



GONÇALO FILIPE
CARAÇA PIRES

**AS TECNOLOGIAS DE
INFORMAÇÃO COMO
FERRAMENTA DE TRABALHO
COM AS PESSOAS EM
SITUAÇÃO DE SEM-ABRIGO**

Relatório de Dissertação de investigação do
Mestrado em Gestão de Sistemas de Informação

ORIENTADOR

Professora Doutora Leonilde Reis

Junho 2021

GONÇALO FILIPE
CARAÇA PIRES

**AS TECNOLOGIAS DE
INFORMAÇÃO COMO
FERRAMENTA DE TRABALHO
COM AS PESSOAS EM
SITUAÇÃO DE SEM-ABRIGO**

Junho 2021

Agradecimentos

A realização do presente trabalho só foi possível graças ao contributo de várias pessoas, pelas quais tive o privilégio de ser apoiado.

Começo por agradecer à minha orientadora, Professora Doutora Leonilde Reis, que me introduziu a este tema e fez os primeiros contactos com as organizações, pela disponibilidade, acompanhamento, motivação e orientação, sem as quais este trabalho não seria possível. O meu obrigado por acreditar sempre que seria capaz de realizar este trabalho com sucesso.

À Professora Doutora Maria Clara Silveira pela disponibilidade demonstrada e conhecimento partilhado e ao Professor Doutor Rui Neves Madeira pela motivação e por todas as cuidadosas sugestões e críticas.

Aos senhores Armando Gomes, Bruno Vilela e Ana Gaspar da NPISA Barreiro, pela disponibilidade, partilha de experiência e material de apoio facultado. Sem a sua ajuda não seria possível realizar este trabalho.

Ao Carlos e à Sara, pela entajuda e horas de discussão partilhadas.

À minha namorada, Mariana, pela paciência, incentivo e força para continuar, especialmente nos dias em que o desânimo era maior.

Aos meus camaradas da Força Aérea Portuguesa com quem trabalho diariamente, pelo apoio e incentivo para desenvolver o trabalho.

Resumo

A temática da Pessoa em situação de sem-abrigo é poucas vezes abordada no contexto das Tecnologias de Informação e Comunicação, o que, junto com a importância social que este tema tem, torna pertinente o seu estudo. Vários aspetos deste tema foram analisados, como por exemplo a caracterização das Pessoas em situação de sem-abrigo, as organizações que diariamente trabalham com elas, bem como o impacto positivo que as Tecnologias de Informação e Comunicação podem ter nestas organizações.

Os objetivos do presente projeto são otimizar o atual Sistema de Informação, que permita um fácil acesso à informação existente sobre as Pessoas em situação de sem-abrigo que são intervencionadas pelas organizações e, posteriormente, que seja possível adicionar e consultar essa informação, armazenando-a de maneira segura. É pretendido que esse Sistema de Informação seja pensado de maneira a ser possível construí-lo tanto num formato WebApp como mobile, para ser possível de aceder tanto num computador como em dispositivos móveis. A metodologia utilizada foi a Design Science Research, que permitiu suportar o desenvolvimento do projeto, após serem identificadas as necessidades da organização e definidos os problemas a abordar. Com base nesta metodologia, descreve-se todo o processo, desde a formulação do problema, à pesquisa efetuada e compreensão da temática, o debate, análise e validação dos requisitos necessários para os intervenientes (recorrendo a notação Unified Modeling Language), as características técnicas do protótipo até ao desenvolvimento do modelo de dados.

Os principais resultados centram-se na possibilidade de caracterizar detalhadamente a Pessoa em situação de sem-abrigo, ao conter uma grande variedade de dados sobre essa pessoa, armazenar esses dados e tratá-los para assim poder ser uma importante ferramenta de análise e apoio à gestão. Ao agrupar numa mesma plataforma toda a informação disponível sobre as Pessoas em situação de sem-abrigo e poder agregar essa informação de diferentes locais e a qualquer hora, através de dispositivos de formato móvel, podem-se criar novas dinâmicas de trabalho para os funcionários da organização, alargando as opções que podem tomar e fornecendo mais e melhor informação para a tomada das mesmas.

Palavras-Chave: Pessoa em situação de sem-abrigo, Linguagem de Modelação Unificada, Análise de Requisitos.

Abstract

The thematic of the homeless people is few times approached on the Information and Communication Technologies context, and that, together with the social importance on this theme, are reasons that turn their study relevant. Several aspects of this theme were analyzed, for example the characterization of the homeless people, the organizations that work with them on a daily basis and the positive impact that the Information and Communication Technologies may have on this organizations.

The goals of this project are to optimize the current Information System, that allows an easy access to the existing information about the homeless people who are intervened by the organizations, and posteriorly, be possible to add and consult that same information, storing the information on a safe way. This Information System projected is intended to be able to support both WebApp as mobile versions, because it is a goal to make the App accessible by computer as on mobile devices. The methodology used was the Design Science Research, which allowed to support the development of the project, after the organization needs are identified and the problems to approach defined. Based on this methodology of processes, the entire process will be described, from the problem formulation, the research done and the thematic understanding, the debate, analysis and the validation of the necessary requirements for the actors (using Unified Modeling Language notation to be easier to understand), the technical features of the prototype to the development of the data model.

The main results that can be extracted focus the possibility to characterize in detail the homeless person, containing a great variety of data about this person, store that data and process it so that way it can be an important tool of analysis and support to the management. By grouping on the same platform all the available information about the people and the ability to upload all that information from different places, at any time, through mobile devices, the creation of new work dynamics for the organization workers may be enabled, widening the options they may take and supplying more and better information to take them.

Keywords: Homeless People, Unified Modeling Language, Requirements analysis.

Índice Geral

Agradecimentos.....	iii
Resumo.....	iv
Abstract	v
Índice Geral.....	vi
Índice de Figuras	viii
Índice de Tabelas.....	ix
Lista de Siglas/Acrónimos	x
1. Introdução.....	1
1.1. Motivação.....	2
1.2. Objetivos	3
1.3. Estrutura	3
2. Enquadramento Teórico.....	5
2.1. Pessoa em Situação de Sem-Abrigo.....	5
2.2. Enquadramento Legal	6
2.3. Estratégia Nacional para a Integração de Pessoas Sem Abrigo	8
2.4. Sistemas de Informação	9
2.5. Segurança de Informação	10
2.6. Tecnologias de Informação e Comunicação	12
2.7. Recolha de Informação.....	14
3. Metodologia.....	15
3.1. A recolha de informação de Pessoa em situação de sem-abrigo.....	18
3.2. Engenharia de Requisitos	20
3.2.1. Início do Processo	22
3.2.2. Levantamento de Requisitos	22
3.2.3. Desenvolvimento dos Requisitos	25
3.2.4. Negociação dos Requisitos.....	25
3.2.5. Documentação de Requisitos	25
3.2.6. Validação de Requisitos	26
3.2.7. Gestão de Requisitos	26
3.3. Técnicas de Análise de Requisitos	27
3.4. Metodologia de Desenvolvimento de Software	30

3.4.1. Metodologia de processos tradicional	30
3.4.1.1. Modelo em Cascata	30
3.4.1.2. Modelo Incremental	31
3.4.1.3. Prototipagem	32
3.4.1.4. Espiral.....	33
3.4.2. Metodologia de processos especializada.....	35
3.4.2.1. Desenvolvimento baseado em componentes.....	35
3.4.2.2. Modelo de métodos formais	35
3.4.3. Metodologia de processo unificado.....	36
3.4.4. Metodologia de processo pessoal e de equipa.....	37
3.4.4.1. Processo de software pessoal	37
3.4.4.2. Processo de software de equipa.....	38
3.5. Metodologia Agile.....	38
3.5.1. <i>Extreme Programming</i>	40
3.5.2. <i>Scrum</i>	41
3.5.3. Desenvolvimento de Software <i>Lean</i>	42
3.6. Sistematização da Metodologia.....	43
4. Caracterização da organização em estudo.....	44
4.1. A instituição	44
4.2. A Problemática.....	44
4.3. Tabela de Casos de Uso	45
4.4. Diagrama de Casos de Uso.....	47
4.5. Descrição de Casos de Uso	48
4.6. Diagrama de Atividades	58
4.7. Diagrama de Estados	59
4.8. Diagrama de Sequência.....	60
4.9. Diagrama de Entidade-Relação	65
5. Análise comparativa de plataformas	67
6. Conclusões e perspectivas de trabalho futuro	70
6.1. Conclusões	70
6.2. Perspetivas de Trabalho Futuro.....	71
Referências	72

Índice de Figuras

Figura 1 – Atividades da engenharia de requisitos	21
Figura 2 – Modelo em Cascata	31
Figura 3 – Modelo Incremental.....	32
Figura 4 - Prototipagem.....	33
Figura 5 – Modelo em Espiral.....	34
Figura 6 – Metodologia do Processo Unificado.....	37
Figura 7 – O processo <i>Extreme Programming</i>	41
Figura 8 – Princípios do desenvolvimento <i>Lean</i>	42
Figura 9 – Diagrama de Casos de Uso	47
Figura 10 – Diagrama de Atividades.....	58
Figura 11 – Diagrama de Estados do Processo Individual.....	59
Figura 12 – Diagrama de Estados Ficha de Avaliação Diagnóstico e Plano de Intervenção...	60
Figura 13 – Diagrama de Sequência “Criar Processo Individual”	61
Figura 14 - Diagrama de Sequência “Consultar Processo”	62
Figura 15 – Diagrama de Sequência “Criar Ficha de Avaliação Diagnóstico”	63
Figura 16 – Diagrama de Sequência “Criar Plano de Intervenção”	64
Figura 17 – Diagrama Entidade - Relação	65
Figura 18 – Ciclo de Vida de uma aplicação no <i>Mendix</i>	68

Índice de Tabelas

Tabela 1 – Critérios da DSR	15
Tabela 2 – Os princípios dos métodos ágeis	39
Tabela 3 – Casos de Uso do Sistema.....	46
Tabela 4 – Descrição de Caso de Uso	48
Tabela 5 - Descrição de Caso de Uso "Sinalizar a Pessoa em Situação de Sem-Abrigo"	49
Tabela 6 – Descrição do caso de uso “Enviar equipa de emergência”.	50
Tabela 7 – Descrição do caso de uso “Criar Processo Individual”.	51
Tabela 8 – Descrição de caso de uso “Criar Utilizador”.....	52
Tabela 9 - Descrição de caso de uso “Atribuir o caso a um Gestor de Caso”.....	53
Tabela 10 – Descrição do caso de uso “Consultar os Processos”.	54
Tabela 11 - Descrição do caso de uso “Editar Processo”.....	55
Tabela 12 - Descrição do caso de uso “Criar Ficha de Avaliação Diagnóstico”.	56
Tabela 13 - Descrição do caso de uso “Criar Plano de Intervenção”.....	57
Tabela 14 - Comparação entre as diferentes plataformas de <i>low-code</i>	67

Lista de Siglas/Acrónimos

DSR – *Design Science Research*

ENIPSA – Estratégia Nacional para a Integração de Pessoas Sem-Abrigo

ENIPSSA – Estratégia Nacional para a Integração de Pessoas em Situação de Sem-Abrigo

ETHOS – *European Typology of Homelessness and Housing Exclusion*

FEANTSA – Federação Europeia das Organizações Nacionais que Trabalham com Sem-Abrigo

GIMAE – Grupo de Implementação, Monitorização e Avaliação da Estratégia

IEC – *International Electrotechnical Commission*

IEEE – *Institute of Electrical and Electronics Engineers*

ISO – *International Organization for Standardization*

NPISA – Núcleo de Planeamento e Intervenção Sem-Abrigo

Pssa – Pessoa em situação de sem-abrigo

RCAAP – Repositórios Científicos de Acesso Aberto de Portugal

RUP – *Rational Unified Process*

SI – Sistema de Informação

TIC – Tecnologias de Informação e Comunicação

UML – *Unified Modeling Language*

1. Introdução

As Pessoas em situação de sem-abrigo (Pssa) são uma realidade com a qual nos deparamos diariamente, colocando desafios à nossa sociedade e exortando valores como a solidariedade e a preocupação para com o próximo, mas não são uma realidade nova no país. Desde o século XIII que existem disposições legais sobre a condição destas pessoas, as quais foram referidas ao longo da nossa história como indigentes, pedintes, vagabundos ou mendigos. Este modo de vida foi desde sempre encarado como errado e alvo de punições e repressões. “Em Portugal, a legislação sobre o fenómeno da vagabundagem e da mendicidade remonta ao reinado de D. Afonso II (1211), havendo posteriormente sucessivas medidas legislativas sobre o assunto. Mais tarde, as Ordenações Afonsinas (1446) e as Ordenações Manuelinas (1521) insistem nas medidas repressivas (Soeiro, 1960)” (Pimenta, 1992, p. 159). Com o fim do Estado-Novo terminou também a legislação que punia este tipo de comportamento, tendo vindo a adoptar-se desde aí uma postura diferente em relação a estas pessoas, olhando para o problema “essencialmente numa dimensão social e estrutural, com causalidades no desenvolvimento socioeconómico e cultural do país e já não numa perspectiva individualista e moralista (incapacidade para providenciar o sustento ou decorrente de uma atitude de preguiça e ociosidade)”, (Fernandes, 2006, p. 3).

Em Portugal as diretrizes para combater esta situação são baseadas nas diretivas europeias emanadas pela Federação Europeia das Organizações Nacionais que Trabalham com Sem-Abrigo (FEANTSA). A FEANTSA é uma rede de Organizações Não-Governamentais (ONG) que têm como objetivo último acabar com a situação de sem-abrigo na Europa, lutando para atenuar a pobreza e exclusão social destas pessoas ou de pessoas em risco de ficar nesta situação, assim como promover o envolvimento de diferentes instituições europeias nesta missão. Segundo a Estratégia Nacional para a Integração de Pessoas Sem-Abrigo 2009 – 2015 (ENIPSA), “Considera-se pessoa sem-abrigo aquela que, independentemente da sua nacionalidade, idade, sexo, condição sócio-económica e condição de saúde física e mental, se encontre sem tecto, vivendo no espaço público, alojada em abrigo de emergência ou com paradeiro em local precário, ou sem casa, encontrando-se em alojamento temporário destinado para o efeito”. Deixou-se então de olhar para estas pessoas como preguiçosas ou ociosas e passou-se a olhar para elas como vítimas de situações que as colocaram numa situação de sem-abrigo, havendo hoje em dia para as ajudar uma rede de instituições governamentais e não-governamentais, que todos os dias estão no terreno a ajudar estas pessoas.

O Projeto teve por objetivo analisar a situação de Pssa no que se refere ao Núcleo de Planeamento e Intervenção Sem-Abrigo (NPISA) do Barreiro e à Cáritas Diocesana de Setúbal, e é para esta última que este projeto será dirigido. Apesar deste projeto ser dirigido para a Cáritas de Setúbal, a informação essencial para conseguir realizar este projeto como as reuniões que permitiram entender como estas instituições trabalham, como estão organizadas e o que pretendem mudar na maneira como tratam a informação foram realizadas com responsáveis da NPISA Barreiro, assim como o material de apoio foi obtido junto desta instituição. É precisamente para ajudar estas instituições no seu dia-a-dia que este projeto nasceu, para que através de inovações tecnológicas no processo de recolha e tratamento de dados o seu trabalho possa ser mais eficiente, reservando mais tempo para o contacto com as pessoas e sendo necessário menos tempo em trabalho administrativo.

