User's Manual*. NOVA University Lisbon. 2021. URL: <https://github.com/joaomlourenco/novathesis/raw/master/template.pdf> (ver p. ii).
- [2] WHE. *wehelpukraine.org*. <https://wehelpukraine.org/Home.aspx> (ver pp. 1, 20).
- [3] P. Vora. «Chapter 8 - Rich Internet Applications». Em: *Web Application Design Patterns*. Ed. por P. Vora. Interactive Technologies. Boston: Morgan Kaufmann, 2009, pp. 225–258. ISBN: 978-0-12-374265-0. DOI: <https://doi.org/10.1016/B978-0-12-374265-0.00008-6>. URL: <https://www.sciencedirect.com/science/article/pii/B9780123742650000086> (ver p. 4).
- [4] React. *React – A JavaScript library for building user interfaces*. <https://reactjs.org/> (ver p. 5).
- [5] *The web framework for perfectionists with deadlines | Django*. <https://www.djangoproject.com/> (ver p. 5).
- [6] R. on Rails. *Ruby on Rails — A web-app framework that includes everything needed to create database-backed web applications according to the Model-View-Controller (MVC) pattern*. <https://rubyonrails.org/> (ver p. 6).
- [7] *Choosing a Tech Stack for Your Project: [A Detailed Process]*. <https://spdload.com/blog/how-to-choose-a-tech-stack/> (ver p. 6).
- [8] Bootstrap. *Bootstrap · The most popular HTML, CSS, and JS library in the world*. <https://getbootstrap.com/> (ver p. 6).
- [9] Tailwind. *Tailwind CSS - Rapidly build modern websites without ever leaving your HTML*. <https://tailwindcss.com/> (ver p. 6).
- [10] G. Whitten. *Low-Code vs. Traditional Development: An Architect's Perspective*. <https://www.outsystems.com/blog/posts/low-code-vs-traditional-development/> (ver p. 7).
- [11] *Aplicações Empresariais | Microsoft Power Apps*. <https://powerapps.microsoft.com/pt-pt/> (ver p. 7).

- [12] UiPath *Business Automation Platform - Leader in RPA & Automation* | UiPath. <https://www.uipath.com/> (ver p. 7).
- [13] Outsystems. *High-Performance Low-Code for App Development* | OutSystems. <https://www.outsystems.com/> (ver p. 8).
- [14] G2. *Best Low-Code Development Platforms in 2023* | G2. <https://www.g2.com/categories/low-code-development-platforms> (ver p. 8).
- [15] F. Gürcan e G. Taentzer. «Using Microsoft PowerApps, Mendix and OutSystems in Two Development Scenarios: An Experience Report». Em: *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. 2021, pp. 67–72. DOI: [10.1109/MODELS-C53483.2021.00017](https://doi.org/10.1109/MODELS-C53483.2021.00017) (ver p. 9).
- [16] R. Waszkowski. «Low-code platform for automating business processes in manufacturing». Em: *IFAC-PapersOnLine* 52.10 (2019). 13th IFAC Workshop on Intelligent Manufacturing Systems IMS 2019, pp. 376–381. ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2019.10.060>. URL: <https://www.sciencedirect.com/science/article/pii/S2405896319309152> (ver p. 9).
- [17] Outsystems. *Forge Homepage: Assets* - OutSystems. <https://www.outsystems.com/forge/> (ver p. 10).
- [18] Atlassian. *Jira | Issue & Project Tracking Software* | Atlassian. <https://www.atlassian.com/software/jira> (ver p. 16).
- [19] Atlassian. *Confluence | Your Remote-Friendly Team Workspace* | Atlassian. <https://www.atlassian.com/software/confluence> (ver p. 17).
- [20] Bizagi *Automation Platform*. <https://www.bizagi.com/en/platform> (ver p. 17).
- [21] Postman *API Platform*. <https://www.postman.com/> (ver p. 18).
- [22] *Chrome Extension* | Scribe. <https://get.scribeshow.com/install-extension/> (ver p. 18).
- [23] *Lighthouse overview - Chrome for Developers*. <https://developer.chrome.com/docs/lighthouse/overview/> (ver pp. 18, 80).
- [24] *Apache JMeter - Apache JMeter™*. <https://jmeter.apache.org/> (ver p. 19).
- [25] *2020-Scrum-Guide-US.pdf*. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf> (ver p. 19).
- [26] *Requisitos funcionais e não funcionais: o que são?* <https://www.mestresdawe.com.br/tecnologias/requisitos-funcionais-e-nao-funcionais-o-que-sao> (ver p. 20).
- [27] Freerice. *Freerice*. <https://freerice.com/> (ver p. 21).
- [28] S. T. Meal. *ShareTheMeal*. <https://sharethemeal.org/en/> (ver p. 21).

-
- [29] C. Miles. *Charity Miles App | Walk, Run, Bike for a Causes*. <https://charitymiles.org/> (ver p. 21).
- [30] J. Torrent-Sellens et al. «Using Digital Platforms to Promote Blood Donation: Motivational and Preliminary Evidence from Latin America and Spain». Em: *International Journal of Environmental Research and Public Health* 18.8 (2021). ISSN: 1660-4601. DOI: [10.3390/ijerph18084270](https://doi.org/10.3390/ijerph18084270). URL: <https://www.mdpi.com/1660-4601/18/8/4270> (ver p. 22).
- [31] S. I. «Software Engineering. 8th ed.» Em: ed. por A.-W. H. E. 2007. Harlow England: Addison-Wesley, 2007. Cap. 6, page 119-120 (ver pp. 24, 44).
- [32] *Outsystems - Load Balancer*. https://success.outsystems.com/documentation/11/setup_and_maintain_your_outsystems_infrastructure/setting_up_outsystems/possible_setups_for_an_outsystems_infrastructure/high_availability_and_scalability_strategies/ (ver p. 46).
- [33] Google. *Autenticação — Centro de segurança da Google*. <https://safety.google/authentication/> (ver p. 49).
- [34] Google. *Como usar o OAuth 2.0 para acessar as APIs do Google | Authorization | Google Developers*. <https://developers.google.com/identity/protocols/oauth2?hl=pt-br> (ver p. 50).
- [35] *reCAPTCHA*. <https://www.google.com/recaptcha/about/> (ver p. 51).
- [36] *Image Optimizer and Compressor lib - Overview | OutSystems*. <https://www.outsystems.com/forge/component-overview/10676/image-optimizer-and-compressor-lib> (ver p. 52).
- [37] *About Touch ID advanced security technology - Apple Support*. <https://support.apple.com/en-us/HT204587> (ver p. 52).
- [38] *What is Web Scraping and What is it Used For? | ParseHub*. <https://www.parsehub.com/blog/what-is-web-scraping/> (ver p. 70).
- [39] *A Complete Web Application Testing Checklist | BrowserStack*. <https://www.browserstack.com/guide/web-application-testing-checklist> (ver p. 75).
- [40] *Injection and Cross Site Script - XSS - OutSystems Best Practices*. https://success.outsystems.com/documentation/best_practices/security/injection_and_cross_site_script_xss/ (ver p. 85).

PERGUNTAS DO QUESTIONÁRIOS

Foi fácil encontrar onde criar redes?

- ☐ Não
- ☐ Sim

Figura A.1: Questionário - Pergunta 1

Achaste fácil a gestão de redes?

- ☐ Não
- ☐ Mais ou menos
- ☐ Sim

Figura A.2: Questionário - Pergunta 2

(Esta pergunta é só para quem criou um Associação)

Foi fácil perceber onde se criavam novos projetos?