1.1. Motivação

Este projeto nasceu da vontade de analisar a situação de Pssa e das associações fazerem uma parceria que permitisse modernizar o Sistema de Informação (SI) existente. Após reuniões com responsáveis da NPISA Barreiro, no IPS e na sede da associação no Barreiro, foi explicado que atualmente o processo de recolha de informação por parte dos colaboradores da associação é feita em papel, a qual é posteriormente passada para formato informático (folha excel) quando os colaboradores na sede têm oportunidade para o fazer. Era pretendido pela associação criar um software que permitisse aceder a estes dados, inserir e consultar a informação, assim como realizar análises estatísticas da mesma. Foi proposto pela orientadora desenvolver este projeto, proposta considerada desafiante, uma vez que proporcionaria a experiência de projetar de raiz a informatização do sistema de informação da associação, acompanhando as diferentes fases desse processo. Outro ponto interessante foi a oportunidade de associar as Tecnologias de Informação e Comunicação (TIC) à área do serviço social, nomeadamente as organizações que trabalham com Pssa, algo que ainda não é muito comum, fazendo com que as TIC possam ter um impacto positivo nas suas vidas. Nos tempos atuais, em que associado à crise pandémica se teme uma deterioração das condições de vida das franjas mais vulneráveis da nossa sociedade, onde estas pessoas se incluem, a realização deste projeto revela-se pertinente. É, assim, motivo para se realizar este projecto criar uma base que permita o desenvolvimento futuro de uma aplicação que informatize os processos de recolha, armazenamento e tratamento de dados junto de Pssa, criar uma base de dados possível de aceder informaticamente, assim como desenvolver

uma ferramenta que interligue todas estas componentes e que possa ser utilizada pela instituição em contexto laboral.

1.2. Objetivos

O objetivo deste projeto é analisar a temática de Pssa e desenvolver um protótipo que sirva de base para o desenvolvimento futuro de uma solução que contenha uma base de dados com a informação recolhida junto de Pssa, assim como a versão móvel que permita aceder e inserir esta informação no terreno e actualizar a mesma online.

Essa solução, que irá permitir criar diferentes perfis de utilizador, terá como objetivos:

- Sinalizar o aparecimento de novas Pssa na zona;
- Enviar as equipas de emergência ao local da nova sinalização;
- Permitir aos gestores de caso e equipas de emergência inserir no sistema dados de Pssa no terreno;
- Consultar esses dados;
- Editar os dados.

O protótipo desenvolvido tem por objetivo otimizar o trabalho que se faz atualmente em papel e pretende primariamente fazer a informatização de todos esses processos que atualmente se fazem em papel, juntando também novas funcionalidades que passarão a ser possíveis com a informatização do processo. Neste relatório apresenta-se a interação realizada atualmente entre a Cáritas Setúbal e as Pssa, que tecnologias são atualmente utilizadas e quais se podem implementar com o objetivo de melhorar toda esta dinâmica, assim como propor o desenvolvimento de uma aplicação tecnológica que permita aos funcionários da Cáritas Setúbal inserir e aceder à informação necessária para desenvolverem o seu trabalho.

1.3. Estrutura

Este relatório está dividido em seis capítulos.

O primeiro capítulo, denominado **Introdução**, contém motivação, objetivos e a estrutura do projecto.

O **Enquadramento Teórico** é o capítulo dois, em que se apresentam em diferentes subcapítulos, os conceitos teóricos que irão servir de base a este projecto, assim como o estado da arte nesses conceitos.

No terceiro capítulo, a **Metodologia**, apresentam-se conceitos teóricos relativos à engenharia de requisitos, abordam-se diferentes metodologias de desenvolvimento de software.

A **Caracterização da Organização em estudo**, que é o quarto capítulo, apresenta o caso prático, a caracterização da organização, a metodologia usada no desenvolvimento do projecto, todo o estudo do caso e planeamento do futuro programa, desde a análise de requisitos, passando pela tabela, diagrama e descrição de casos de uso, diagramas de actividades, estados e sequências até à modelação de dados.

O quinto capítulo, a **Análise comparativa de plataformas**, veremos as plataformas que foram estudadas para desenvolver o software planeado.

Por último, no sexto capítulo, **Conclusões e perspectivas de trabalho futuro**, apresentar-se-ão as conclusões obtidas e as perspectivas que existem para se desenvolver, bem como as perspectivas de trabalho futuro.

2. Enquadramento Teórico

Neste capítulo procede-se ao desenvolvimento do enquadramento teórico de alguns conceitos que são fundamentais. Será abordada a temática de Pssa, o enquadramento legal das pessoas nesta situação, a Estratégia Nacional para a Integração das Pessoas em Situação de Sem-Abrigo (ENIPSSA), a Cáritas de Setúbal, os sistemas de informação, a segurança de informação, as tecnologias de informação e a recolha de informação.

2.1. Pessoa em situação de sem-abrigo

- 1. Todos têm direito, para si e para a sua família, a uma habitação de dimensão adequada, em condições de higiene e conforto e que preserve a intimidade pessoal e a privacidade familiar.*

Artigo 65º da Constituição da República Portuguesa (Diário da República n.º 86/1976, Série I de 1976-04-10)

Vemo-los nas ruas das nossas cidades, muitas vezes com a indiferença de quem vê algo natural e que já faz parte da paisagem urbana, no entanto esta realidade não deve ser encarada como algo normal, mas sim como um problema complexo, que em cada caso existe por diferentes razões e fatores. Como escreve Anderson (2003), é um problema que deve ser olhado e pensado multidisciplinarymente, pois numa era em que a interação entre diferentes áreas e a cooperação no trabalho são cada vez mais normais, o contributo de diferentes áreas de estudo sobre este problema pode ser essencial para que se encontrem soluções adequadas.

Em Portugal este é um problema relativamente recente, não por nunca ter existido, mas sim porque nunca foi olhado como um problema social, mas sim como uma opção dos indivíduos que vivem desta maneira. Como refere Pimenta (1992) existem leis desde o século XIII sobre este fenómeno, na altura referido como vagabundagem ou mendicidade, legislação essa de carácter punitivo para os seus praticantes, tendo já no século XX durante o período do Estado Novo esta temática sido alvo de abundante legislação, sempre de carácter repressivo/punitivo. Apenas após os 25 de Abril de 1974 o paradigma mudou, passando a ser oficialmente encarado como um problema mais social que individual, sendo o Decreto-Lei nº 365 de 1976 o “assumir” pelo estado de que deve desempenhar um papel ativo para ajudar estas pessoas.

Apesar do termo “Sem-Abrigo” indicar uma característica destas pessoas (a ausência de habitação) não se deve olhar para este problema como apenas o problema da falta de abrigo,

que uma vez resolvido, a situação das mesmas mudaria e integrar-se-iam na sociedade normalmente. O problema é muito mais profundo e complexo que isso, acontecendo em algumas situações as pessoas nesta situação terem casa, mas abandonarem-na após um processo de ruturas relacionais. Para Alfredo Bruto da Costa (1998), a pobreza e a exclusão são apontadas como algumas das causas da situação de sem-abrigo, ao gerarem percursos de vida que podem culminar numa situação de sem-abrigo. Pese embora esta abordagem de Alfredo Bruto da Costa (1998) sugerir que não é possível definir um padrão pré-definido destes indivíduos, num estudo realizado em 1997 para a Federação Europeia das Organizações Nacionais que Trabalham com os Sem-Abrigo (FEANTSA), ele reconheceu que existem traços comuns entre esta população, como serem maioritariamente homens, jovens adultos, com baixas habilitações escolares e solteiros. A maioria era desempregada, tendo uma percentagem elevada de doenças mentais, alcoolismo e toxicodependência. Pode-se assim ter uma ideia tanto da complexidade do problema, como da população alvo que se irá tentar ajudar com este projeto.

2.2. Enquadramento Legal

A caracterização de Pssa inclui a análise do enquadramento em diversos domínios, nomeadamente do que se refere à estratégia nacional e à Federação Europeia das Organizações Nacionais. Em Portugal, as estratégias existentes para integrar as Pssa foram aprovadas em Resolução do Conselho de Ministros. A Resolução do Conselho de Ministros n.º 2/2020, (RCM, 2020), altera a Estratégia Nacional para a Integração das Pessoas em Situação de Sem-Abrigo 2017-2023 (ENIPSSA 2017-2023), estratégia essa que foi aprovada na Resolução do Conselho de Ministros n.º 107/2017, de 25 de julho. A implementação da ENIPSSA 2017-2023 realiza-se através de Planos de Ação bienais, aprovadas em sede da Resolução do Conselho de Ministros n.º 107/2017 de 25 de julho (RCM, 2017), que incluem os eixos, objetivos estratégicos e ações executadas através de atividades, metas, indicadores, orçamento (direto e indireto), calendário e entidades (responsáveis e parceiras), (ENIPSSA, 2017).

A assunção de competências ao nível da implementação da ENIPSA 2009-2015, a respetiva monitorização e avaliação de todo o processo, era, de acordo com a Resolução (RCM, 2020), responsabilidade de um grupo interministerial, coordenado pelo Instituto da Segurança Social, constituído por um conjunto de entidades públicas e privadas, denominado por Grupo de Implementação, Monitorização e Avaliação da Estratégia.

Na Europa, as Pssa são abordadas de forma diferente de acordo com o país. O *European Typology of Homelessness and Housing Exclusion* (ETHOS), que é um estudo que revê as definições de Pssa existentes assim como as realidades com que os prestadores de serviços lidam diariamente, foi desenvolvido pela FEANTSA. Estas categorias tentam, assim, cobrir todas as situações de vida que representem formas de sem-abrigo em toda a Europa:

- falta de teto (sem abrigo de qualquer tipo);
- falta de casa (com um lugar para dormir, mas temporário em instituições ou abrigo);
- viver em habitações inseguras (ameaças devido a arrendamentos precários, despejos, violência doméstica);
- viver em habitações inadequadas (em caravanas, em parques de campismo ilegais, em habitações impróprias, em sobrelotação extrema).

Atualmente, a situação dos jovens nesta temática tem merecido destaque por parte da FEANTSA. A investigação internacional e o conhecimento dos serviços de primeira linha (FEANTSA, 2020), indicam que os jovens sem-abrigo poderão ter um percurso diferente de Pssa “comuns” dadas as necessidades distintas. Esta situação prevê uma definição de sem-abrigo para os jovens no sentido de as organizações trabalharem para a erradicação dos sem-abrigo nesta faixa etária. Freek Spinnewijn, Diretor da FEANTSA, afirmou que "Provavelmente, estamos a testemunhar o crescimento mais rápido dos sem-abrigo na União Europeia desta geração. Nesse contexto, é extremamente difícil ser otimista em relação ao futuro. Mas existem algumas razões para ter esperança" (FEANTSA, 2019). Na Europa, (FEANTSA, 2020), temos de repensar a forma como percebemos e definimos os sem-abrigo e, em particular, os sem-abrigo jovens. Segundo Baptista & Knutagård (2019), existem alguns casos em que a linha entre os serviços de sem-abrigo e os serviços sociais não é clara, no que ao apoio e salvaguarda de adultos vulneráveis diz respeito. Em alguns casos, os serviços de sem-abrigo centram-se em abrigos de emergência e, em graus variados, em habitações transitórias de local único. De acordo com (FEANTSA, 2017), um exemplo de uma resposta com resultados comprovados liderada pela habitação a Pssa é a Habitação Primeiro, que envolve o acesso rápido à habitação com muito poucas condições, a par de um pacote intensivo de apoio centrado na pessoa.

2.3. Estratégia Nacional para a Integração de Pessoas em situação de sem-abrigo

Para a organização, “A necessidade de uma resposta efetiva da sociedade aos casos de pessoas em situação de sem-abrigo ou em risco é justificada, em primeira instância, pela realidade concreta de uma forma extrema de pobreza e de exclusão social pela qual muitos indivíduos, ao serem afastados dos múltiplos contextos de participação cívica, deixam de poder fruir das oportunidades de realização em igualdade de direitos com os demais cidadãos.” (ENIPSSA, 2020). É para oferecer esta resposta que em 2007 o governo constituiu um grupo de trabalho interinstitucional/intersectorial com o objetivo de cumprir as diretrizes europeias nesta matéria, tendo isto tido como resultado a criação da Estratégia Nacional para a Integração das Pessoas Sem-Abrigo (ENIPSA): Prevenção, Intervenção e Acompanhamento 2009-2015.

Este plano de 2009-2015 teve como resultado um relatório apresentado em 2017, tendo como principais conclusões a estratégia ter contribuído positivamente para a reflexão desta problemática, ao colocar o foco no envolvimento de várias entidades públicas e privadas na conceção, implementação e monitorização.

Em 2017 é criada nova estratégia, sendo redefinida para ENIPSSA (Estratégia Nacional para a Integração das Pessoas em situação de sem-abrigo) para o período 2017-2023. Esta redefinição surgiu por se considerar que “a designação de “sem-abrigo” deveria ser alterada para “Pessoa em situação de sem-abrigo”, uma vez que não se deve assumir o “sem-abrigo” como condição de vida de uma pessoa, mas sim como uma situação que poderá caracterizar uma determinada fase na vida de uma pessoa e que se deseja ser de transição na vida do indivíduo.” (ENIPSSA, 2020). Esta estratégia é composta por três eixos de intervenção, que são:

- promoção do conhecimento do fenómeno de Pssa, informação, sensibilização e educação;
- reforço de uma intervenção promotora da integração de Pssa;
- coordenação, monitorização e avaliação da ENIPSSA 2017-2023.

A ENIPSSA é composta por uma Comissão Interministerial e uma Comissão Consultiva, o Grupo de Implementação, Monitorização e Avaliação da Estratégia (GIMAE) que integra um Núcleo Executivo, e a nível local os NPISA.

2.4. Sistemas de Informação

O conceito de SI é, como seguidamente apresentado face às tendências de alguns autores, em sentido lato, existindo, no entanto, visões comuns a todos. Segundo Laudon & Laudon (2020), um SI é um sistema que permite a recolha e transformação de dados em informação, de fontes internas e externas e que armazena e transmite essa informação de forma apropriada aos gestores, possibilitando as tomadas de decisão atempadas para planeamento, coordenação e controlo das atividades pelas quais são responsáveis. Para Bourgeois et al., (2019), os SI podem ser vistos como tendo cinco principais componentes: hardware, software, dados, pessoas e processos. É ao perceber como estes cinco componentes podem interagir e criar valor a uma organização que se pode compreender os SI. Para Rainer Jr. et al. (2016), um SI recolhe, processa, armazena, analisa e dissemina informações com um propósito específico, abrange entradas (dados) e saídas (relatórios, cálculos), processa essas entradas e gera saídas que são enviadas para o usuário ou outros sistemas. Segundo Basden (2017), os SI consistem em cinco áreas de pesquisa e prática: O uso humano dos computadores, a natureza dos computadores, o desenvolvimento dos SI, os componentes das TIC e as TIC como ecologia social. Este autor defende que se deve considerar estas cinco áreas em conjunto pois isso vai proporcionar uma abordagem mais holística, uma vez que estas cinco áreas não são todas tidas em conta em conjunto quando se analisa as SI. Por último, há que referir Albertão (2005), para quem um SI é composto por um conjunto de elementos e componentes relacionados entre eles, com uma determinada ordem, de maneira a recolherem, processarem e disseminarem os dados e informações, sendo essas informações usadas no processo de tomada de decisão.

Face ao exposto, considera-se que os SI têm implícitos conceitos chave que os constituem e que são referidos, explícita ou implicitamente pelos autores, como a recolha e processamento de dados e a sua transformação em informação, ou o conhecimento gerado por essa informação, que a torna numa valiosa ajuda na tomada de decisão. Podemos ver em comum a todos os conceitos apresentados que um SI envolve a entrada dos dados, o seu processamento e o output destes (informação). Estas características dos SI são fundamentais para este trabalho, que tem como *core* a recolha, processamento, armazenamento e análise da informação recolhida sobre as Pssa. A transformação dos dados recolhidos em informação relevante através destes processos dos SI é um dos grandes objetivos do trabalho. Os dados recolhidos junto de Pssa são importantes para que os objetivos da organização sejam atingidos, já que são eles que vão

permitir aos funcionários da Cáritas traçar um perfil de Pssa intervencionada, e consequentemente, definir um plano de acompanhamento mais eficaz.