- ☐ Não
- ☐ Mais ou menos
- ☐ Sim

Figura A.3: Questionário - Pergunta 3

(Esta pergunta é só para quem criou um Associação)

Foi fácil encontrar onde gerir projetos (sem a utilização do manual de utilização)?

- ☐ Não
- ☐ Mais ou menos
- ☐ Sim

Figura A.4: Questionário - Pergunta 4

(Esta pergunta é só para quem criou um Associação)

Achaste fácil gerir os projetos?

- ☐ Não
- ☐ Mais ou menos
- ☐ Sim

Figura A.5: Questionário - Pergunta 5

(Esta pergunta é só para quem criou um Associação)

Foi fácil criar eventos?

- ☐ Não
- ☐ Mais ou menos
- ☐ Sim

Figura A.6: Questionário - Pergunta 6

Achaste fácil inscrever em projetos?

- ☐ Não
- ☐ Mais ou menos
- ☐ Sim

Figura A.7: Questionário - Pergunta 7

Achaste fácil inscrever em eventos?

- ☐ Não
- ☐ Mais ou menos
- ☐ Sim

Figura A.8: Questionário - Pergunta 8

APÊNDICE A. PERGUNTAS DO QUESTIONÁRIOS

Achaste que os projetos sugeridos (no final do ecrã de detalhe de um projeto) fazia sentido face às categorias que escolheste?

- ☐ Não
- ☐ Alguns
- ☐ Sim

Figura A.9: Questionário - Pergunta 9

Foi necessário a constante utilização do manual de utilização?

- ☐ Sim
- ☐ Por vezes
- ☐ Nunca usei

Figura A.10: Questionário - Pergunta 10

De 1 a 5 (sendo 1 muito lenta e 5 muito rápida) a nível de desempenho em que nível achas que aplicação está?

- | | 1 | 2 | 3 | 4 | 5 | |
|-------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|--------------|
| Muito lenta | ☐ | ☐ | ☐ | ☐ | ☐ | Muito rápida |

Figura A.11: Questionário - Pergunta 11

De 1 a 5 (sendo 1 muito difícil e 5 muito fácil) com que facilidade conseguiste utilizar a aplicação (sem recorrer ao manual de utilização).

- | | 1 | 2 | 3 | 4 | 5 | |
|---------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-------------|
| Muito difícil | ☐ | ☐ | ☐ | ☐ | ☐ | Muito fácil |

Figura A.12: Questionário - Pergunta 12

Achaste que a aplicação tinha algum problema critico? Se sim qual(quais)?

A sua resposta

Figura A.13: Questionário - Pergunta 13

O que achas que seria um bom requisito a ser implementado para esta aplicação
(Funcionalidades que deveria ter)?

A sua resposta

Figura A.14: Questionário - Pergunta 14

O que mais gostaste da aplicação?

A sua resposta

Figura A.15: Questionário - Pergunta 15

O que menos gostaste da aplicação?

A sua resposta

Figura A.16: Questionário - Pergunta 16

O que achas que deveria ser melhorado?

A sua resposta

Figura A.17: Questionário - Pergunta 17

PERGUNTAS SQL

Neste anexo está presente as perguntas utilizadas na procura de projetos.

Pergunta usada na lista de projetos no ecrã **Projects**:

```

SELECT
    A.[Id],
    {Network}.[Id],
    {Network}.[Name],
    A.[Name],
    A.[Description],
    {ProjectEXT}.[Image],
    Principal.[Id],
    Principal.[Name],
    AVG({Project_User}.[Rating])
FROM
    {Project} A
INNER JOIN
    {Network} ON A.[AssociationId] = {Network}.[Id]
LEFT JOIN
    {ProjectEXT} ON A.[Id] = {ProjectEXT}.[ProjectId]
LEFT JOIN
    {Project_User} ON (A.[Id] = {Project_User}.[ProjectId]
        AND {Project_User}.[Rating] <> 0)
LEFT JOIN
    {Project_Category} ON A.[Id] = {Project_Category}.[ProjectId]
LEFT JOIN
    {Category} ON {Project_Category}.[CategoryId] = {Category}.[Id]
INNER JOIN -- Principal Category
    {Category} AS Principal ON
        {Project_Category}.[CategoryId] = Principal.[Id] AND
        {Project_Category}.[IsPrincipal] = 1
WHERE
    (
        (CAST({Project_User}.[Rating] as VARCHAR(1))

```

```

        IN (@RatingFilter) AND
        @RatingFilter <> '')
    OR
        @RatingFilter = ''
    )
AND
    (
        ({Category}.[Id] IN (@CategoryList) AND @CategoryList <> '')
    OR
        @CategoryList = ''
    )
GROUP BY
    A.[Id], {Network}.[Id], {Network}.[Name],
    A.[Name], A.[Description], {ProjectEXT}.[Image],
    Principal.[Id], Principal.[Name]
HAVING
    (
        (A.[Name] LIKE '%' + @Search + '%' AND @Search <> '')
    OR
        @Search = ''
    )
ORDER BY (SELECT NULL) OFFSET @StartIndex
        ROWS FETCH NEXT @MaxRecords ROWS ONLY;

```

Pergunta usada na lista de projetos no ecrã **MyProjects**:

```

SELECT
    A.[Id],
    {Project_User}.[Id],
    {Project_User}.[CompanyId],
    {Project_User}.[StatusId],
    {Project_User}.[IsFavorite],
    {Project_User}.[Rating],
    {Project_User}.[Comment],
    {Network}.[Id],
    {Network}.[Name],
    A.[Name],
    A.[Description],
    {ProjectEXT}.[Image],
    Principal.[Id],
    Principal.[Name],
    AVG({Project_User}.[Rating])
FROM
    {Project} A
INNER JOIN
    {Network} ON A.[AssociationId] = {Network}.[Id]

```

```
LEFT JOIN
    {ProjectEXT} ON A.[Id] = {ProjectEXT}.[ProjectId]
INNER JOIN
    {Project_User} ON A.[Id] = {Project_User}.[ProjectId]
LEFT JOIN
    {Project_Category} ON A.[Id] = {Project_Category}.[ProjectId]
LEFT JOIN
    {Category} ON {Project_Category}.[CategoryId] = {Category}.[Id]
INNER JOIN -- Principal Category
    {Category} AS Principal
        ON {Project_Category}.[CategoryId] = Principal.[Id] AND
        {Project_Category}.[IsPrincipal] = 1
WHERE
    {Project_User}.[UserId] = @UserId
AND
    (
        ({Category}.[Id] = @CategoryId AND @CategoryId IS NOT NULL)
        OR
        @CategoryId = 0
    )
GROUP BY
    A.[Id], {Project_User}.[Id], {Project_User}.[UserId],
    {Project_User}.[CompanyId], {Project_User}.[StatusId],
    {Project_User}.[IsFavorite], {Project_User}.[Rating],
    {Project_User}.[Comment], {Network}.[Id], {Network}.[Name],
    A.[Name], A.[Description], Principal.[Id], Principal.[Name],
    {ProjectEXT}.[Image]
HAVING
    (
        (A.[Name] LIKE '%' + @Search + '%' AND @Search <> '')
        OR
        @Search = ''
    )
ORDER BY (SELECT NULL) OFFSET @StartIndex
        ROWS FETCH NEXT @MaxRecords ROWS ONLY;
```