2.5. Segurança de Informação

No ponto anterior foi apresentado o conceito de Informação e como pode ser valiosa para uma organização, assim a segurança de informação é também uma preocupação face à temática em estudo. Segundo Whitman & Mattord (2017), a preocupação com a segurança de informação começou com a segurança dos computadores, que despertou consciências e se desenvolveu por altura da Segunda Guerra Mundial. O desenvolvimento das primeiras máquinas capazes de quebrar os códigos de comunicação inimigas levou a que diversas camadas de segurança fossem construídas para manter a integridade dos dados. Com o final da Segunda Guerra Mundial e a chegada dos anos da Guerra Fria, as máquinas continuaram a evoluir para realizar tarefas mais complexas, o que fez surgir a necessidade de continuarem a comunicar de maneira menos pesada e mais fácil. Esta necessidade de comunicação militar levou à criação da ARPANET, a predecessora da Internet. Desde essa altura até aos dias de hoje as máquinas continuaram a evoluir para ficarem cada vez mais complexas e a necessidade humana de troca de informações continuou sempre a evoluir, de maneira que hoje em dia a informação valiosa circula a uma velocidade e em quantidades até há poucos anos inimagináveis. Empresas conseguem ter acesso a muitas informações dos seus clientes não só através da sua pegada digital, mas também muitas vezes com informação disponibilizada diretamente pelos mesmos.

Para as empresas dos dias de hoje é necessário ter a segurança e o controle como prioridades (Laudon & Laudon, 2020). Para este autor, como os SI em diferentes localizações podem ser conectados por redes de telecomunicações, o potencial de riscos como acesso não autorizado, uso indevido ou fraude aumenta, pois deixa de estar circunscrito a apenas um lugar e passa a ser possível a partir de qualquer ponto de acesso. Para melhor percebermos o que é a segurança no contexto da informação, podemos ter em conta a visão de Whitman & Mattord (2017), para quem Segurança é a proteção contra adversários, que podem causar mal à organização, intencionalmente ou não. Para estes autores, tem tudo que ver com riscos: que medidas preventivas utilizamos para evitar que aconteçam, e como minimizamos/mitigamos os seus efeitos se acontecerem. Esses riscos podem ter diferentes naturezas, como atos não-intencionais, desastres naturais, falhas técnicas, falhas de gestão e atos deliberados. De todos estes elementos podem vir ameaças que podem comprometer a segurança da informação.

Assim, para que exista Segurança de Informação, há três aspetos que têm de ser respeitados segundo a ISO/IEC 27002: 2013 – *Information technology — Security techniques — Code of practice for information security controls*: confidencialidade (garantir que informação só é consultada por quem tem autorização para o fazer), integridade (garantir que a informação é exata e credível) e disponibilidade (garantir que a informação e os ativos associados são acedidos por quem tem autorização, quando necessário). Esta tríade de aspetos que é imperativo respeitar também é referida por Bourgeois et al. (2019), referindo que a confidencialidade é o princípio de que a informação apenas deve ser acedida por quem necessita de o fazer, a integridade é o princípio de que a informação acedida não foi alterada e representa o que foi pretendido aquando da sua inserção e o princípio da disponibilidade, que significa que a informação pode ser acedida e modificada num prazo adequado por qualquer pessoa devidamente autorizada. Para que a confidencialidade e a disponibilidade sejam asseguradas, é essencial que exista um sistema que controle a gestão de identidades e acessos. Para Schwartz & Machulak (2018), a gestão de identidades engloba a gestão de perfil e password, assim como a sincronização que tem de existir quando a informação de alguém é criada ou atualizada, ao fazer com que sejam desencadeados processos no sistema quando isto acontece. Já a gestão de identidades e acessos engloba a autenticação do utilizador e os diferentes sistemas que permitem que o mesmo se ligue à rede. Segundo Madureira (2010), a gestão de identidades diz respeito ao ciclo de vida de um utilizador na organização, sendo que todas as informações de identificação relativas ao tempo de em que o utilizador esteve na organização fazem parte desta gestão, enquanto a gestão de acessos diz respeito ao controlo de acesso dos utilizadores após se autenticarem no sistema, definindo a informação a que o utilizador pode aceder enquanto está no sistema da organização. Apesar destes dois conceitos serem diferentes, só fazem sentido quando utilizados em conjunto. Para Stallings & Brown (2018), um sistema de controlo de acesso permite mediar a interação entre o utilizador e os recursos de um sistema. Este implementa políticas de segurança que definem quem acede, como acede, quando acede e ao que acede. Segundo este autor, existem três funções principais num sistema de controlo de acessos: autenticação (a verificação da validade das credenciais de quem está a tentar aceder ao sistema), autorização (verificação dos acessos aos recursos do sistema disponíveis para esse utilizador) e auditoria (Monitorização e armazenamento de toda a atividade de um sistema, de maneira a verificar se as suas políticas de acesso estão a ser respeitadas e se houve falhas de segurança) e no sistema de controlo de acessos existem três conceitos base, que são o sujeito (que é a entidade capaz de aceder a objetos, seja ele um utilizador, um processo ou as máquinas),

o objeto (um recurso do sistema com acesso controlado) e os direitos de acesso (que definem a forma como o sujeito vai interagir com o objeto).

Como é referido no Regulamento (UE) 2016/679 do Parlamento Europeu e do Conselho, que é o novo Regulamento Geral sobre a Proteção de Dados (RGPD), a rápida evolução tecnológica e a globalização, aliados ao aumento do fluxo transfronteiriço de dados pessoais, criaram novos desafios em matéria de proteção de dados. Para lidar com isto a UE aprovou a 27 de abril de 2016 o novo RGPD, o qual veio atualizar o original que data de 1995, e com isso responder a muitos dos novos desafios nesta temática que surgiram desde essa data e para os quais o regulamento em vigor estava desatualizado. A segurança de informação é relevante para este trabalho, dado que irão ser recolhidos e armazenados dados pessoais de Pssa, será imperativo que a segurança dos mesmos seja garantida. Nesse sentido considera-se premente criar um sistema de gestão de identidades e acesso que satisfaça as necessidades da organização, respeitando não só os princípios e conceitos falados neste subcapítulo, como o RGPD.

2.6. Tecnologias de Informação e Comunicação

As TIC potenciam delinear estratégias no sentido de dotar os trabalhadores da organização de ferramentas tecnológicas que lhes permitam recolher e aceder à informação sobre as pessoas intervencionadas. Para isso serão necessárias TIC como tablets ou smartphones com ligação à internet. As visões sobre o que se considera TIC variam conforme os autores. Por exemplo, para Laudon & Laudon (2020) por tecnologia de informação entende-se todo o software e todo hardware que uma empresa necessita para atingir os seus objetivos organizacionais. Para Castells (2012) as tecnologias de informação são um conjunto de tecnologias que permitem difundir e reproduzir informação. Essas tecnologias estão em microeletrónica, computação (hardware e software), telecomunicação, radiodifusão e optoelectrónica. Segundo o mesmo autor, houve grandes avanços nestas tecnologias nas últimas duas décadas do século XX.

Segundo Shun-Ping Chen (2020), desde o desenvolvimento dos cabos de comunicações e das tecnologias sem fio no século XIX, como o telefone ou o telégrafo, as TIC conheceram avanços tremendos no século XX, surgindo uma grande variedade de invenções nesta área, que continuam a ser desenvolvidas e melhoradas. O objetivo destas TIC é criar um canal de ligação de alta qualidade e fiável, que transmita os dados de informação multimédia entre a fonte de informação e o destino. Uma das invenções nesta área foram os computadores. Quando os computadores surgiram eram vistos como máquinas para fazer cálculos, armazenar informação

e automatizar processos de negócio, mas com o evoluir da tecnologia, foi ficando obvio que muitas funções de comunicação podiam ser integradas nos mesmos, o que levou a que muitas empresas, ainda nos anos oitenta, combinassem os seus departamentos de telecomunicações e sistemas de informação em apenas um, criando assim departamentos de Tecnologias de Informação e Comunicação (Bourgeois et al., 2019). Ainda segundo este autor, esta capacidade para os computadores comunicarem e facilitarem a comunicação entre os seus utilizadores, que tem na criação da internet o seu exemplo mais famoso, tem tido um impacto tremendo na digitalização do mundo atual.

Vivemos num mundo cada vez mais digital, em que o processo atual de transformação tecnológica expande-se exponencialmente em razão de sua capacidade de criar uma interface entre campos tecnológicos mediante uma linguagem digital comum na qual a informação é gerada, armazenada, recuperada, processada e transmitida (Castells, 2012). Portanto, a tecnologia de informação está hoje constantemente presente no nosso quotidiano, interligando-se com diferentes campos e tecnologias sempre com o objetivo de obter informação.

Já Lojkine (1995) aborda estas transformações nas tecnologias de informação e no mundo como uma “revolução informacional”, comparando-a à revolução industrial, explicando que a revolução industrial foi marcada pela mecanização de funções que até então eram feitas manualmente, e na revolução tecnológica, procura-se a automatização de funções intelectuais. Isto toca novamente na ideia de que vivemos num mundo digital, mas para este autor também é fundamental notar que o produto da indústria também está a mudar, deixando de ser apenas algo físico produzido numa fábrica para se passar a olhar para algo intangível como a informação e o conhecimento como mercadoria. Geralmente, a Pssa poderá ser encarada como um “excluído digital” – alguém que, segundo Castells (2012), ou não tem acesso à rede de comunicações, ou tem acesso, mas a capacidade técnica de a usar é muito baixa, ou tem acesso e até tem capacidades técnicas para a usar, mas não tem as capacidades cognitivas para discernir que informação é relevante e como usar as redes de comunicação de maneira a que tenham um efeito positivo na sua vida. Por isso, o técnico de ação social que trabalhe com as Pssa tem também a responsabilidade de ajudar estas pessoas a poderem usar estas tecnologias para melhorar a sua vida. Assim, as TIC abordam duas perspetivas: por um lado notando que houve uma transformação na indústria que fez com que a informação e o conhecimento que são gerados por um mundo cada vez mais interligado através das TIC passassem a ser um ativo comercializável. Por outro, definindo o que são as TIC e como evoluíram.

2.7. Recolha de Informação

A recolha de informação é algo que, hoje em dia, acontece a todo o momento, nas mais variadas situações do dia-a-dia. Basta passar um dia com um smartphone conectado à internet, e sem se dar por isso, já foram recolhidos inúmeros dados do utilizador. Por exemplo, dados de posicionamento, recolhidos através de GPS que permitem perceber por onde a pessoa andou, dados das pesquisas realizadas num motor de busca ou através de cookies que rastreiam os sites que a pessoa visitou e quanto tempo passou neles, dados esses que quando tratados podem gerar informação muito valiosa para as empresas, pois todos combinados permitem construir um perfil do consumidor, saber o que lhe interessa, o que procura, e direcionar muito mais eficazmente as suas ofertas para os consumidores mais propensos a quererem os seus produtos. A informação é, portanto, o “sumo” que se extrai dos dados recolhidos depois de tratados. Segundo Moresi (2003), a técnica de recolha de dados é o conjunto de processos e instrumentos que permitem registar as informações, o controlo e a análise de dados.

Para se recolher os dados, podem-se adotar métodos de investigação qualitativos ou quantitativos. Os primeiros, assentam no paradigma naturalista, segundo o qual “a realidade é múltipla e surge progressivamente através de um processo dinâmico que consiste em comunicar com os indivíduos num determinado meio” (Fortin, 2009, p. 41). Também segundo a autora, os métodos quantitativos, por sua vez, assentam no paradigma positivista, para o qual a realidade é só uma e estática. Para Fortin (2009), o paradigma naturalista é baseado numa forma holística de ver o ser humano, carregando algumas crenças. A realidade vai depender das perceções de cada um e variar com o tempo, ou seja, a experiência de vida e o ambiente envolvente de cada pessoa vão definir como ela percebe a realidade. Já o paradigma positivista tem a sua origem nas ciências físicas e para ele a verdade é objetiva e os factos não são influenciados pelo contexto ambiental ou histórico-social, sendo que o ser humano é formado por partes que podem ser medidas e quantificadas. O objetivo da investigação é estabelecer factos e determinar as relações entre eles. Dentro destes métodos de investigação, existem ligadas a cada um deles técnicas e instrumentos com os quais se irá recolher os dados, que serão tratados para se transformarem em informação. As técnicas que poderão ser usadas na recolha de dados são (Mertens, 2019): Observação, Análise Documental, Portfólios, *Focus Group*, Inquérito; *Testing*; *Delphi Method*; *Photovoice*.

3. Metodologia

No que concerne à metodologia adotada, considera-se que a *Design Science Research* (DSR) é adequada para realizar o projeto face à especificidade do tema. A DSR envolve a criação de um artefacto e/ou teoria e design como meio para melhorar o estado corrente das práticas utilizadas, mas também o conhecimento de desenvolvimento existente (Baskerville et al., 2018). É uma metodologia que, devido ao seu pragmatismo, é adequada à investigação de problemas de natureza prática, em vez de, por outro lado, focar-se na verificação de leis naturais ou teorias comportamentais (Hevner et al., 2004). Por conter as características descritas anteriormente e em função do carácter prático do projeto, decidiu-se usar a metodologia DSR para abordar a problemática em questão. Através da construção e aplicação do artefacto para um determinado fim, que será constituído por um número previamente definido de fases, conseguimos compreender e solucionar um problema.

Estes autores (Hevner et al., 2004), também definem os critérios que deverão ser tidos em conta aquando da realização da pesquisa. São eles os seguintes:

Tabela 1 – Critérios da DSR

Critérios do Método DSR	Resumo
Pragmatismo	Tentar melhorar a prática e a teoria. A teoria é valorizada na mesma medida em que influencia positivamente a prática
Avaliação do Design	A avaliação do artefacto deve ser rigorosa, de maneira a provar a sua utilidade, qualidade e eficácia
Contextualização	Documentar todo o processo, desde a pesquisa, aos resultados de investigação, até às alterações ao que estava inicialmente planeado.
Flexibilidade e Interatividade	Quem realiza a pesquisa deve estar envolvido em todos os processos do projeto. As pesquisas podem ser feitas de maneira flexível, podendo-se recorrer a várias técnicas para a recolha e análise dos dados.
Design como Processo de Pesquisa	Devem ser usados os meios ao nosso dispor para se encontrar um artefacto que seja indicado, desde que estes meios satisfaçam as leis existentes no ambiente do problema.

Rigor da Pesquisa	A pesquisa deve aplicar métodos rigorosos para construir e avaliar o artefacto.
Relevância do Problema	O problema deve ser resolvido através do desenvolvimento de soluções tecnológicas que possam ser relevantes e/ou importantes
Contribuição do Design	Este método deve ter como um dos objetivos contribuir claramente nas áreas específicas dos artefactos desenvolvidos, nas fundamentações e nas metodologias de design.
Comunicação da Pesquisa	A comunicação da pesquisa deve ser feita de maneira a ser entendida tanto pelo público com mais background na área das tecnologias como para o público da área da gestão.

Fonte: Adaptado de Hevner et al., (2004)

Segundo Hevner, et al. (2004), a DSR permite elencar as necessidades da organização e definidos os problemas que interessam ao investigador, suportar o desenvolvimento dos referidos artefactos. Para se avaliar se esses artefactos estão realmente a cumprir a sua função, estão sujeitos a avaliações que justifiquem a sua necessidade, e para sustentar estas atividades de desenvolvimento, construção e avaliação do artefacto, a base de conhecimento atual necessita de ser consultada e se possível atualizada. Essa base de conhecimento, ou estado da arte, são os fundamentos e métodos reconhecidos pelos diferentes autores de versões da teoria, reconhecidos em âmbito académico, e sustentam especialmente as fases de justificação e avaliação dos artefactos desenvolvidos.

Uma das abordagens ao método DSR realizada por Vaishnavi et al. (2019), em que o autor advoga que é indicada para desenvolver projetos na área dos SI. Esta abordagem é composta pelas seguintes fases:

1. **Consciencialização do problema** – Nesta etapa deve-se identificar e compreender a problemática que queremos estudar e que nos propomos a resolver. Deve-se começar por analisar conceitos, teorias e trabalhos relevantes que sejam úteis para melhor identificarmos e entendermos o problema e a sua possível solução. Para fazer isto, deveremos efetuar extensa pesquisa e revisão de literatura que nos sirva de suporte teórico ao desenvolvimento do projecto. Pesquisas sobre SI, Tecnologias de Informação, Segurança de Informação, estudos sobre a realidade nacional e internacional de Pssa e a

estrutura governamental que tem a responsabilidade de oferecer respostas a esse problema foram alguns dos temas pesquisados no Google Academic, internet, RCAAP, material fornecido pela NPISA, artigos científicos e livros;

2. **Sugestão** – Uma vez identificado o problema, deve-se sugerir soluções para essa problemática. Deve-se construir artefactos de modo a apresentar soluções criativas que possam ser usadas, abordagens novas a elementos já existentes ou a criação de novas teorias e modelos. Nesta etapa já teremos uma grande quantidade de informação recolhida da etapa anterior, pelo que se deve avaliar a viabilidade do projeto pensado e preparar a fase seguinte, de desenvolvimento do mesmo. Esta será a fase em que serão propostos à NPISA os objetivos do projeto e se chegará a um consenso em relação à viabilidade ou não das propostas apresentadas. Isso será feito com recurso a reuniões e análise de documentos.

3. **Desenvolvimento** – Devem-se analisar os artefactos que foram propostos no ponto anterior para solucionar o problema e desenvolvê-los. Estes artefactos poderão ser, segundo a teoria de March & Smith (1995), constructos (são aplicados para descrever os problemas e para especificar as respetivas soluções e também utilizados para descrever e ponderar as tarefas), modelos (quando queremos retratar/representar/descrever uma realidade, uma vez que tanto apresentam as variáveis de determinado sistema, como também suas relações, devendo sempre tentar captar a realidade detalhadamente), métodos (são algoritmos usados para a realização de tarefas. Tanto podem ser representados graficamente ou em algoritmos específicos e têm como principal vantagem favorecer a transformação dos sistemas, promovendo assim a sua melhoria) ou instâncias (podem ser regras que servem para orientar a utilização de um artefacto num ambiente real. Informam como se deverá implementar os artefactos desenvolvidos e que resultados se pode esperar deles), podendo também por vezes ser uma combinação destes. Se os desenvolvimentos aparentarem ser adequados, irão ser posteriormente avaliados, se ao invés o artefacto não se mostrar adequado, poderemos voltar à etapa de consciencialização do problema, de maneira a compreendê-lo melhor e construir um artefacto mais indicado (Lacerda et al., 2013). Este desenvolvimento será a parte da programação, em que se desenvolverá a plataforma de maneira a ir de encontro às necessidades da organização.

4. **Avaliação** – Nesta etapa é verificado o artefacto criado, no ambiente para o qual foi pensado e avalia-se os resultados dessa verificação. Quem lida com a investigação, seja quem a realiza como quem a ela acede, pode adquirir conhecimentos e novas aprendizagens. Esta avaliação será realizada ao longo de todo o processo de desenvolvimento e recorrendo a metodologia existente, definida por Hevner et al. (2004), sendo esses métodos a observação, o método analítico, o experimental, testes e argumentação.
5. **Conclusão** – Aqui deverá-se apresentar os principais resultados do estudo realizado, o que foi feito, a sua pertinência, a inovação e concluir sobre o sucesso ou insucesso do projeto desenvolvido.

3.1. A recolha de informação de Pessoa em situação de sem-abrigo

A recolha de informação de Pssa é um dos pilares de todo este processo. Tudo começa com a sinalização de Pssa. Quando uma entidade (pode ser um cidadão ou uma instituição) identificam em determinado sítio da cidade alguém que aparenta estar em situação de sem-abrigo. Esta informação é comunicada ao coordenador da entidade local da área em questão, que comunica com as equipas de intervenção de rua, que vão averiguar a situação em questão: se a pessoa é nova nessa situação ou já era alguém sinalizado, se for alguém novo nessa situação, recolher informações básicas através de um questionário com a função de identificar/caracterizar os indivíduos, sendo assim feita a primeira recolha de informação sobre a Pssa.

Após reuniões com funcionários do NPISA Barreiro e avaliação de documentação facultada pelos mesmos, foi possível perceber que estes no seu trabalho de recolha de informação usam tanto o método quantitativo como o qualitativo. As técnicas mais usadas por estes profissionais para recolherem informação são o inquérito e a observação, e entre os instrumentos aplicados para recolher informação encontram-se o questionário, a entrevista, o diário do investigador, as análises estatísticas e de conteúdo e as *checklists*. Tendo por base esta documentação, verifica-se que nesta ficha devem constar a seguinte informação:

- Nome;
- Data de nascimento;
- Morada;
- Naturalidade e Nacionalidade;

- Género;
- Documentos de Identificação (Cartão de Cidadão (CC), Passaporte, Numero de Identificação da Segurança Social (NISS), Numero de Identificação Fiscal (NIF), Cartão de Utente do Serviço Nacional de Saúde);
- Estado Civil;
- Tipo de agregado;
- Alojamento atual;
- Razões para estar nesta situação;
- Alojamento anterior;
- Habilitações literárias;
- Emprego/Ocupação;
- Fonte de Rendimento;
- Contactos sociais da pessoa;
- Dados de Saúde;
- Identificações complementares;
- Identificação de Apoios (Apoios sociais que já tenha, e quem lhos presta).

Depois de recolhidos e validados os dados, o processo é atribuído pelo coordenador a um gestor de caso. Este tem como função contactar com a Pssa para a qual foi designado, tendo desde logo acesso à ficha de identificação preenchida anteriormente, devendo, no entanto, validar junto da pessoa intervencionada a informação aí veiculada, tal como preencher alguns campos que possam ter ficado por preencher. Deve também preencher uma ficha de avaliação diagnóstico, em formato de papel, que tem por objetivo aprofundar as informações recolhidas anteriormente, registar as condicionantes e necessidades prioritárias da pessoa (económicas, alojamento, segurança, situação familiar, saúde, emprego, justiça, documentação ou outra problemática não referida), identificar os recursos da pessoa (o que sabe fazer, que habilitações e competências tem), as suas expectativas, que processos ou intervenções sociais de outras entidades estão em curso e registar outras informações complementares que possam interessar. A partir deste momento, pode-se delinear um Plano de Intervenção. O Plano de Intervenção tem como objetivo definir um plano de ação, em conjunto com a pessoa intervencionada, em que se definem objetivos/metapas para esta pessoa atingir, que podem ser de todo o tipo: passar a ter cuidado com a aparência, melhorar a higiene, ir ao médico, arranjar um emprego ou melhorar as apetências sociais, por exemplo. Nesse plano também devem constar

compromissos, que tanto podem ser ações que a pessoa se propõe a realizar como podem ser do gestor de caso, um campo para fazer o acompanhamento da situação, que permita ao gestor de caso ir anotando o desenvolvimento da pessoa e por fim uma secção onde se irá colocar as respostas que o gestor de caso acha que devem ser criadas a nível superior para atender às necessidades de Pssa.

Após toda esta recolha de informação, é necessário ao gestor passar esta informação para formato digital. Isto significa que o gestor de caso, ao chegar à organização onde trabalha, terá de passar todas estas informações para um ficheiro que ficará gravado numa pasta no computador. Atualmente a informação é assim armazenada, num ficheiro Excel para cada caso, guardadas numa pasta no computador da organização. Este processo gera uma duplicação de trabalho e conseqüente perda de tempo desnecessária para o gestor de caso, para além de uma clara falta de segurança no armazenamento dos dados, que pode comprometer a sua integridade e não evita minimamente possíveis fugas de informação. É assim recorrendo a estas folhas de cálculo que as estatísticas sobre as informações recolhidas serão feitas, assim como será retirada a informação relevante que permitirá eventualmente melhorar o processo de tomada de decisão.

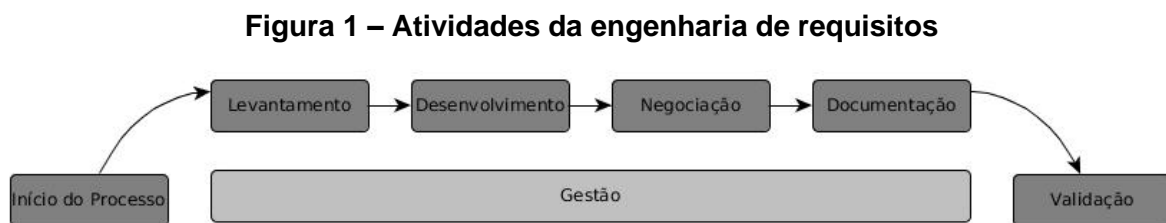
3.2 Engenharia de Requisitos

Para se construir um sistema de informação que satisfaça as necessidades dos utilizadores, é necessário avaliar que informação é necessário recolher e qual deve constar no sistema. Esse processo é conhecido como Análise ou Engenharia de Requisitos. Vários autores já se debruçaram sobre a definição de requisito e como se aplica nos sistemas de informação, sendo unânime o conceito de que os requisitos de um sistema existem em função do que os clientes necessitam. Segundo o *Institute of Electrical and Electronics Engineers (IEEE) Standard Glossary of Software Engineering Terminology* (IEEE, 1997), um requisito é uma condição ou capacidade necessária para um utilizador resolver um problema ou alcançar um objetivo.

Para Fernandes & Machado (2017), “um requisito é uma capacidade que um sistema deve possuir, para satisfazer necessidades de utilizadores”. Segundo Sommerville (2019), quando os requisitos de um sistema são definidos na fase inicial do seu desenvolvimento, eles representam as especificações que devem ser implementadas no sistema. Afirmam Caldas Jr. e Masiero (1999) que, para que o desenvolvimento do sistema vá ao encontro das necessidades dos utilizadores e clientes, é fundamental uma representação completa e correta, isto considerando

que a satisfação dos requisitos que foram especificados é uma pré-condição básica para o sucesso de qualquer sistema.

O engenheiro de requisitos deve definir e estabelecer o conjunto de informações que lhe permitem identificar o problema, sendo por isso sua tarefa entender o problema, numa linguagem consonante com a dos utilizadores, e definir um sistema que satisfaça as suas necessidades (Leffingwell e Widrig, 2000). Segundo Sommerville (2019), o que se pretende por intermédio dos requisitos é expor o sistema, explicar o seu comportamento, as suas funções e restrições. Podem-se considerar diversos tipos de requisitos, entre os quais: os requisitos gerais, que descrevem de uma forma geral as propriedades do sistema; os requisitos funcionais, que definem o que o sistema pode ou não fazer; os requisitos de implementação, que descrevem de que maneira o sistema deve ser implementado; os requisitos de performance, que especificam as performances mínimas que poderão ser aceites no sistema e os requisitos de acessibilidade, que indicam as normas que definem quem e como se pode aceder ao sistema. Para Pressman & Maxim (2019) o processo de engenharia de requisitos deve ter as seguintes atividades: (1) início do processo, (2) levantamento de requisitos, (3) desenvolvimento de requisitos, (4) negociação de requisitos, (5) documentação de requisitos, (6) validação de requisitos e (7) gestão de requisitos.



Fonte: Adaptado de Fernandes & Machado (2017)

Apesar da Ilustração acima mostrada ser sequencial, é importante referir que esta deve ser vista de maneira flexível pois ocorrem frequentemente alterações à sequência de acontecimentos representada. A gestão de requisitos é uma atividade paralela a todas as outras, uma vez que desde o momento em que termina a primeira etapa do processo, a gestão de requisitos interage com todas as outras atividades (Fernandes & Machado, 2017). As fases englobam o início do processo, levantamento de requisitos, desenvolvimento de requisitos, negociação de requisitos, documentação de requisitos, validação de requisitos e gestão de requisitos.

3.2.1 Início do processo

Segundo Fernandes & Machado (2017), o processo de engenharia de requisitos inicia-se através de uma necessidade ou uma expectativa de negócio, portanto nesta atividade é obrigatório identificar o objetivo do projeto. Também segundo estes autores é realizado um estudo de viabilidade, com o objetivo de analisar o sistema como um todo.

De modo a facilitar esta atividade, sugerem-se três grupos de questões:

1. Questões relativas ao objetivo do projeto: Os autores Pressman & Maxim (2019) recomenda as seguintes questões para esta fase:
 - Quem faz o pedido ao sistema?
 - Quem irá usar o sistema?
 - Quais são os benefícios económicos do sistema?
 - Há outras fontes que é possível usar/consultar?
2. Questões para perceber o problema do cliente. Questões sugeridas (Pressman & Maxim, 2019):
 - Que características deseja ver incluídas nos outputs que o sistema vai produzir?
 - Que problemas resolve a solução?
 - Conseguir descrever o ambiente organizacional onde o sistema irá trabalhar?
 - Será o sistema afetado por alguma questão ou restrição de desempenho?
3. Questões para perceber a eficácia da comunicação entre intervenientes: Este tipo de questões ajuda os intervenientes a perceberem se a comunicação entre eles está a ser bem-sucedida. Questões sugeridas (Pressman & Maxim, 2019):
 - Considera-se a pessoa correta para responder às minhas questões?
 - Acha que as minhas questões são pertinentes para o seu problema?
 - Estou a colocar um número excessivo de questões?
 - Há questões que eu deveria estar a colocar e não estou?
 - Há alguém que possa providenciar informações adicionais?

3.2.2. Levantamento de Requisitos

O levantamento de requisitos é a atividade onde são recolhidos os requisitos das fontes através de uma técnica de levantamento (Fernandes & Machado, 2017). Segundo os autores Christel e Kang (1992), a primeira tarefa do levantamento de requisitos passa por identificar os parceiros que são fontes de requisitos. Estes podem ser (IEEE, 2014):

1. **Objetivos:** Podem ser chamados de interesse de negócio ou fator crítico de sucesso. Refere-se aos objetivos de alto nível de um sistema de informação;

2. **Conhecimento do domínio:** O conhecimento do domínio dá ao engenheiro de requisitos de software o conhecimento de que necessita para debater com *stakeholders* sobre o modo como irá operar o sistema de informação projetado, o que faz com que sejam uma ajuda importante nas entrevistas de levantamento de requisitos.

3. **Stakeholders:** São pessoas que influenciam ou são influenciados por um projeto. O problema que origina o projeto por vezes é identificado nesta fonte (futuros utilizadores), contudo há outros tipos de *stakeholders* que não têm um impacto tão direto no projeto, mas se forem ignorados, podem influenciar de maneira negativa o sucesso da solução. Fernandes & Machado (2017), identificam os seguintes tipos de *stakeholders*:

- **Utilizador:** Pessoa que lida diretamente com o sistema;
- **Cliente:** Alguém que paga para adquirir um sistema;
- **Especialista:** Alguém que tem um conhecimento profundo de determinado domínio ou assunto;
- **Desenvolvedor:** Pessoa que realiza atividades que contribuem para o desenvolvimento e manutenção de um dado sistema técnico;
- **Inspetor:** Indivíduo que supervisiona o sistema;
- **Stakeholder negativo:** Alguém que deseja que o sistema não seja desenvolvido, nem colocado em operação.

4. **Regras de negócio:** Estados que definem ou restringem algum aspeto da estrutura ou comportamento do negócio;

5. **Ambiente operacional:** Requisitos derivam do ambiente em que o sistema de informação irá ser executado;

6. **Ambiente organizacional:** O engenheiro de requisitos de software deve ter sensibilidade para não fazer com que o sistema de informação force mudanças não planeadas em processos de negócio. Também deve ter em consideração que o sistema de informação será condicionado pelo ambiente da organização, ou seja, a sua estrutura, cultura e políticas internas.