Perguntas usadas na lista de projetos sugeridos no ecrã **Project_Details**:

```
SELECT TOP 10
    {Project}.[Id],
    {Project}.[Name],
    {Project}.[Description],
    {Network}.[Name],
    {ProjectEXT}.[Image],
    {Project_Category}.[Id],
    AVG({Project_User}.[Rating])
```

```

FROM
    {Project}
LEFT JOIN
    {ProjectEXT} ON {Project}.[Id] = {ProjectEXT}.[ProjectId]
INNER JOIN
    {Network} ON {Project}.[AssociationId] = {Network}.[Id]
INNER JOIN
    {Project_Category}
    ON {Project}.[Id] = {Project_Category}.[ProjectId] AND
        {Project_Category}.[IsPrincipal] = 1
LEFT JOIN
    {Project_User}
    ON ({Project}.[Id] = {Project_User}.[ProjectId])
WHERE
    {Project}.[Id] <> @ProjectId AND
    {Project_Category}.[CategoryId] = @CategoryId AND
    {Project}.[Id] NOT IN (
        SELECT
            {Project_User}.[ProjectId]
        FROM
            {Project_User}
        WHERE
            {Project_User}.[UserId] = @UserId
    )
GROUP BY
    {Project}.[Id],
    {Project}.[Name],
    {Project}.[Description],
    {Network}.[Name],
    {ProjectEXT}.[Image],
    {Project_Category}.[Id],
    {Project_User}.[Rating]

SELECT TOP 10
    {Project}.[Id],
    {Project}.[Name],
    {Project}.[Description],
    {Network}.[Name],
    {ProjectEXT}.[Image],
    {Project_Category}.[Id],
    AVG({Project_User}.[Rating])
FROM
    {Project}
LEFT JOIN
    {ProjectEXT} ON {Project}.[Id] = {ProjectEXT}.[ProjectId]

```

```
INNER JOIN
    {Network} ON {Project}.[AssociationId] = {Network}.[Id]
INNER JOIN
    {Project_Category}
        ON {Project}.[Id] = {Project_Category}.[ProjectId] AND
        {Project_Category}.[IsPrincipal] = 1
LEFT JOIN
    {Project_User} ON {Project}.[Id] = {Project_User}.[ProjectId]
WHERE
    {Project_Category}.[CategoryId] IN (@CategoryList_Text) AND
    {Project}.[Id] NOT IN (
        SELECT
            {Project_User}.[ProjectId]
        FROM
            {Project_User}
        WHERE
            {Project_User}.[UserId] = @UserId
    UNION
        SELECT TOP 1
            {Project}.[Id]
        FROM
            {Project}
        WHERE
            {Project}.[Id] = @ProjectId
    )
GROUP BY
    {Project}.[Id],
    {Project}.[Name],
    {Project}.[Description],
    {Network}.[Name],
    {ProjectEXT}.[Image],
    {Project_Category}.[Id],
    {Project_User}.[Rating]
```

MANUAL DE UTILIZAÇÃO - CRIAÇÃO DE PROJETOS



Figura II.1: Manual de utilização - Criação de Projetos (passo 1)

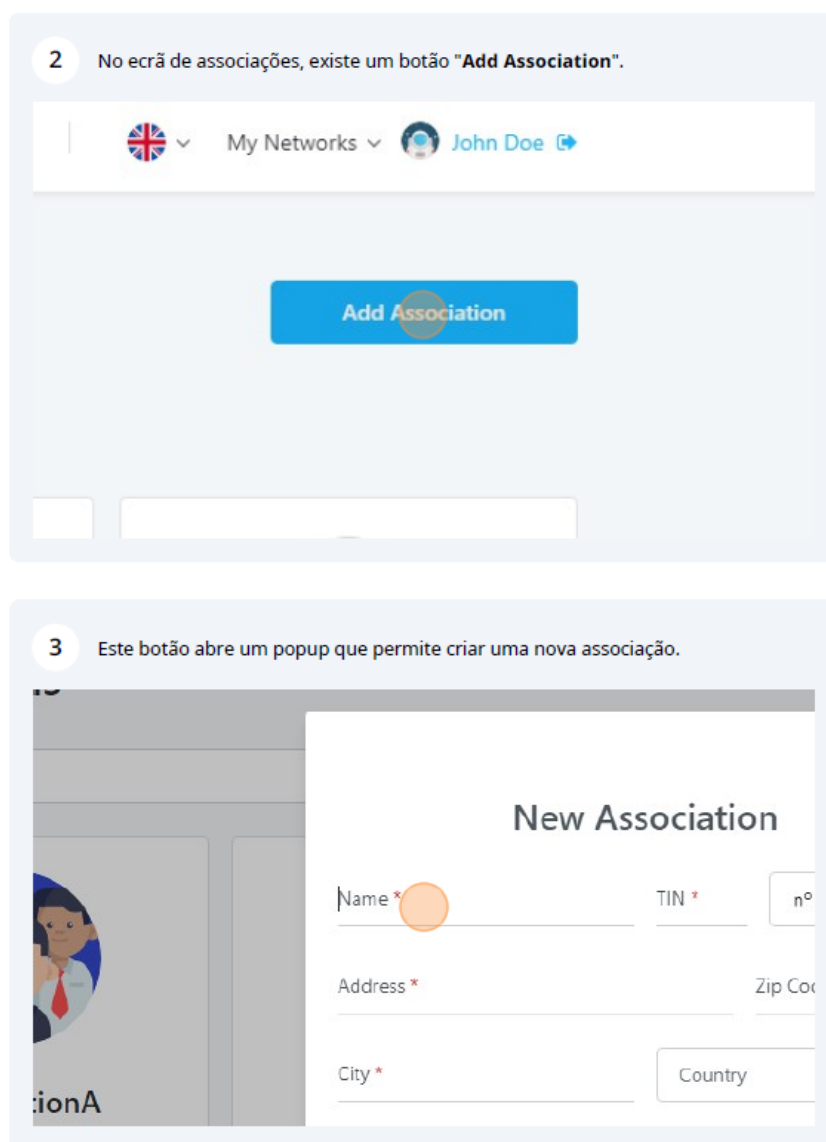


Figura II.2: Manual de utilização - Criação de Projetos (passos 2 e 3)