De modo a levantar corretamente requisitos das fontes, existem técnicas que auxiliam a atividade. Segundo Fernandes e Machado (2017), as técnicas de levantamento de requisitos podem ser separadas em três categorias: individuais, grupos de pessoas e artefactos.

Técnicas de levantamento de requisitos individuais:

- Entrevista: Colocar questões a pessoas que irão lidar com o sistema;
- Pesquisa: Técnica que visa utilizar questionários para recolher e tratar informação obtida de múltiplos respondentes;
- Introspeção: O engenheiro de software coloca-se na pele do cliente ou utilizador final e define os requisitos;
- Etnografia: Estudo do comportamento de pessoas no seu ambiente natural.

Técnicas de levantamento de requisitos baseadas em grupos de pessoas:

- Brainstorming: Técnica que congrega um grupo de pessoas de modo a facilitar a troca de ideias;
- Foco de grupo: Convocatória de um grupo de pessoas para discutir um ponto específico;
- Aplicação conjunta da conceção: Tem o objetivo de juntar peritos do problema e engenheiros com o objetivo de se obter mais informação do domínio do problema;
- Trabalho cooperativo: trabalho que resulta de uma interação entre vários intervenientes. Para auxiliar esta técnica existe software que facilita o trabalho em grupo executado cooperativamente.

Técnicas de levantamento baseadas em artefactos:

- Análise do domínio: Analisar o domínio onde o sistema estará localizado de maneira a encontrar elementos comuns a todos os sistemas do domínio;
- Análise orientada a objetos: Estudar os requisitos na perspectiva de objetos do mundo real e das suas classes;
- Prototipagem: Técnica valiosa para clarificar requisitos ambíguos;
- Cenário: Pequena história que descreve o comportamento funcional de um sistema e ilustra uma sequência específica de ações e eventos necessários para a execução;
- Modelação de objetivos: Técnica que estrutura os objetivos hierarquicamente;
- Personagem: Técnica onde se pretende fazer uso de uma pessoa fictícia que representa um tipo importante de utilizadores do produto em desenvolvimento.

3.2.3. Desenvolvimento dos Requisitos

Nesta fase o objetivo passa por analisar e classificar os requisitos levantados na atividade anterior (Fernandes & Machado, 2017). Os requisitos levantados apresentam sempre problemas, por isso é preciso tratá-los para que se aproximem mais da solução. O desenvolvimento de requisitos é dividido nas seguintes tarefas (IEEE, 2014):

- Detetar e resolver conflitos entre requisitos;
- Descobrir os limites do sistema e como ele deve interagir com o seu ambiente operacional e organizacional;
- Desenvolver requisitos de sistema (normalmente derivados de requisitos de utilizador).

3.2.4. Negociação de Requisitos

Como é natural, todas as pessoas querem o produto mais vantajoso para elas, o que faz com que pretendam que o produto final tenha um número de características incompatíveis com o número de recursos disponíveis para o projeto (Pressman & Maxim, 2019).

Enquanto um cliente quer sempre gastar o menor número de recursos, os utilizadores querem um produto que satisfaça da melhor forma as suas necessidades, o que faz com que um produto nunca agrade totalmente a todos os *stakeholders*. Porém, é necessário encontrar uma solução que agrade minimamente a todos, e é precisamente este o objetivo da fase de negociação de requisitos.

A negociação, para Aurum & Wohlin (2005), são interações entre participantes que os levam a um acordo comum, e inicia-se quando os participantes começam a comunicar os seus objetivos, e acaba quando chegam a acordo. A negociação de requisitos compreende as seguintes tarefas (Aurum & Wohlin, 2005; Fernandes & Machado, 2017):

- 1. Pré-negociação:** Nesta fase faz-se a definição do problema, identificação de *stakeholders*, levantamento de objetivos e análise desses objetivos;
- 2. Negociação:** Os *stakeholders* chegam a um acordo sobre as questões em conflito;
- 3. Pós-negociação:** Os *stakeholders* analisam e avaliam as saídas da negociação e sugerem uma renegociação se necessário.

3.2.5. Documentação de Requisitos

Nesta atividade espera-se que o produto da engenharia de requisitos seja criado. Segundo Fernandes & Machado (2017), o produto da engenharia de requisitos pode ser um documento

ou vários, escritos em linguagem natural ou auxiliada de diagramas, com um grande nível de formalidade ou não, dependendo das características do sistema.

A atividade de documentação de requisitos é dividida em duas tarefas (Pandey et al., 2010):

- 1. Identificação dos requisitos:** Atribuição de um identificador único a cada requisito;
- 2. Especificação de requisitos:** O documento de especificação é produzido após os requisitos estarem devidamente identificados.

Para assegurar uma boa estrutura do documento de requisitos, os requisitos devem ser distinguidos pelo seu tipo (Fernandes & Machado, 2017).

3.2.6. Validação de Requisitos

A atividade de validação é o processo de avaliar sistemas no fim do seu processo de desenvolvimento para assegurar a conformidade com os requisitos (IEEE, 1990).

Para Boehm (1984, cit. por Pressman & Maxim, 2019), é necessário evitar confusões na diferença entre as atividades de validar e verificar. Enquanto a atividade de validar pressupõe que o projeto está a desenvolver o produto correto, a atividade de verificar questiona se o produto está a ser bem desenvolvido. A validação de requisitos tem por objetivo definir se o sistema está de encontro ao pretendido pelo cliente (Fernandes & Machado, 2017), são declarados de modo não ambíguo, e as suas inconsistências, omissões e erros são detetados e corrigidos (Pressman & Maxim, 2019).

3.2.7. Gestão de Requisitos

Para Pressman & Maxim (2019), gestão de requisitos é um conjunto de atividades que ajudam a equipa de desenvolvimento a identificar, controlar, rastrear e modificar requisitos.

Fernandes e Machado (2017) acrescentam que a gestão de requisitos é uma atividade paralela a todas as atividades da engenharia de requisitos, e esta inicia no fim da primeira iteração de todas as outras atividades, desde que haja uma mudança dos requisitos ao longo do processo de desenvolvimento. Para Davis (2005), a gestão de requisitos é um conjunto de atividades que consiste em recolher requisitos e identificar os corretos, para os satisfazer e documentar. Há 3 subconjuntos importantes de atividades da gestão de requisitos:

- Levantamento de requisitos: recolha de requisitos candidatos de clientes, utilizadores, especialistas do domínio e outros *stakeholders*;

- Triagem de requisitos: determinar quais os requisitos que devem ser satisfeitos quando analisados dentro do contexto dos recursos de desenvolvimento disponíveis, tempo de mercado, objetivos de receita, e retorno de investimento;
- Especificação de requisitos: documentar o comportamento externo do sistema desejado.

Segundo Cheng & Atlee (2009), a gestão de requisitos engloba atividades relacionadas com a gestão de projeto ou a fase de requisitos. Estas atividades incluem rastreabilidade, análise de impacto, estimativas de custo, gestão de risco e a gestão de variação de requisitos.

A gestão de requisitos está preocupada em gerir grandes quantidades de informação relacionadas com os requisitos que foram levantados nas fases anteriores (Aurum & Wohlin, 2005; Fernandes & Machado, 2017). A gestão de requisitos é uma atividade de engenharia de requisitos que é paralela a todas as atividades deste circuito, uma vez que desde que é feita a primeira iteração do processo de engenharia de requisitos, ou existe uma mudança em algum requisito, a gestão de requisitos começa a estabelecer relações com todas as atividades da engenharia de requisitos. É necessário constatar que os requisitos mudam durante fases mais avançadas do ciclo de vida do produto, logo a gestão de requisitos é uma atividade que é iniciada no final da primeira iteração do desenvolvimento dos requisitos e mantêm-se até ao final do ciclo de vida do produto.

3.3. Técnicas de análise de requisitos

Segundo Aguilar et al. (2017), “um dos fatores mais importantes para obter um software de sucesso é a primeira fase do processo de desenvolvimento de software: a engenharia de requisitos”. Sendo assim, uma vez que esta fase é tão vital e é um processo complexo, como já verificado acima, as técnicas a usar para fazer a análise de requisitos são diversas. Iremos de seguida ver as técnicas referidas por estes autores:

Entrevistas: A entrevista é a maneira de descobrir as opiniões que os *stakeholders* e outros utilizadores têm do sistema que está a ser desenvolvido. Esta pode ser realizada de duas maneiras: Entrevista fechada, em que o entrevistador tem um conjunto de perguntas pré-definidas e entrevista aberta, que em que ao invés do entrevistador ir com perguntas pré-definidas, é discutido o que é esperado do sistema. A vantagem desta técnica de recolha de requisitos é dar muita informação ao entrevistador, o ponto mais negativo é que dá muita informação qualitativa que pode ser difícil de analisar e pode haver opiniões divergentes dos diferentes entrevistados.

Análise de documentação: Esta técnica consiste no estudo de documentos relacionados com o sistema que vai ser implementado, documentos esses relacionados com os objetivos e atividade da organização.

Observação: Observar diretamente no local de trabalho as práticas profissionais desenvolvidas pela organização.

Casos de Uso: Esta técnica pretende definir os requisitos, retratando o fluxo completo de eventos para as partes interessadas na forma de um estilo de contar histórias. Os casos de uso são informais e fáceis de usar, ajudando a entender os requisitos e a validá-los com as partes interessadas.

Cenários: Uma vez que definem as interações entre os utilizadores e o sistema, os cenários são normalmente usados depois de se recolher os requisitos iniciais. Estes cenários são bastante uteis para validar requisitos e desenvolver casos de testes.

Focus Group: É um grupo de discussão informal e de tamanho reduzido com o propósito de obter informação qualitativa em profundidade. As pessoas são convidadas para participar da discussão sobre determinado assunto. Requer muito esforço para conduzir a reunião, pois é sempre difícil lidar com todas as partes interessadas ao mesmo tempo.

Brainstorming: O *brainstorming* é uma técnica de reuniões de grupo cujo objetivo é gerar ideias em um ambiente livre de críticas ou julgamentos. O *brainstorming* pode ajudar a gerar uma ampla variedade de visões do problema e formulá-lo de várias maneiras, especialmente no início do processo de engenharia de requisitos, quando os requisitos ainda são muito vagos.

Questionário: É um método de obtenção de requisitos que tem como principais vantagens ser simples e requerer menos tempo e custo. Para obter resultados precisos, o questionário deve ser claro, conciso e estruturado para obter requisitos, objetivos e restrições genuínos do utilizador. No entanto, essa técnica pode ter como desvantagens os participantes não responderem ou responderem de maneira displicente ao questionário.

Prototipagem: Ajuda os *stakeholders* a desenvolver uma forte noção sobre a aplicação, a qual ainda não foi implementada, uma vez que através da visualização da mesma eles podem identificar os reais requisitos e fluxos de trabalho do sistema. É muito utilizado quando os

stakeholders são incapazes de expressar os seus requisitos ou se os mesmos não têm nenhuma experiência com o sistema.

Forma do contrato: Consiste em preencher formulários indicando os requisitos. Pode ser extenso de acordo com o tamanho do projeto e, portanto, a sua utilização no desenvolvimento de software pode ser monótona.

Linguagem Natural: A linguagem natural é uma fonte importante de informação, porque na maioria dos domínios é o modo mais comum de representação do conhecimento. Existem duas categorias: interação direta com o utilizador usando linguagem natural e extração de requisitos de um documento de linguagem natural. O maior apelo da linguagem natural está em seu vocabulário, informalidade e sintaxe preexistentes. A existência de um vocabulário de milhares de palavras pré-definidas, usadas para descrever qualquer conceito possível, torna os meios de comunicação eficientes na língua natural. É familiar para o utilizador e o analista e não requer tempo para aprender. No entanto, existem duas limitações claras: o facto de a linguagem natural ser muito complexa e ambígua.

Na etapa de levantamento de requisitos pretendeu-se perceber a atividade que o sistema iria suportar. Esse levantamento implica o registo das atividades desenvolvidas e necessidades que são identificadas pelos utilizadores. Para realizar o levantamento das necessidades, foi utilizada a entrevista, que é uma técnica tradicional simples de utilizar e que, numa fase inicial, produz bons resultados na obtenção de dados e de fluxos de informação. Assim, foram utilizados métodos de conversação, que segundo Marcuschi (2006, p. 15) é “uma interação verbal centrada, que se desenvolve durante o tempo em que dois ou mais interlocutores voltam sua atenção visual e cognitiva para uma tarefa comum”, sendo um meio de comunicação que é uma forma natural de expressar necessidades e ideias, que ao responder às perguntas, é bastante eficaz para identificar e compreender as necessidades dos entrevistados, devendo o entrevistador dar espaço ao entrevistado para apresentar as suas necessidades. Foram efetuadas entrevistas a membros da administração da NPISA Barreiro e gestores de caso.

Foram também utilizados métodos analíticos de análise de conteúdos, que consistiram no estudo e reutilização de documentação para a identificação de requisitos a serem implementados. Uma grande variedade de documentação pôde ser analisada incluindo estrutura organizacional do núcleo e da estrutura que lida com esta temática em Portugal, relatórios anuais, leis, questionários, fichas de diagnóstico e planos de intervenção.

3.4. Metodologia de Desenvolvimentos de Software

Para Pressman & Maxim (2019), um processo de desenvolvimento de software não é uma receita rígida de como construir um software, mas sim uma abordagem que se consegue adaptar e que permite à equipa de desenvolvimento escolher o conjunto apropriado de ações e tarefas a fazer. “Uma estrutura de processos estabelece as fundações para um processo de engenharia de software completo ao identificar um pequeno número de estruturas de atividades que são aplicáveis a todos os projetos de software, independentemente do seu tamanho e complexidade.” (Pressman & Maxim, 2019, p.17) Para vários autores, a estrutura genérica do processo para a engenharia de software contém cinco atividades: comunicação, planeamento, modelação, construção e desenvolvimento (Sommerville, 2019), (Pressman & Maxim, 2019).

3.4.1. Metodologia de processos tradicional

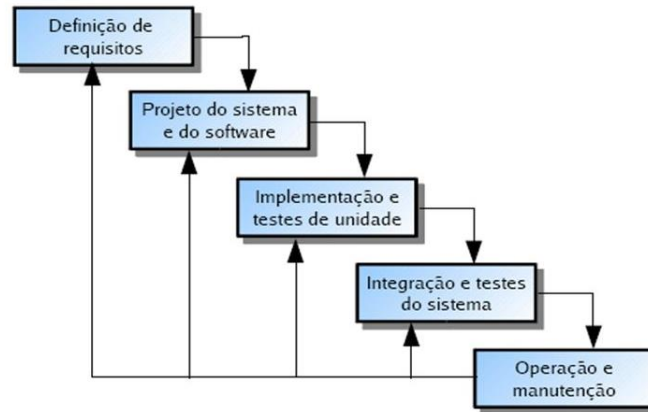
Os modelos tradicionais foram, segundo Pressman & Maxim (2019), originalmente propostos para trazer ordem ao caos que era o desenvolvimento de software, tendo conseguido alguma estrutura para o trabalho de engenharia de software e criado orientações para as equipas de software. De seguida serão apresentados alguns destes métodos, que afirma este autor, são caracterizados por serem consistentes e terem ordem. Eles também têm em comum determinar um conjunto de elementos: atividades estruturais, ações de engenharia de software, tarefas, produtos de trabalho, certificados de qualidade e mecanismos de troca de controlo para cada projeto. Cada um destes métodos também contém um fluxo de trabalho, ou seja, a maneira como os elementos estão interligados.

3.4.1.1. Modelo em Cascata

O modelo em cascata é um modelo que tem como ideia estruturante, segundo Engholm (2010), que para uma fase começar, a anterior deve estar terminada, seguindo assim um rumo linear e faseado por etapas. Ideia semelhante é defendida por Sommerville (2019), que afirma que o modelo em cascata é constituído por atividades sequenciais, em que uma atividade apenas se inicia após a atividade anterior estar terminada. Para Pressman & Maxim (2019), quando o trabalho flui de maneira linear, desde a fase de comunicação até à fase de desenvolvimento e quando os requisitos estão bem definidos e são estáveis, estão reunidas as condições para se fazer uma abordagem com o Modelo em Cascata. Para Sommerville (2019), estas fases são: Definição de Requisitos, Projeto do sistema e do software, implementação e testes de unidade,

integração e testes do sistema e operação e manutenção; estas correspondem respectivamente às fases de comunicação, planeamento, modelação, construção e desenvolvimento.

Figura 2 – Modelo em Cascata

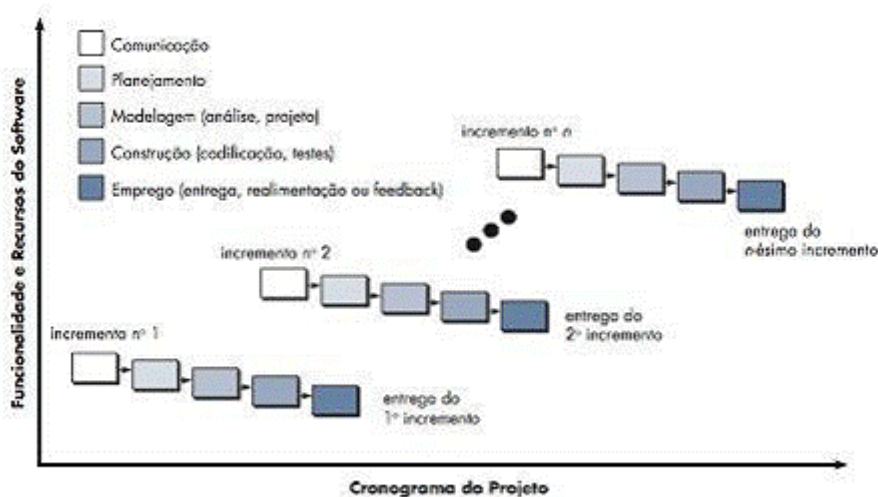


Fonte: Sommerville, (2019)

3.4.1.2. Modelo Incremental

O modelo incremental combina elementos de fluxos de processo lineares e paralelos. Este modelo aplica melhorias em processo linear, repetindo o processo e melhorando ao longo do tempo de vida do projeto (Pressman & Maxim, 2019). Para Sommerville (2019) este modelo utiliza os elementos do Modelo em Cascata, mas utilizando-os consecutivamente, de maneira a melhorar o projeto de cada vez que existe uma iteração. Quando o modelo incremental é usado, o primeiro incremento é, regra geral, o produto base, ou seja, preenche muitos dos requisitos básicos necessários mas muitas das funcionalidades suplementares continuam por fazer. Isto faz com que o produto seja avaliado pelos utilizadores, que assim dão feedback sobre que características devem ser desenvolvidas para melhorar as funcionalidades do produto, repetindo este processo após a implementação de cada incremento, até o produto estar finalizado (Fernandes & Machado, 2017). O modelo incremental foca-se em entregar um produto operacional em cada incremento. Os incrementos iniciais são versões mais básicas do produto final, mas mesmo assim têm a capacidade de servir o utilizador e oferecer uma plataforma que pode ser avaliada por este (Pressman & Maxim, 2019).

Figura 3 – Modelo Incremental



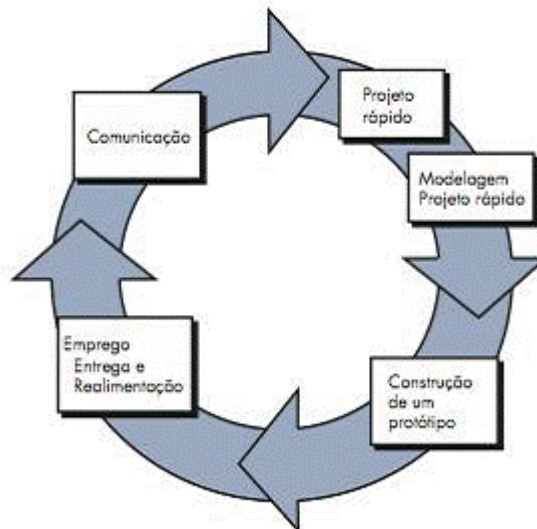
Fonte: Pressman & Maxim (2019)

3.4.1.3. Prototipagem

A prototipagem é um modelo que, juntamente com o modelo Espiral (o qual irá ser abordado de seguida) se engloba no grupo dos modelos evolutivos. Segundo Pressman & Maxim (2019), os modelos evolutivos são iterativos, o que permite desenvolver versões cada vez mais completas do software. Um protótipo é uma versão inicial de um sistema de software, usado para demonstrar conceitos, experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções (Sommerville, 2019). Para Pressman & Maxim (2019), o modelo de prototipagem permite a quem desenvolve e aos *stakeholders* terem uma ideia mais clara do que desenvolver em alturas em que os requisitos possam ser mais confusos. Protótipos do sistema permitem aos utilizadores ver quão bem o sistema dá suporte ao seu trabalho. Eles podem obter novas ideias para requisitos e encontrar pontos fortes e fracos do software; podem, então, propor novos requisitos do sistema. Além disso, o desenvolvimento do protótipo pode revelar erros e omissões nos requisitos propostos. A função descrita em uma especificação pode parecer útil e bem definida. No entanto, quando essa função é combinada com outras, os utilizadores muitas vezes percebem que a sua visão inicial foi incorreta ou incompleta. A especificação do sistema pode então ser modificada para refletir o entendimento dos requisitos alterados (Sommerville, 2019). Segundo Pressman & Maxim (2019), o protótipo começa com a comunicação. A equipa de desenvolvimento encontra-se com os outros *stakeholders* para definir os objetivos gerais do software, identificar que requisitos são conhecidos e sublinhar que áreas que necessitam de definição. O protótipo é planeado rapidamente, e a modelação, na forma de projeto rápido, ocorre. Esse projeto rápido foca-se na representação dos aspetos do software que ficarão

visíveis aos utilizadores finais. Daqui segue-se a construção do protótipo. O protótipo é testado e avaliado pelos *stakeholders* que dão feedback que é usado para melhorar os requisitos. Isto faz com que o protótipo seja afinado para satisfazer as necessidades de muitos dos *stakeholders*, enquanto ao mesmo tempo permite à equipa de desenvolvimento perceber melhor o que tem de ser feito.

Figura 4 - Prototipagem



Fonte: Pressman & Maxim (2019)

3.4.1.4. Espiral

O modelo Espiral é um modelo evolutivo que combina a natureza interativa da prototipagem com o controlo e sistematização do Modelo em Cascata (Pressman & Maxim, 2019).

Este modelo foi pela primeira vez proposto por Boehm em 1988, e foi descrito por ele como “um gerador de processos orientado para o risco, usado para guiar a engenharia de sistemas intensiva simultaneamente entre múltiplos *stakeholders*. Ele tem duas características que o distinguem. Uma é ser uma abordagem cíclica para incrementar o grau de definição e implementação do sistema, ao mesmo tempo que reduz o seu nível de risco. A outra é ser um conjunto de marcos que garantem que o comprometimento do *stakeholder* em achar soluções para o sistema que sejam factíveis e satisfatórias para todas as partes.” (Pressman & Maxim, 2019, p. 48). Já Sommerville (2019), caracteriza-o como um modelo que combina prevenção e tolerância a mudanças, assume que mudanças são um resultado de riscos de projeto e inclui atividades explícitas de gestão de riscos para a sua redução. Ao usar este modelo, o software é desenvolvido numa série de lançamentos evolutivos, que nos primeiros lançamentos pode ser um protótipo, e quanto mais adiantado está o estágio, mais completas são as versões lançadas

3.4.2. Metodologia de processos especializada

Esta metodologia de processos transporta muitas das características dos modelos de processos tradicionais apresentados anteriormente, no entanto estes modelos são aplicados quando é escolhida uma abordagem de engenharia de software mais específica (Pressman & Maxim, 2019).

3.4.2.1. Desenvolvimento baseado em componentes

Este modelo incorpora muitas das características do modelo em espiral, no entanto ele contrói aplicações a partir de componentes de software existentes (Pressman & Maxim, 2019). Este modelo começa por identificar os componentes que podem ser usados. Segundo Pressman & Maxim (2019), independentemente da tecnologia usada para criar os componentes, o modelo incorpora os seguintes passos:

1. Os produtos baseados em componentes disponíveis são pesquisados e avaliados para o uso específico nesta aplicação;
2. São considerados os problemas relacionados com a integração dos componentes;
3. É desenhada uma arquitetura de sistemas que possa acomodar os componentes;
4. Os componentes são integrados na arquitetura;
5. Testes são conduzidos para garantir que funcionam corretamente.

3.4.2.2. Modelo de métodos formais

O modelo de métodos formais permite especificar, desenvolver e verificar um sistema aplicando uma linguagem rigorosa e matemática. Segundo Pressman & Maxim (2019), quando este método é usado durante o desenvolvimento, ele proporciona um mecanismo para eliminar muitos dos problemas que são difíceis de ultrapassar usando outros métodos, como a ambiguidade e inconsistências que podem ser descobertos mais facilmente usando a aplicação dos métodos de análise matemáticos. Este modelo tem como principal vantagem entregar um software livre de defeitos, no entanto as suas limitações impedem que se torne num modelo *mainstream*. Essas limitações são o excessivo tempo e dinheiro que este modelo consome, a complexidade que faz com que não haja muitos desenvolvedores habilitados a fazê-lo e que seja difícil explicá-lo aos utilizadores.

3.4.3. Metodologia de processo unificado

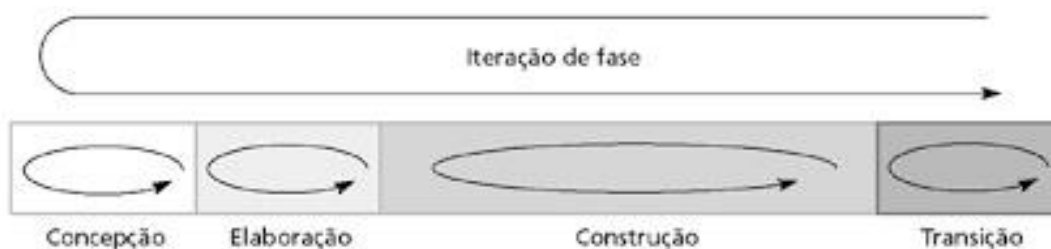
O modelo de processo unificado, também chamado de *Rational Unified Process* (RUP), foi pela primeira vez abordado por Jacobson, Booch & Rumbaugh em 1999, na sua obra *The Unified Process*. Segundo Pressman & Maxim (2019), o Processo Unificado reconhece a importância da comunicação com o consumidor e do uso de métodos simplificados para descrever a visão do consumidor em relação ao sistema (os casos de uso). Este modelo ajuda o arquiteto a focar-se nos objetivos certos, como a compreensibilidade, confiança em mudanças futuras e reutilização (Jacobson et al., 1999), colocando assim ênfase na importância do papel da arquitetura de software. Estes autores começaram por propor uma análise orientada a objetos e métodos de design, que acabariam por criar a *Unified Modeling Language* (UML). Esta linguagem proporciona a tecnologia necessária para suportar a engenharia de software orientada a objetos, no entanto não proporciona a estrutura de processos para guiar as equipas de desenvolvimento na sua aplicação da tecnologia (Pressman & Maxim, 2019) – e é então que surge o modelo unificado de processo, um modelo iterativo e incremental que pode ser adaptado para as necessidades específicas do projeto.

Para Sommerville (2019), O RUP é um modelo que identifica quatro fases distintas no processo de software (Conceção, Elaboração, Construção e Transição) enquanto para Pressman & Maxim (2019), existe mais uma fase (Produção). Estas fases são caracterizadas da seguinte maneira:

1. **Concepção:** Nesta fase deve-se estabelecer um business case para o sistema, e deve-se identificar todos das entidades que irão interagir com o sistema e definir como irão ser essas interações. Nesta fase deve-se avaliar a contribuição do sistema para o negócio para avaliar a viabilidade do projeto.
2. **Elaboração:** Nesta fase deve-se compreender o problema, estabelecer uma estrutura de arquitetura para o sistema, desenvolver o plano de projeto e identificar os riscos. Após esta fase deverá existir um modelo de requisitos para o sistema.
3. **Construção:** Nesta fase, funcionam em simultâneo o projeto, a programação e os testes do sistema. No final desta fase deverá existir um sistema já a funcionar.
4. **Transição:** Esta fase implica a transferência do sistema da comunidade de desenvolvimento para a comunidade de utilizadores e o seu funcionamento em ambiente real.

5. **Produção:** Nesta fase existe a implantação do software. Nesta fase é avaliada a utilização do software, é criada a estrutura de suporte para o funcionamento do software e os *reports* de defeitos e pedidos para alterações no software são submetidos e avaliados.

Figura 6 – Metodologia do Processo Unificado



Fonte: Sommerville, (2019)

3.4.4. Metodologia de processo pessoal e de equipa

Para Pressman & Maxim (2019), o melhor processo de construção de software deve ser um que seja próximo das pessoas que irão trabalhar com ele, pois num cenário ideal estar-se-á a construir um processo que responda tanto às necessidades de quem está a criar como da equipa e da organização que o irá utilizar. Com disciplina, trabalho e coordenação, é possível criar um processo de software pessoal ou um processo de software de equipa (Ferguson et al., 1997).

3.4.4.1. Processo de software pessoal

Segundo Pressman & Maxim (2019), este processo pega nos processos pessoais que os desenvolvedores têm para construir software, com ênfase na avaliação entre a quantidade de trabalho produzido e a qualidade do mesmo. Segundo este autor, o modelo define cinco atividades estruturais:

1. **Planeamento:** Define-se os requisitos, estima-se o número de defeitos que existirão, são identificadas as tarefas de desenvolvimento e é criado um calendário para o projeto.
2. **Design de Alto Nível:** As especificações externas para cada componente são construídas e desenvolvidas e é definido um design para os componentes. Nesta fase constroem-se protótipos, quando necessário, e os incidentes são registados e seguidos.
3. **Revisão do Design de Alto Nível:** Nesta fase faz-se a verificação dos erros identificados na fase anterior. Recolhem-se dados de tarefas importantes e resultados do trabalho.
4. **Desenvolvimento:** Aqui, o design de componentes é refinado e revisto. É feito o código para o software, é revisto, compilado e testado. Recolhem-se dados de tarefas importantes e resultados do trabalho.

5. **Autópsia:** Analisando os dados recolhidos nas outras fases, é definido o grau de efetividade do processo. Estes dados devem guiar o caminho para se melhorar a efetividade dos processos.

Este processo é uma abordagem disciplinada e baseada em dados métricos, e quando bem aplicado pode, segundo Ferguson et al. (1997), resultar em melhorias na produtividade e qualidade do software.

3.4.4.2. Processo de Software de equipa

O objetivo deste processo é, segundo Ferguson et al. (1997), construir uma equipa que se auto-organiza para produzir software de alta qualidade. Os objetivos para este processo são:

1. Construir equipas que planeiam e monitorizam o seu trabalho, estabelecem objetivos e possuem os seus processos e planos. Estas equipas devem ter entre 3 a 20 engenheiros.
2. Mostrar aos gestores como treinar e motivar as suas equipas e como devem manter a performance no máximo.
3. Fornecer orientação para melhoramentos em organizações com alto nível de maturidade.
4. Facilitar o ensino em Universidades de qualidades técnicas de nível industrial.

Para Pressman & Maxim (2019), uma equipa auto-organizada tem consciência dos seus objetivos gerais, define papéis e responsabilidades para cada membro da equipa, analisa dados quantitativos do projeto, consegue identificar um processo apropriado para o projeto e define como o vai implementar, define os standards usados para avaliar o trabalho desenvolvido, está de maneira contínua a verificar os riscos e corrigi-los, assim como rastreia, gere e atualiza o status do projeto.