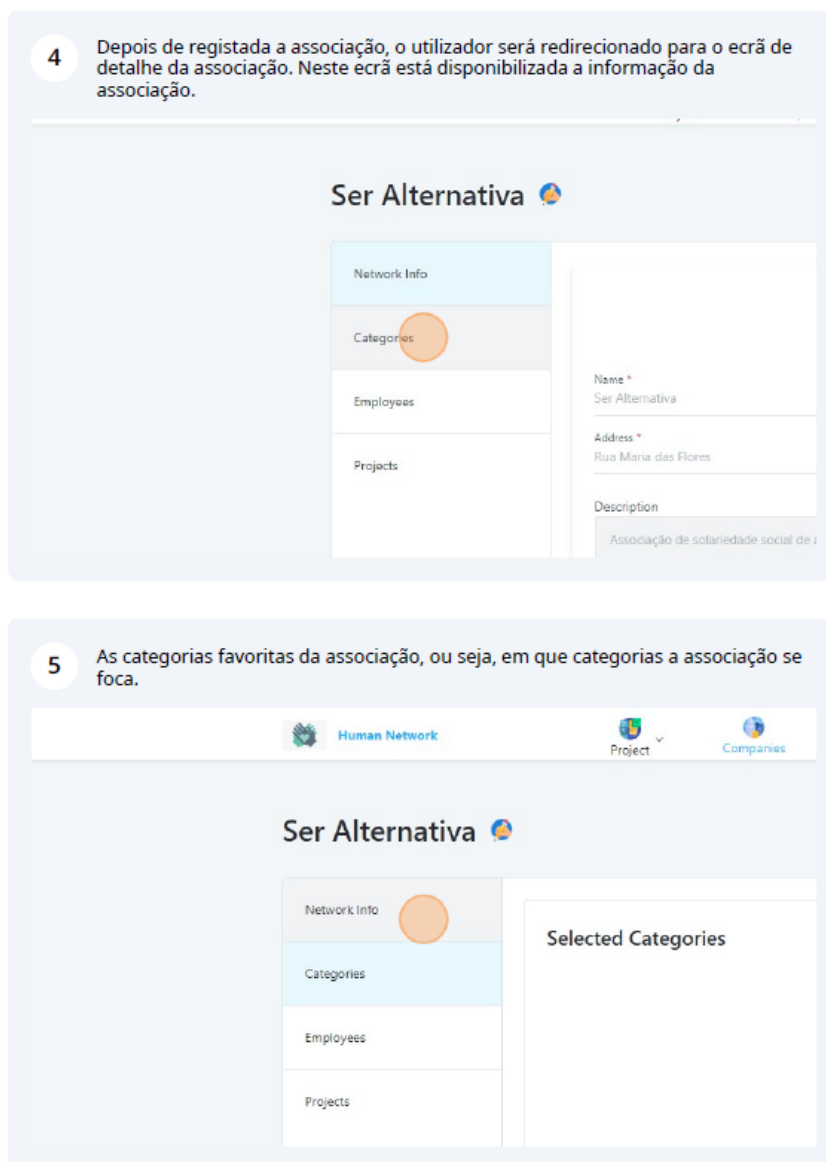


Figura II.3: Manual de utilização - Criação de Projetos (passos 4 e 5)

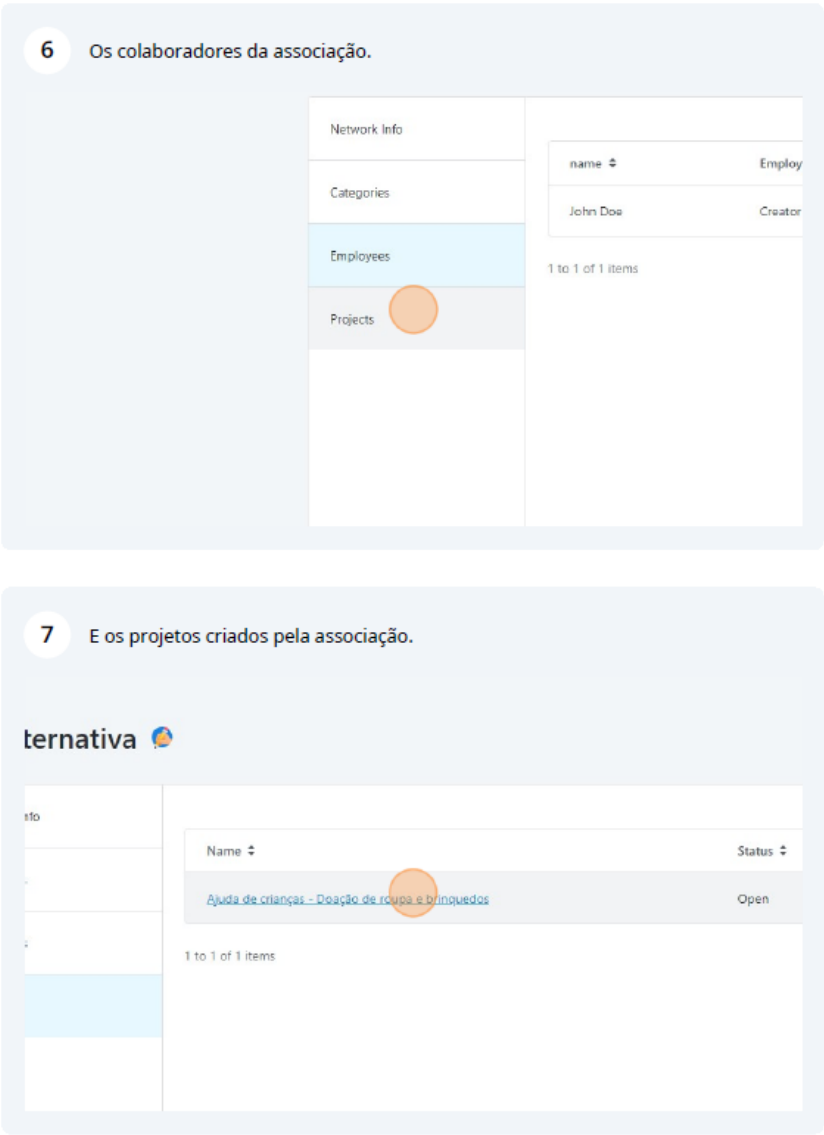


Figura II.4: Manual de utilização - Criação de Projetos (passos 6 e 7)

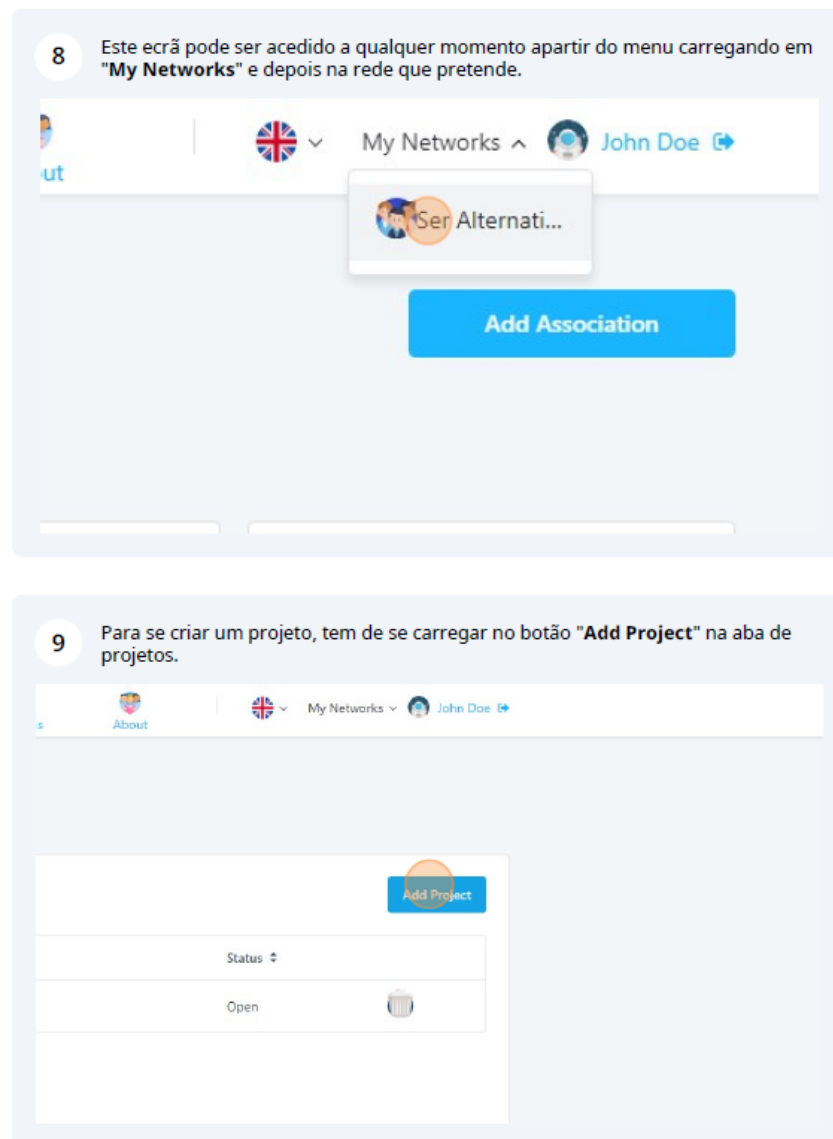


Figura II.5: Manual de utilização - Criação de Projetos (passos 8 e 9)