3.5. Metodologia Agile

Atualmente as empresas operam globalmente, num mundo que está constantemente em mudança, o que faz com que precisem de responder de maneira rápida e constante a novas oportunidades de negócio ou ameaças que surjam dos seus concorrentes. Segundo Pressman & Maxim (2019), na economia moderna é difícil prever como um sistema informático vai evoluir com o passar do tempo. O mercado muda constantemente, as necessidades dos utilizadores também e novas ameaças surgem inesperadamente, o que faz com que muitas vezes não seja possível definir totalmente os requisitos antes de iniciar o projeto – é necessário ser ágil o suficiente para responder a toda esta fluidez do mercado. Segundo Sommerville (2019), para as empresas o desenvolvimento e entrega rápidos são o requisito mais crítico para o

desenvolvimento de sistemas de software, estando dispostas por isso a trocar a qualidade e o compromisso com requisitos do software por uma implantação mais rápida do software de que necessitam. Embora existam muitas abordagens para o desenvolvimento rápido de software, elas compartilham algumas características fundamentais (Sommerville, 2019):

1. Os processos de especificação, projeto e implementação são intercalados. Não há especificação detalhada do sistema, e a documentação do projeto é minimizada ou gerada automaticamente pelo ambiente de programação usado para implementar o sistema. O documento de requisitos do usuário apenas define as características mais importantes do sistema.
2. O sistema é desenvolvido em uma série de versões. Os usuários finais e outros *stakeholders* do sistema são envolvidos na especificação e avaliação de cada versão. Eles podem propor alterações ao software e novos requisitos que devem ser implementados em uma versão posterior do sistema.
3. Interfaces de utilizador do sistema são geralmente desenvolvidas com um sistema interativo de desenvolvimento que permite a criação rápida do projeto de interface por meio de desenho e posicionamento de ícones na interface. O sistema pode, então, gerar uma interface baseada na Web para um navegador ou uma interface para uma plataforma específica, como, por exemplo, o Microsoft Windows.

Para Sommerville (2019), existem diferentes métodos ágeis, que se distinguem pelas diferentes propostas de processos para alcançar tal objetivo. Apesar de diferentes propostas, todos esses métodos são baseados na noção de desenvolvimento e entrega incremental, e, portanto, comungam um conjunto de princípios, como se pode ver na tabela 2:

Tabela 2 – Os princípios dos métodos ágeis

Princípios	Descrição
Envolvimento do cliente	Os clientes devem estar intimamente envolvidos no processo de desenvolvimento. O seu papel é fornecer e priorizar novos requisitos do sistema e avaliar as suas iterações.
Entrega incremental	O software é desenvolvido em incrementos com o cliente, especificando os requisitos para serem incluídos em cada um.

Pessoas, não processos	As habilidades da equipe de desenvolvimento devem ser reconhecidas e exploradas. Membros da equipe devem desenvolver as suas próprias maneiras de trabalhar, sem processos prescritivos.
Aceitar as mudanças	Deve-se ter em mente que os requisitos do sistema vão mudar, por isso, deve-se projetar o sistema de maneira a acomodar essas mudanças.
Manter a simplicidade	Focar a simplicidade, tanto do software a ser desenvolvido quanto do processo de desenvolvimento. Sempre que possível, deve-se trabalhar ativamente para eliminar a complexidade do sistema.

Fonte: Adaptado de Sommerville, (2019)

3.5.1. *Extreme Programming*

O *Extreme Programming* é a abordagem ao desenvolvimento *Agile* mais conhecida e difundida (Pressman & Maxim, 2019), (Sommerville, 2019). Em *Extreme Programming*, os requisitos são expressos como cenários, que são implementados diretamente como uma série de tarefas. Os programadores trabalham em pares e desenvolvem testes para cada tarefa antes de escreverem o código. Quando o novo código é integrado no sistema, todos os testes devem ser executados com sucesso. Há um curto intervalo entre os *releases* do sistema (Sommerville, 2019). *Extreme Programming* envolve uma série de práticas que refletem os princípios dos métodos ágeis (Sommerville, 2019):

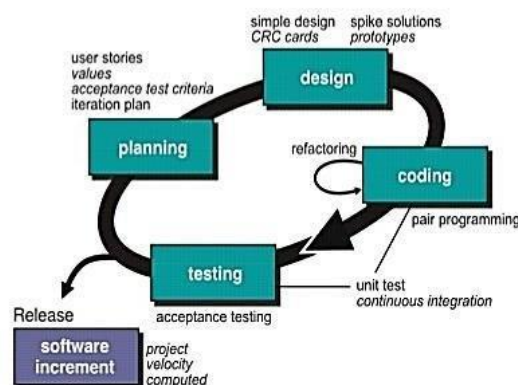
1. **O desenvolvimento incremental** é sustentado por meio de pequenos e frequentes *releases* do sistema. Os requisitos são baseados em cenários ou em simples histórias de clientes, usadas como base para decidir a funcionalidade que deve ser incluída em um incremento do sistema.
2. **O envolvimento do cliente** é sustentado por meio do engajamento contínuo do cliente com a equipe de desenvolvimento. O representante do cliente participa do desenvolvimento e é responsável por definir os testes de aceitação para o sistema.
3. **Pessoas — não processos** — são sustentadas por meio de programação em pares, propriedade coletiva do código do sistema e um processo de desenvolvimento sustentável que não envolve horas de trabalho excessivamente longas.
4. **As mudanças** são feitas por meio de *releases* contínuos para os clientes, do desenvolvimento *test-first*, da refatoração para evitar a degeneração do código e integração contínua de novas funcionalidades.

5. A **manutenção da simplicidade** é feita por meio da refatoração constante que melhora a qualidade do código, bem como por meio de projetos simples que não antecipam desnecessariamente futuras mudanças no sistema.

Segundo Pressman & Maxim (2019), estas regras e práticas ocorrem dentro do processo, que contém quatro estruturas diferentes: planejamento, design, programação e teste.

Figura 7 – O processo *Extreme Programming*

Extreme Programming (XP)



Fonte: Adaptado de Pressman & Maxim (2019).

O método *Extreme Programming* tem sido alvo de discussões e análises no meio acadêmico. Segundo Stephens & Rosenberg (2003), apesar de haver muitas práticas desta metodologia que são valiosas, outras foram valorizadas em demasia e algumas são até problemáticas. Para estes autores, a natureza interdependente da metodologia *Extreme Programming* é tanto uma mais-valia como pode ser uma fraqueza, especialmente se as empresas apenas adotarem algumas das suas práticas, reduzindo assim a eficácia de todo o processo. Entre estas práticas são identificadas como especialmente problemáticas a volatilidade dos requisitos, as necessidades conflitantes dos utilizadores, a recolha dos requisitos informalmente e a falta de design formal.

3.5.2. *Scrum*

Os princípios do *Scrum* são usados para guiar atividades de desenvolvimento num processo que incorpora a seguinte estrutura de atividades: requerimentos, análise, design, evolução e entrega (Pressman & Maxim, 2019). Segundo Pressman & Maxim (2019), dentro de cada uma destas atividades da estrutura ocorrem as tarefas com um padrão de processo chamado

“sprint”. Esses sprints são um conjunto de unidades de trabalho que têm como objetivo realizar um dos requisitos definidos previamente, dentro de determinado tempo. Isto permite que os membros da equipa trabalhem num ambiente estável, apesar de ser com objetivos de curto-prazo. Para Sommerville (2019), a característica inovadora do *Scrum* é esta fase. Um sprint do *Scrum* é uma unidade de planeamento na qual o trabalho a ser feito é avaliado, os recursos para o desenvolvimento são selecionados e o software é implementado. No fim de um sprint, a funcionalidade completa é entregue aos *stakeholders*. Para além do *Sprint*, existem mais três padrões de processo: o *backlog*, as reuniões *scrum* e as *demos*. O *backlog* são o conjunto de requisitos, listados por prioridades, que acrescentam valor para o utilizador; As reuniões *scrum* são rápidas reuniões (com duração normal de 15 minutos) realizadas diariamente pela equipa *scrum*, onde três perguntas são respondidas por cada um: O que fez desde a última reunião de equipa, que obstáculos encontrou e o que espera ter realizado aquando da próxima reunião; As *demos* são as entregas do software incrementado ao utilizador, de maneira a que ele possa avaliar o que foi implementado.

3.5.3. Desenvolvimento de Software *Lean*

Este método de desenvolvimento adapta os princípios de trabalho *Lean* para o mundo da engenharia de software.

Figura 8 – Princípios do desenvolvimento *Lean*



Fonte: Agile Methods: LSD, Projetos e TI, 18 de Maio de 2020

Segundo Poppendieck & Poppendieck (2013), estes princípios são sete: Eliminar o desperdício, integrar qualidade, criar conhecimento, adiar compromissos, entregar depressa, respeitar as pessoas e otimizar o todo.

3.6 Sistematização da metodologia

Face ao exposto, a metodologia subjacente ao trabalho, e atendendo à especificidade da problemática em estudo, a metodologia adotada é a DSR. Como estruturação da solução subjacente a Pssa, apresentam-se os seguintes diagramas e tabelas:

- Diagrama de Caso de Uso;
- Tabela de descrição de Casos de Uso;
- Diagrama de Atividades;
- Diagrama de Estados;
- Diagrama de Sequência;
- Diagrama Entidade-Relação.

Os diagramas elaborados são os que se consideram como sendo os adequados para descrever o problema e o modo como a aplicação irá funcionar.

4. Caracterização da Organização em estudo

Neste capítulo irá ser analisada a instituição e o caso a tratar. Serão apresentadas as diversas tabelas e diagramas que ajudam a explicar a situação e a dar uma ideia do que se pretende para a aplicação, a que necessidades deverá responder e como o irá fazer.

4.1. A Instituição

A Instituição para a qual o projeto é destinado é a Cáritas de Setúbal. Como refere no site oficial, “A Cáritas de Setúbal é um serviço da Igreja Diocesana para a promoção da sua ação social. Tem como atividade primordial a animação da pastoral social que visa a criação e funcionamento de serviços paroquiais para o melhor conhecimento dos problemas. É a partir deste conhecimento que procura agir diretamente na prevenção e solução dos problemas. Dada a complexidade dos fenómenos sociais contemporâneos, resultante das suas características multidimensionais, outras das preocupações fundamentais é o contributo possível para a transformação social, nomeadamente nos domínios das relações sociais, dos valores, em ordem ao desenvolvimento solidário”. Ainda no site oficial, pode ler-se que a instituição tem como um dos fundamentos ser ““a opção preferencial da Igreja pelos pobres”, e os mais pobres dos pobres são todos aqueles que associam à sua pobreza a exclusão social. Nesse sentido, são disponibilizadas, entre outras, respostas para indivíduos portadores de deficiência, sem abrigo e passantes, toxicodependentes, alcoólicos, mulheres em situação de risco, seropositivos e doentes com sida”. Como se pode ver por esta descrição, ajudar os pobres, neste caso as Pssa, está na génese desta instituição, e é para a ajudar nesta missão que este projeto foi feito.

4.2. A problemática

A necessidade organizacional de possuir uma plataforma que permitisse armazenar e informatizar toda a informação relativa aos casos de Pssa acompanhados é algo sentido na instituição há algum tempo. Assim, foi desenvolvido um protótipo não funcional que permita armazenar a informação recolhida junto de Pssa, com as bases para o desenvolvimento futuro de uma solução que deve ter a capacidade de tratar a informação, permitindo criar estatísticas, construir gráficos e tabelas com a informação armazenada. Considera-se também premente que os Gestores de Caso rentabilizem o tempo que passam com as Pssa, o que poderia ser conseguido se passassem menos tempo em processos burocráticos, como o preenchimento de inquéritos quando estão junto de Pssa e a passagem destes para o formato informático existente (um ficheiro Excel com a informação de Pssa), o que exigia uma duplicação de trabalho. A

ideia para acabar com esta duplicação de trabalho, e até para permitir novas interações e dinâmicas dos Gestores de Caso no contacto com as Pssa, foi sugerido que a solução também tivesse formato móvel, podendo assim ser uma *App* instalada no *smartphone* dos Gestores de Caso, possível de sincronizar com a base de dados.

A realização da investigação iniciou-se com a realização de reuniões com representantes da instituição, que irão expor o problema e indicar quais as necessidades que existem e às quais pretendem que o projeto responda, assim como facultar informação necessária para se delinear o artefacto. Com a informação fornecida realizar-se-á a análise de requisitos necessária para perceber que informação é relevante para a associação e como a podemos estruturar. Após isso estar definido, seguir-se-á a fase de construção e desenvolvimento da aplicação, seguida dos testes e pequenas “afinações” sugeridas pela Cáritas Setúbal para aproximar o projeto do pretendido, de maneira a melhorar até ter o produto final.

4.3. Tabela de Casos de Uso

Uma das técnicas que iremos utilizar para fazer o levantamento de requisitos será a dos casos de uso. Um caso de uso é uma descrição das possíveis sequências de interações entre o sistema sob discussão e seus atores externos, relacionados a um objetivo específico (Cockburn, 2000). Segundo Booch et al. (2012), “um caso de uso é uma descrição de um conjunto de sequências de ações, inclusive variantes, que um sistema executa para produzir um resultado de valor observável por um ator.” Uma vez que o caso de uso descreve interações entre o sistema e os seus atores, deve-se criar uma tabela de casos de uso, que contenha os atores que vão fazer parte do sistema, assim como os seus objetivos.

Tabela 3 – Casos de Uso do Sistema

Ator	Objetivo
Sinalizador	- Sinalizar Pssa
Equipa de Emergência	- Criar novo Processo
Coordenador	- Enviar equipa de emergência para confirmar sinalização - Criar Utilizadores - Atribuir o Caso a um Gestor de Caso - Consultar Processos
Gestor de Caso	- Consultar Processos - Editar Processo - Inserir Ficha de Avaliação Diagnóstico - Inserir Plano de Intervenção
Pssa	- Beneficiar do Plano de Intervenção definido

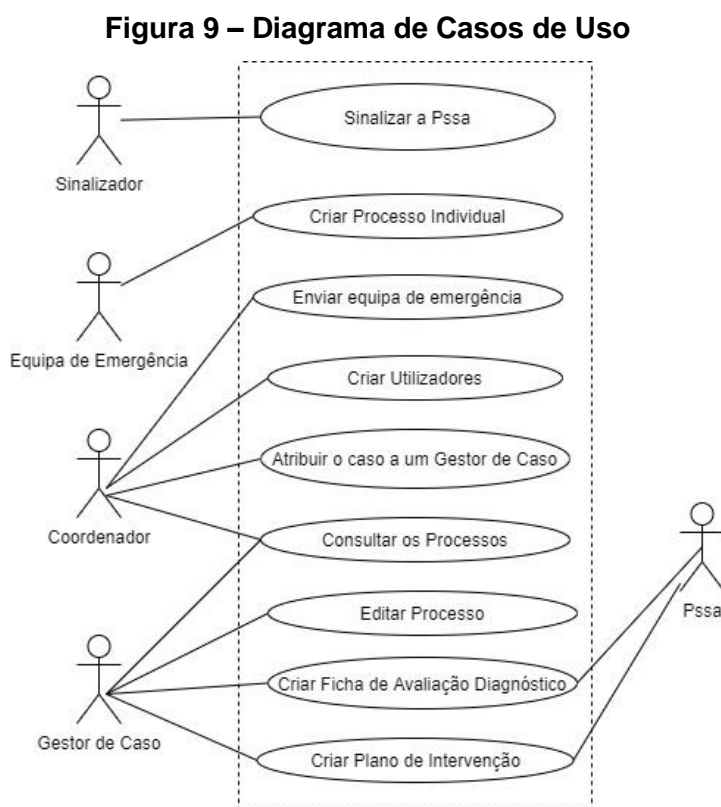
Esta tabela mostra o papel que cada ator tem no sistema:

- **Sinalizador:** É um utilizador final do sistema. O perfil de Sinalizador é um perfil único, que permite a qualquer pessoa, mediante uma validação rápida (facultar o NIF e o nº do CC), aceder ao sistema para sinalizar. Ao se deparar com uma nova Pssa na cidade, utiliza o sistema para dar o alerta às autoridades competentes, apenas o podendo usar para isso.
- **Equipa de Emergência:** É um utilizador final do sistema, que irá interagir com o mesmo quando for enviada a um local onde foi sinalizada uma Pssa, com o objetivo de confirmar a sua existência, estabelecer com ela o primeiro contacto da instituição e criar um novo processo para essa pessoa.
- **Coordenador:** É um utilizador final do sistema, que pode interagir com o mesmo para enviar equipas de emergência que confirmem a sinalização, criar utilizadores (gestores de caso e equipas de emergência) e atribuir-lhes perfis de utilizador, atribuir um novo caso a um Gestor de Caso e consultar os processos por estes criados.
- **Gestor de Caso:** É um utilizador final do sistema, que o irá utilizar para consultar processos já existentes, editar processos, inserir fichas de avaliação diagnóstico e planos de intervenção nos processos.