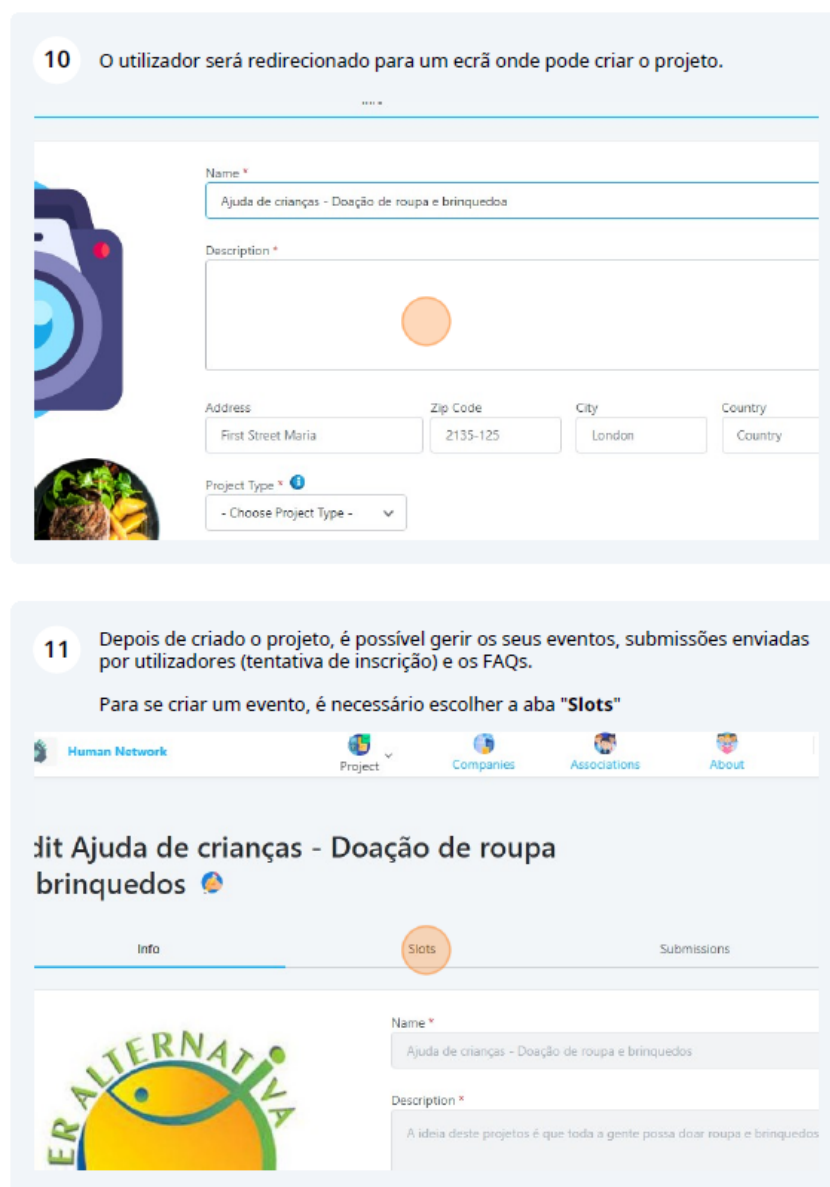


Figura II.6: Manual de utilização - Criação de Projetos (passos 10 e 11)

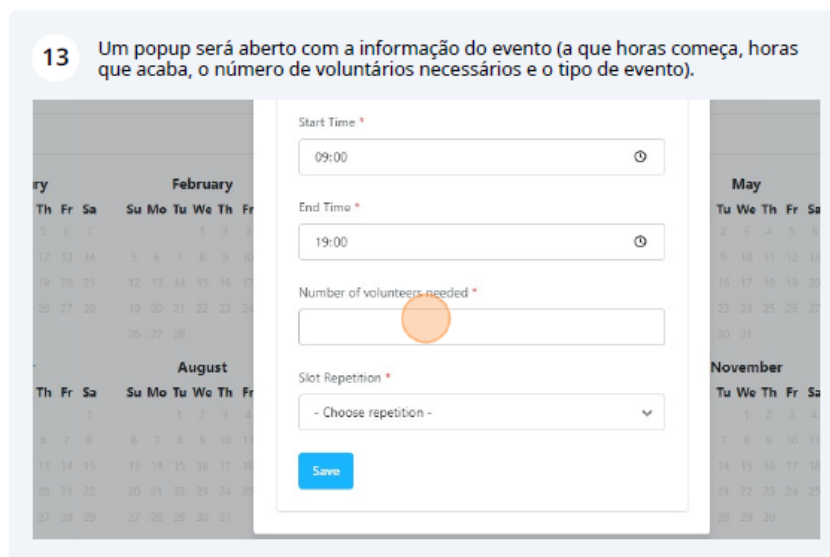
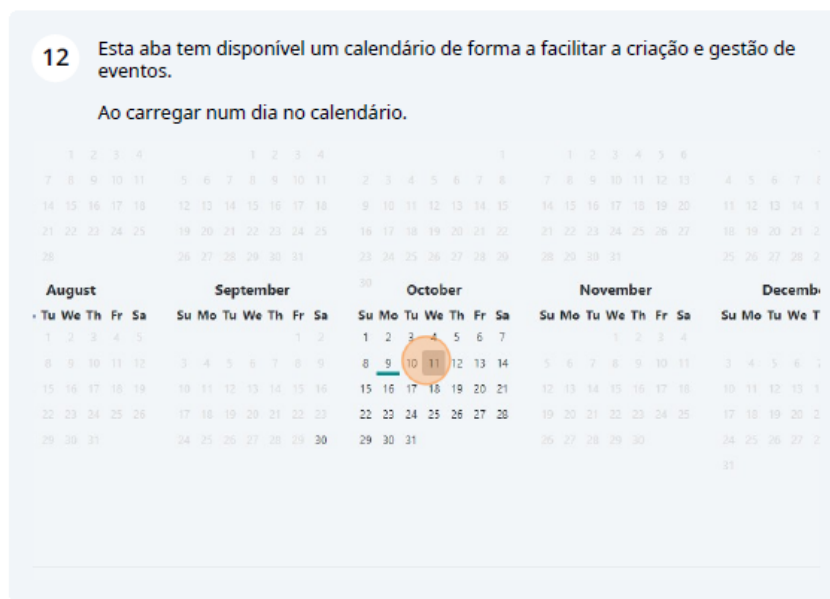


Figura II.7: Manual de utilização - Criação de Projetos (passos 12 e 13)



Figura II.8: Manual de utilização - Criação de Projetos (passos 14 e 15)

MANUAL DE UTILIZAÇÃO - INSCRIÇÃO EM PROJETOS

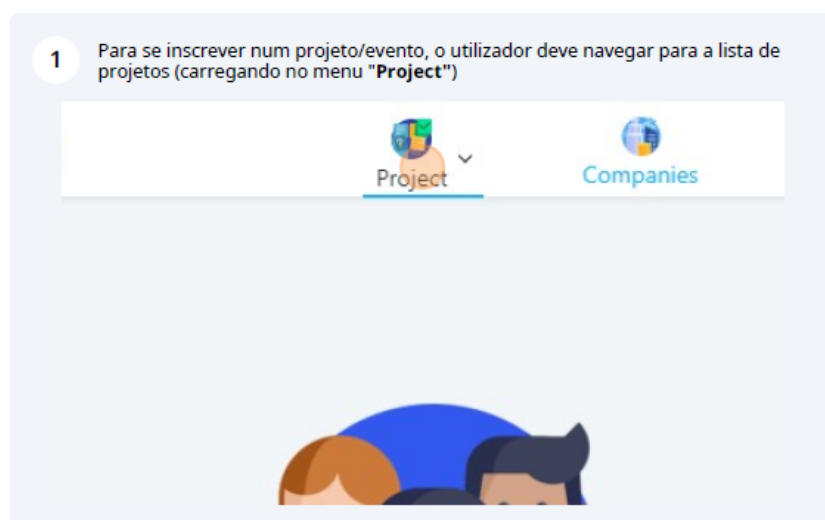


Figura III.1: Manual de utilização - Inscrição em Projetos (passo 1)

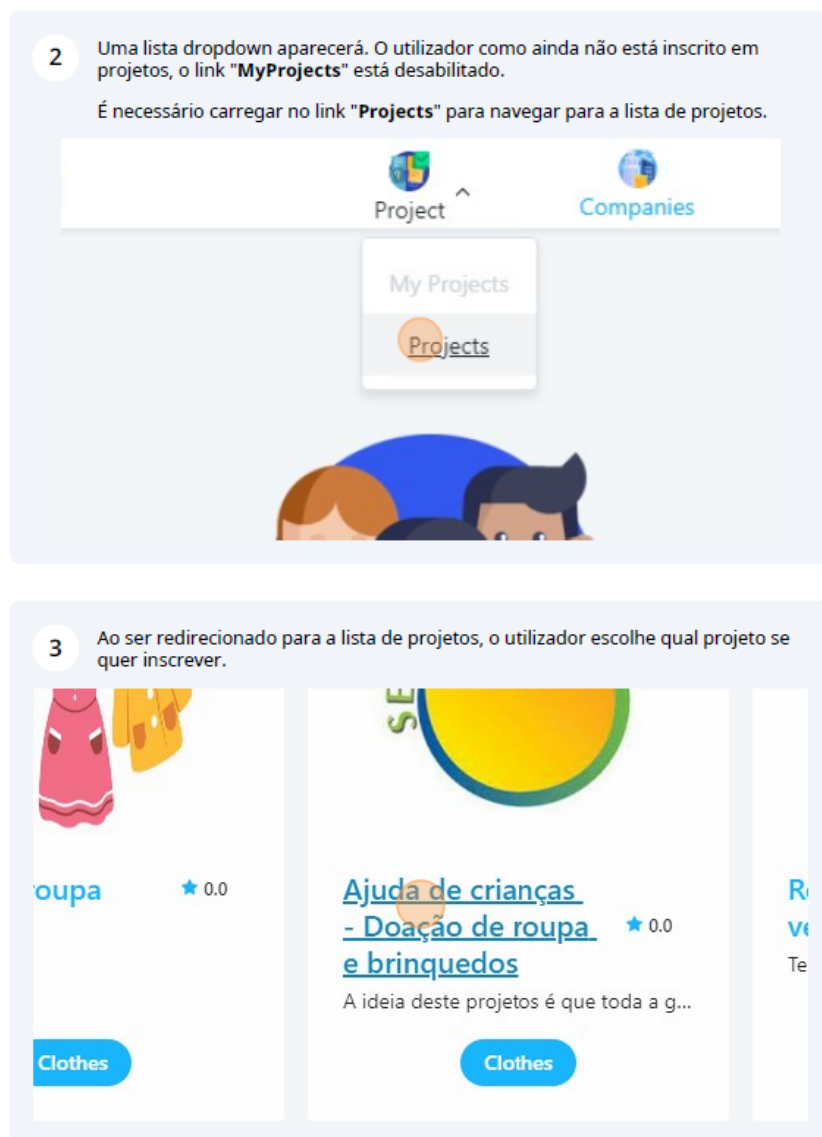


Figura III.2: Manual de utilização - Inscrição em Projetos (passos 2 e 3)

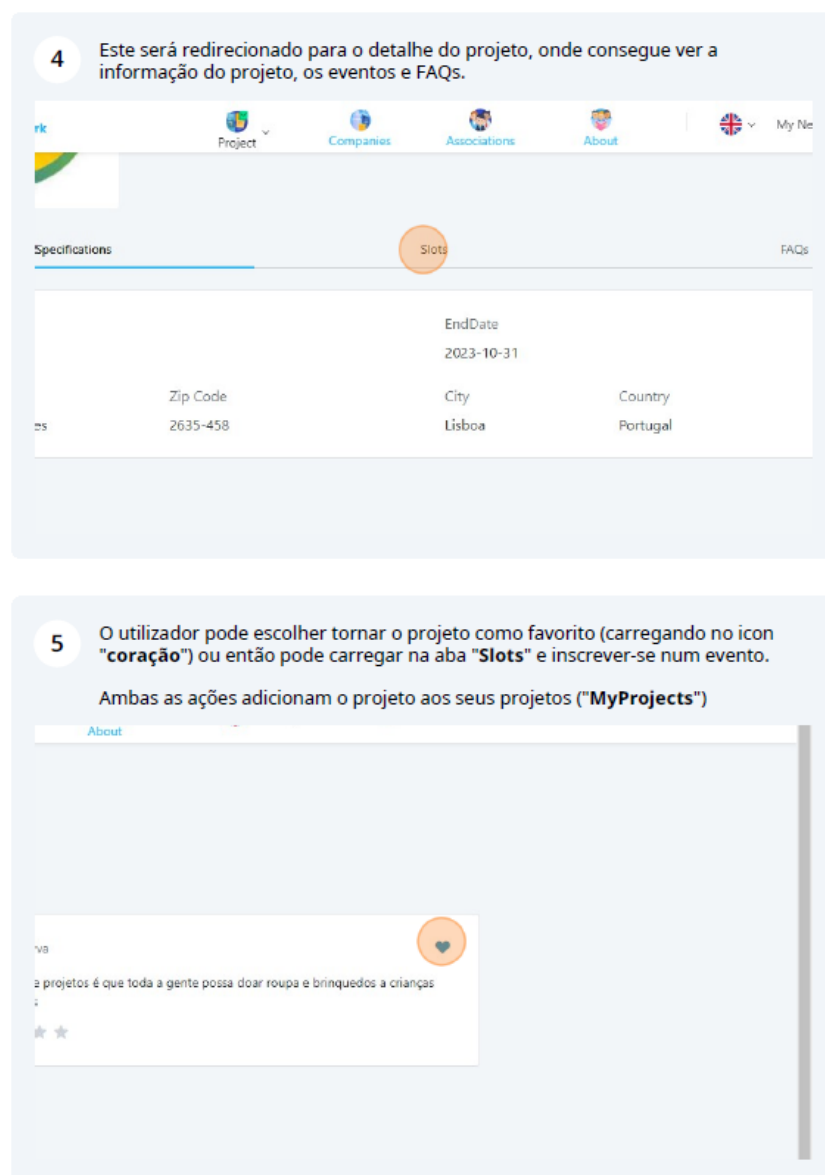


Figura III.3: Manual de utilização - Inscrição em Projetos (passos 4 e 5)