- **Pssa:** A Pssa é alguém que não irá ter contacto direto com o sistema, apenas providenciar os dados necessários.

4.4. Diagrama de Casos de Uso

Para Booch et al. (2012), um diagrama de caso de uso é um diagrama que mostra um conjunto de casos de uso e atores e seus relacionamentos. Para estes autores, o diagrama de casos de uso é uma das peças principais quando se trata de modelar o comportamento de um sistema, mostrando todas as suas funcionalidades, ao meter em evidência interações externas com outras entidades. Na Figura 9 é apresentado o diagrama de casos de uso do sistema. Depois de analisados os atores do sistema e os objetivos destes, é altura de avançar para a fase seguinte: desenhar um diagrama de casos de uso com os atores especificados anteriormente, tal como os casos de uso e as interações dentro do sistema, pretendendo-se com ele mostrar como vai funcionar o sistema.



Face ao exposto na Figura 9, salienta-se que todos os casos de uso pressupõem que exista um *login* válido. Para proceder a cada operação será necessário os atores estarem registados e introduzirem email e password válidos no sistema.

4.5. Descrição de Casos de Uso

Uma vez identificada a forma como os atores irão interagir com o sistema, de seguida irá se documentar os casos de uso. Cada caso de uso precisa de ser documentado com o fluxo de eventos e isto é feito tendo por base a visão dos atores que fazem parte do processo. Esta descrição deverá conter o que o sistema tem de fornecer ao ator quando este executa o caso de uso, mostrando habitualmente como começa e acaba o caso de uso. Segundo Cockburn (2000), deverá ser especificado o que se espera que o sistema forneça ao ator aquando da execução do caso de uso, mostrando-o do início ao fim, mostrando a razão da necessidade do sistema e porque são uteis como auxiliar na análise de requisitos do sistema.

Não há uma forma *standard* de fazer a descrição do caso de uso, havendo vários *templates* propostos por diferentes autores. Rebecca Wirfs-Brock (1993), apresenta como hipótese para descrever os casos de uso o método da conversação, em que tem duas colunas, sendo que na primeira coluna tem as ações do ator principal e na coluna da direita as ações do sistema. O *template* proposto por (Cockburn, 2000), tem como principais características: uma coluna de texto em prosa, os passos numerados, não tem ações com “*if*” e a convenção de numeração nas seções de “*extensões*” que envolvem combinações de dígitos e letras. Neste caso será usado um *template* semelhante ao proposto pela *Rational Software Corp* (2001), abaixo ilustrado.

Tabela 4 – Descrição de Caso de Uso

Nome	Nome do caso de uso descrito.
Breve Descrição	Descrição resumida do caso em questão.
Atores	Intervenientes do caso de uso.
Trigger	O que desencadeia o caso de uso.
Pré-Condição	O que o sistema precisa que aconteça antes de começar o caso de uso.
Cenário de Sucesso	É o cenário de como tudo corre, partindo do princípio de que não ocorrem erros nem problemas e de que acontece o resultado esperado.
Fluxos Secundários	São os cenários que podem acontecer no caso de algo correr mal.
Pós-Condição	É o resultado do caso de uso.
Requisitos Especiais	Requisitos não funcionais relacionados com o caso de uso.

De seguida serão descritos os casos de uso apresentados anteriormente, recorrendo a este *template*.

Sinalização da Pessoa em Situação de Sem-Abrigo

Quando o ator “Sinalizador” interage com o sistema, vai desencadear uma sequência de eventos que permitem desenvolver este caso de uso. A tabela seguinte descreve o processo que o Sinalizador desencadeia ao sinalizar uma Pessoa em Situação de Sem Abrigo.

Tabela 5 - Descrição de Caso de Uso "Sinalizar a Pessoa em Situação de Sem-Abrigo"

Nome	Sinalizar a Pessoa em Situação de Sem-Abrigo.
Breve Descrição	Este caso de uso visa descrever o processo de sinalização de Pessoa em Situação de Sem-Abrigo, inserindo no sistema a localização de uma nova pessoa nessa situação.
Atores	Sinalizador.
Trigger	n.a.
Pré-Condição	Login Válido.
Cenário de Sucesso	<ol style="list-style-type: none">1. O ator seleciona a opção “Sinalizar Pessoa em Situação de Sem-Abrigo”;2. O sistema apresenta um mapa (pode ser feito usando o <i>maps</i> da <i>Google</i>, por exemplo);3. O ator seleciona o ponto no mapa onde se encontra a pessoa a sinalizar;4. O sistema apresenta uma janela com a morada desse ponto e a opção para cancelar ou gravar;5. Gravar;6. O sistema envia uma notificação “Sinalização Efetuada Com Sucesso”.
Fluxos Secundários	<ol style="list-style-type: none">3.a) O sistema não reconhece aquele ponto;4.a) Sistema não tem morada para o ponto indicado;5.a) O ator opta por cancelar a ação.
Pós-Condição	Sistema envia automaticamente uma mensagem e email ao Coordenador, a informar que foi feita uma nova sinalização.
Requisitos Especiais	n.a.

Este é o primeiro passo para a Pssa entrar no sistema. É depois de ser feita esta ação que todo o processo se desencadeia.

Enviar equipa de emergência

Depois de efetuada com sucesso a sinalização, serão enviados para o coordenador tanto uma mensagem no sistema como um email automático a avisar que uma nova sinalização foi feita. De seguida o coordenador irá enviar uma equipa de emergência ao local para confirmar a localização e recolher os dados da pessoa sinalizada.

Tabela 6 – Descrição do caso de uso “Enviar equipa de emergência”.

Nome	Enviar equipa de emergência.
Breve Descrição	Após ser sinalizado no sistema um novo caso de pessoa em situação de sem-abrigo, o coordenador envia uma equipa de emergência ao local para confirmar esta sinalização.
Atores	Coordenador.
Trigger	Nova sinalização no sistema.
Pré-Condição	Login Válido.
Cenário de Sucesso	<ol style="list-style-type: none"> 1. O ator seleciona a opção “Enviar Equipa Emergência”; 2. O sistema apresenta uma lista com as Equipas de Emergência existentes; 3. Selecionar a Equipa de Emergência pretendida; 4. Associar a sinalização a essa equipa; 5. Enviar; 6. O sistema mostra a notificação “Equipa de Emergência Enviada”.
Fluxos Secundários	<ol style="list-style-type: none"> 2. a) Não existem Equipas de Emergência criadas; 5. a) O ator opta por cancelar a ação.
Pós-Condição	Sistema envia automaticamente mensagem e email à equipa a informar que lhe foi atribuído no sistema uma nova sinalização.
Requisitos Especiais	n.a.

Esta ação será a que validará a existência da Pssa referida e permitirá o primeiro contacto de membros da organização com a mesma.

Criar Processo Individual

Uma vez confirmado com sucesso a existência de uma nova pessoa em situação de sem-abrigo na zona, cabe à equipa de emergência criar o processo, ao inserir a informação correspondente à pessoa identificada. Na tabela seguinte iremos ver a descrição deste processo.

Tabela 7 – Descrição do caso de uso “Criar Processo Individual”.

Nome	Criar Processo Individual.
Breve Descrição	A “Equipa de Emergência”, depois de ser enviada a uma nova sinalização pelo coordenador, e de confirmar a existência de uma nova pessoa em situação de sem-abrigo, cria um processo para essa pessoa com os seus dados de identificação.
Atores	Equipa de Emergência.
Trigger	n.a.
Pré-Condição	Login Válido.
Cenário de Sucesso	<ol style="list-style-type: none">1. O ator seleciona a opção “Criar novo processo”;2. O sistema exibe uma lista com as sinalizações à espera de criação de processo individual;3. O ator seleciona a correspondente;4. O sistema apresenta uma tabela com os dados de identificação para preencher;5. Preencher os itens obrigatórios e gravar;6. O sistema mostra uma notificação “Novo Processo Criado com Sucesso”.
Fluxos Secundários	<ol style="list-style-type: none">3. a) Tentar gravar o processo sem um dos itens obrigatórios estar preenchido;3. b) O ator opta por cancelar a ação.
Pós-Condição	n.a.
Requisitos Especiais	n.a.

A ação da equipa de emergência é extremamente importante, tanto porque são eles que vão “validar” a veracidade da sinalização, assim como introduzir os primeiros dados no sistema da

Pssa que a irão identificar, como por os membros desta equipa serem os primeiros da organização a terem contacto com a Pssa.

Criar Utilizador

O Coordenador terá as permissões para criar outros utilizadores, nomeadamente gestores de caso. Quando o ator “Coordenador” interage com o sistema para fazer isso, vai desencadear este caso de uso.

Tabela 8 – Descrição de caso de uso “Criar Utilizador”.

Nome	Criar Utilizador.
Breve Descrição	Este caso de uso visa descrever o processo de Criação de novos utilizadores.
Atores	Coordenador.
Trigger	n.a.
Pré-Condição	Login Válido.
Cenário de Sucesso	<ol style="list-style-type: none"> 1. O ator seleciona a opção “Criar Utilizador”; 2. O sistema apresenta uma tabela com os campos para preencher do novo utilizador (ID, nome, permissões, etc...); 3. No fim de estar preenchido, o ator grava a informação; 4. O sistema envia uma notificação “Novo Utilizador Criado com Sucesso”.
Fluxos Secundários	<ol style="list-style-type: none"> 3. a) Tentar gravar os dados sem um dos itens obrigatórios estar preenchido; 3. b) O ator opta por cancelar a ação.
Pós-Condição	n.a
Requisitos Especiais	n.a.

É através deste caso de uso que novos utilizadores, com diferentes níveis de permissões, irão ser criados.

Atribuir o caso a um Gestor de Caso

O Coordenador, depois de criar os novos utilizadores, poderá atribuir-lhe processos de pessoas sem-abrigo que foram sinalizadas. Após as pessoas em situação de sem-abrigo serem sinalizadas, o processo fica pendente da validação pelas equipas de emergência enviadas ao local. Assim que se confirme que é um caso válido, o processo poderá ser atribuído a um gestor de caso.

Tabela 9 - descrição de caso de uso “Atribuir o caso a um Gestor de Caso”

Nome	Atribuir o caso a um Gestor de Caso.
Breve Descrição	O ator “Coordenador” poderá atribuir os novos processos, resultantes das sinalizações realizadas, aos gestores de caso que o próprio criou no sistema.
Atores	Coordenador.
Trigger	n.a.
Pré-Condição	Login Válido.
Cenário de Sucesso	1. O ator seleciona a opção “Atribuir Processo a Gestor de Caso”; 2. O sistema apresenta uma lista com os utilizadores existentes a quem o processo poderá ser atribuído; 3. Escolher o gestor de caso e gravar; 4. O sistema envia a notificação “Processo “x” atribuído”.
Fluxos Secundários	2. a) Não aparecem utilizadores na lista; 3. a) O ator opta por cancelar a ação.
Pós-Condição	Sistema envia automaticamente mensagem e email ao gestor de caso a informar que lhe foi atribuído no sistema um novo caso.
Requisitos Especiais	n.a.

Este é o caso de uso que definirá que Gestor de Caso o caso de Pssa em questão. Esta decisão irá ligar o Gestor de Caso à Pssa, sendo o responsável por a acompanhar e ser a ligação entre esta e a organização.

Consultar os Processos

Depois dos processos serem criados, poderão ser acessados pelo coordenador e pelo respectivo gestor de caso. Na tabela seguinte iremos ver a descrição do processo desencadeado quando um destes dois atores decide fazer esta ação.

Tabela 10 – Descrição do caso de uso “Consultar os Processos”.

Nome	Consultar os Processos.
Breve Descrição	O ator “Coordenador” e os atores “Gestor de Caso” poderão consultar os processos existentes (no caso destes últimos, apenas os processos que lhes estão atribuídos).
Atores	Coordenador e Gestor de Caso.
Trigger	n.a.
Pré-Condição	Login Válido.
Cenário de Sucesso	1. O ator seleciona a opção “Consultar Processos”; 2. O sistema apresenta uma lista com os processos existentes; 3. Escolher o processo pretendido e consultar.
Fluxos Secundários	3. a) Não ter permissão para consultar o processo escolhido; 3. b) O ator opta por cancelar a ação.
Pós-Condição	n.a.
Requisitos Especiais	n.a..

Com este caso de uso irá ser possível ao Coordenador consultar os processos que pretender, e aos Gestores de Caso consultar os processos de Pssa que têm atribuídas.

Editar Processo

Depois de estar criado o processo, o gestor de caso terá a necessidade de o ir editando e atualizando, seja a inserir nova informação sobre a pessoa intervencionado, seja a atualizar a sua ficha de avaliação diagnóstico, seja a atualizar o plano de intervenção. Na tabela seguinte iremos ver a descrição deste processo.

Tabela 11 - Descrição do caso de uso “Editar Processo”.

Nome	Editar Processo.
Breve Descrição	O ator “Gestor de Caso” irá ter que ir atualizando os seus processos, acrescentando novas informações e alterando antigas informações que possam estar erradas.
Atores	Gestor de Caso.
Trigger	n.a.
Pré-Condição	Login Válido.
Cenário de Sucesso	<ol style="list-style-type: none"> 1. O ator seleciona a opção “Editar processo”; 2. O sistema apresenta uma lista dos processos existentes; 3. O ator escolhe o processo pretendido; 4. O sistema apresenta três tabelas: <ul style="list-style-type: none"> • Ficha de Identificação (o que é preenchido na opção “criar novo processo”); • Ficha de Avaliação Diagnóstico; • Plano de Intervenção; 5. Escolher a tabela a editar; 6. Fazer as alterações pretendidas e gravar; 7. O sistema apresenta a mensagem “Gravado com sucesso”.
Fluxos Secundários	<ol style="list-style-type: none"> 2. a) Tentar aceder a um processo ao qual não tem acesso; 4. a) Tentar preencher o plano de intervenção sem existir ficha de avaliação diagnóstico para aquele processo; 6. a) Tentar gravar o processo sem um dos itens obrigatórios estar preenchido; 6. b) O ator opta por cancelar a ação.
Pós-Condição	n.a.
Requisitos Especiais	n.a.

Com este caso de uso irá ser possível ao Gestor de Caso editar ou adicionar informação aos processos de Pssa que lhe estão atribuídas. Toda a informação recolhida pelo Gestor de Caso após a criação do processo individual pela Equipa de Emergência será adicionada nesta opção.

Criar Ficha de Avaliação Diagnóstico

O Gestor de Caso, depois de entrar em contacto com a Pssa que lhe foi atribuída, elabora uma Ficha de Avaliação Diagnóstico onde faz um diagnóstico aprofundado da pessoa. Na tabela seguinte iremos ver a descrição deste processo.

Tabela 12 - Descrição do caso de uso “Criar Ficha de Avaliação Diagnóstico”.

Nome	