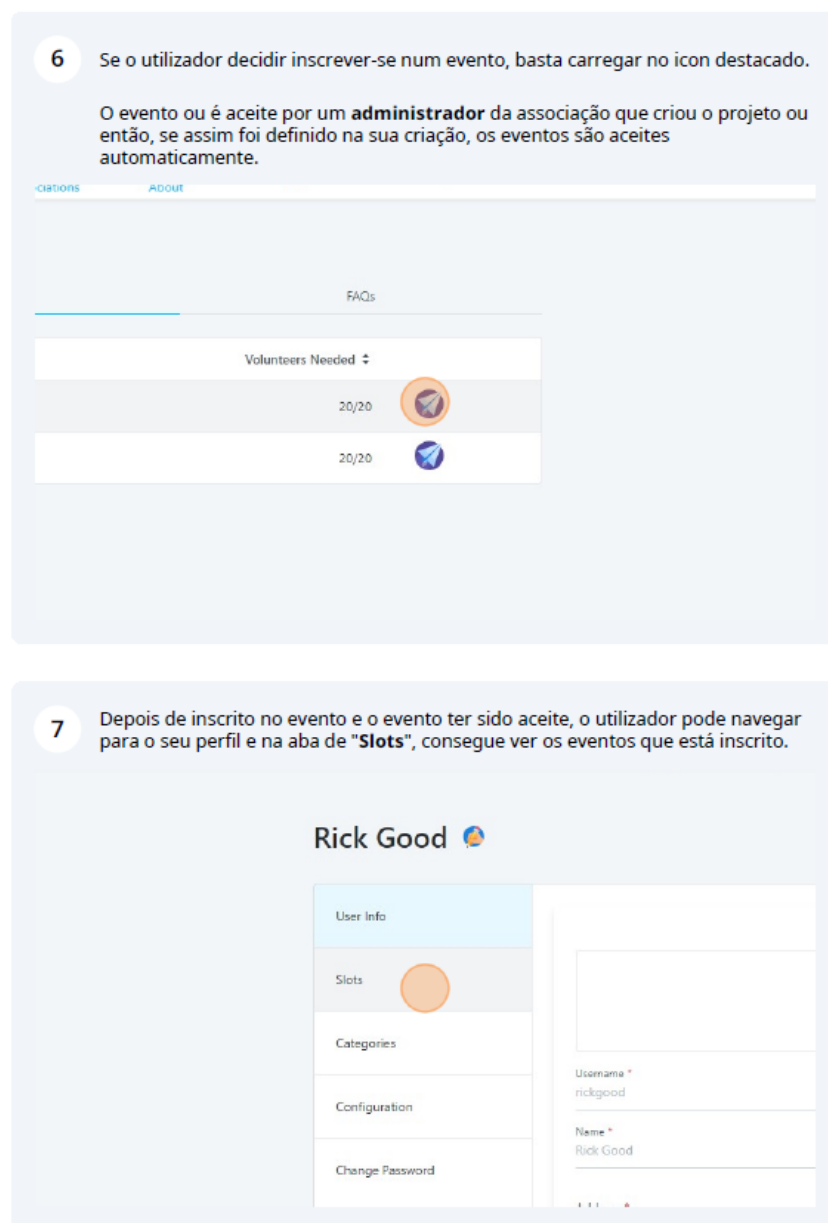


Figura III.4: Manual de utilização - Inscrição em Projetos (passos 6 e 7)

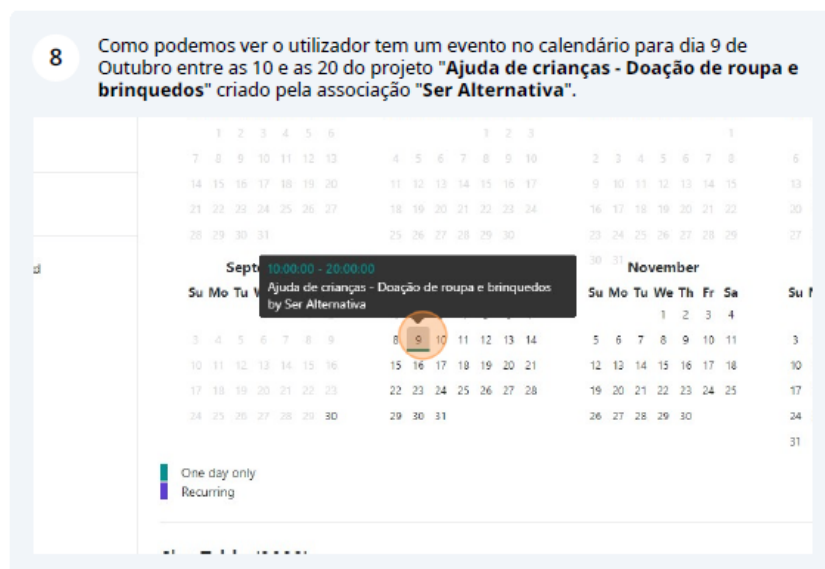


Figura III.5: Manual de utilização - Inscrição em Projetos (passos 8 e 9)

MANUAL DE UTILIZAÇÃO - PERFIL DE UTILIZADOR

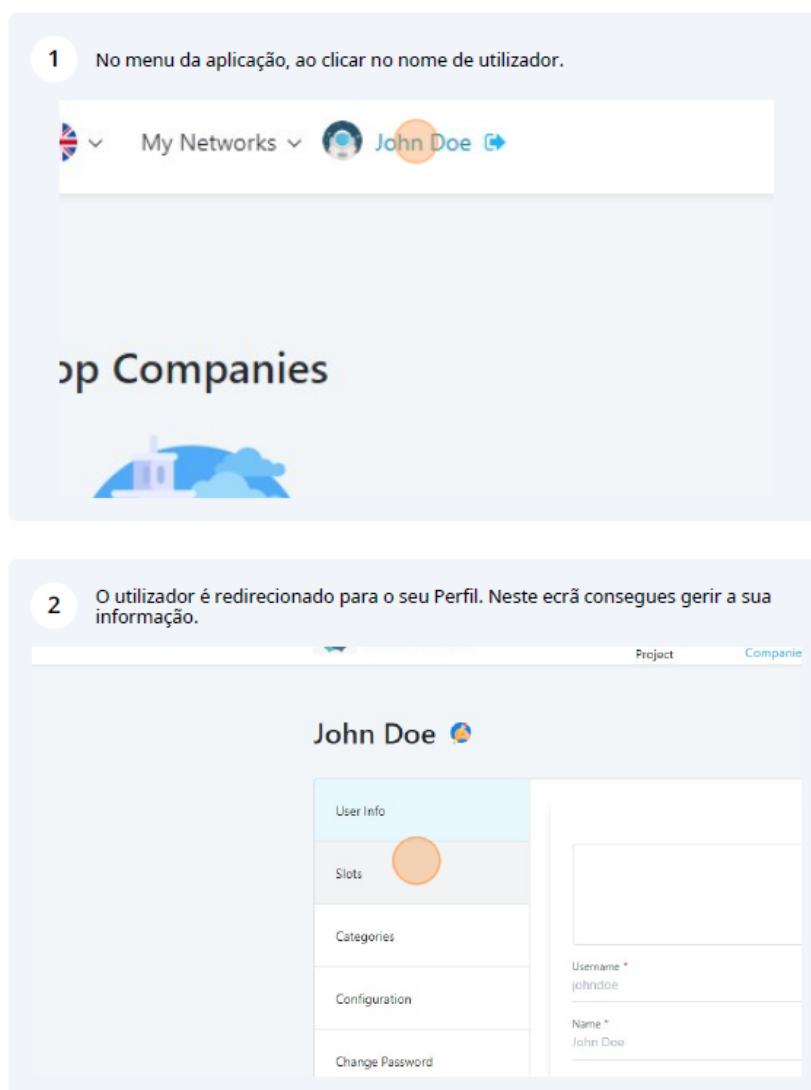
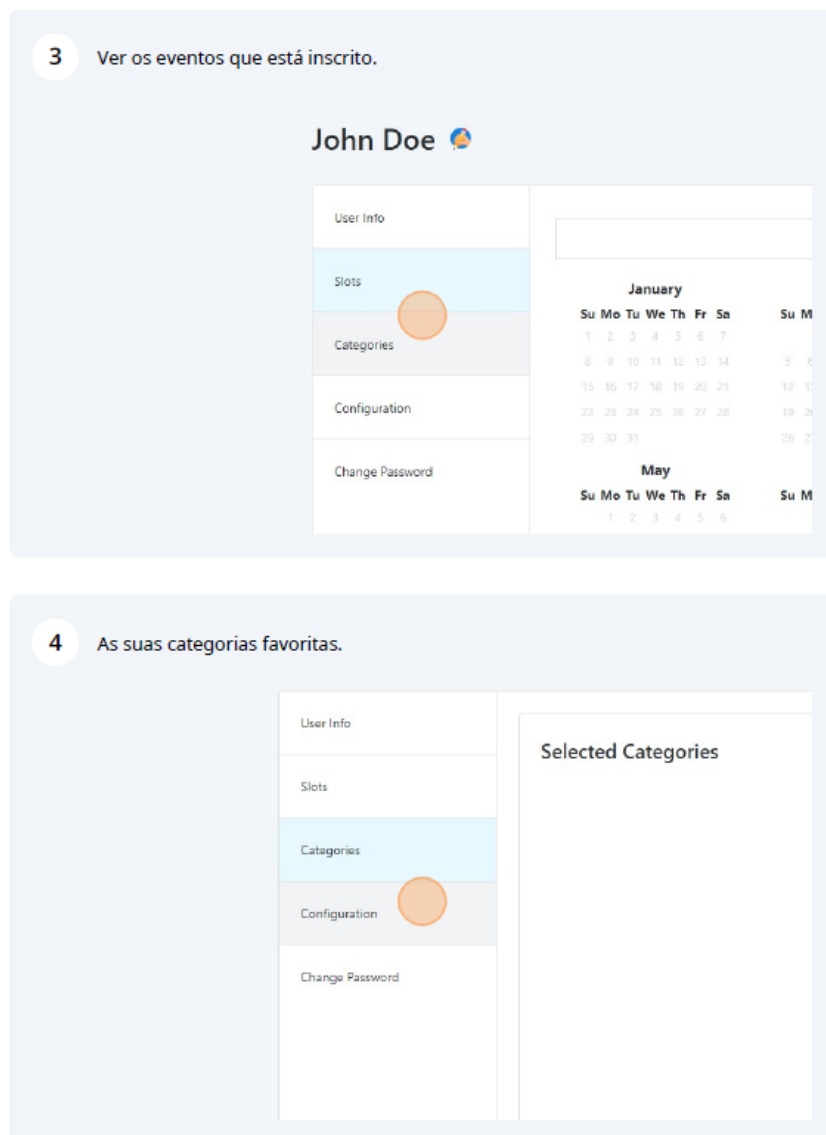


Figura IV.1: Manual de utilização - Perfil (passos 1 e 2)



User Info

Slots

Categories

Configuration

Change Password

Selected Categories

Figura IV.2: Manual de utilização - Perfil (passos 3 e 4)

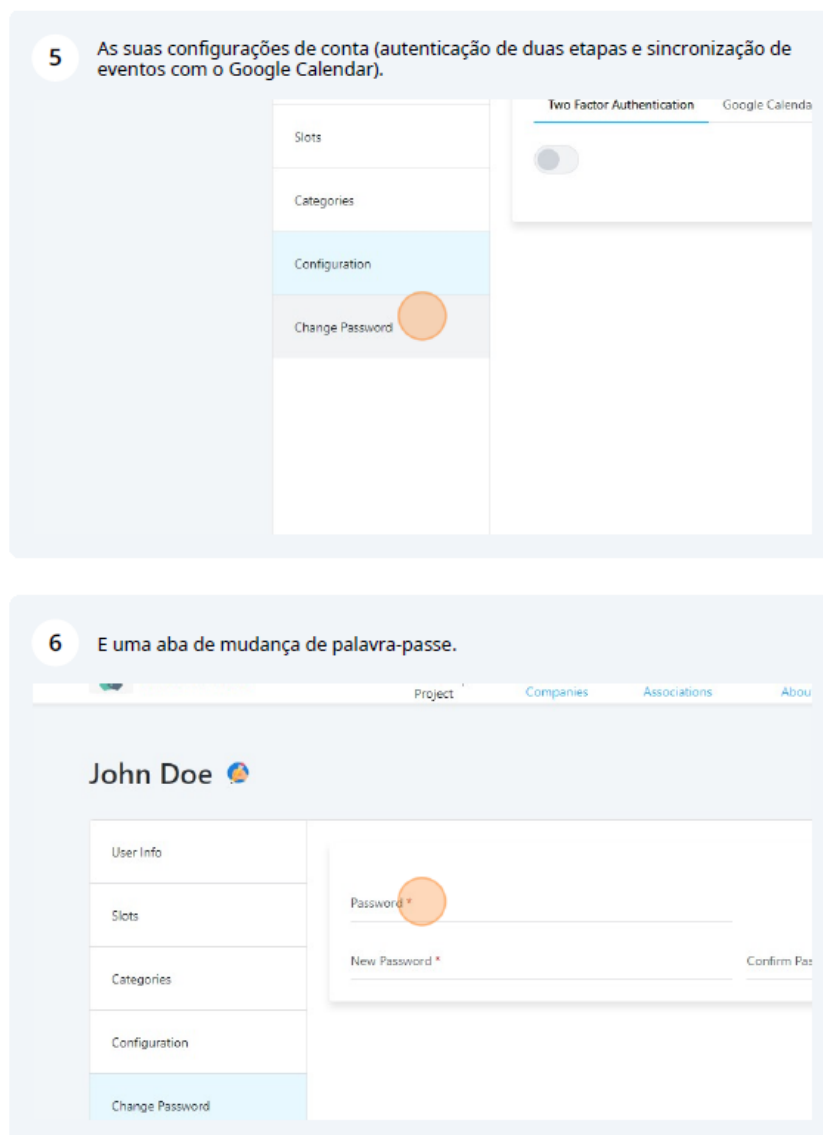


Figura IV.3: Manual de utilização - Perfil (passos 5 e 6)



ROBOTS.TXT

Sitemap: https://kinetit.outsystemscloud.com/HumanNetwork_UI/sitemap.xml

User-agent:*

Allow: /

User-agent: **Adsbob-Google**

Allow: **/store**

User-agent: **Googlebot**

Allow: /

Disallow: **/profile**

SITEMAP.XML

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <urlset xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9 http://www.
   ↳ sitemaps.org/schemas/sitemap/0.9/sitemap.xsd" xmlns:xsi="http://www.w3.org/2001/
   ↳ XMLSchema-instance" xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
4
5 <!-- created with Free Online Sitemap Generator www.xml-sitemaps.com -->
6
7 <url>
8   <loc>https://kinetit.outsystemscloud.com/HumanNetwork_UI/Home</loc>
9   <lastmod>2023-09-28T21:51:28+00:00</lastmod>
10 </url>
11
12 </urlset>
```



Human
Network
2023

Federico
Lopes

NEW

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY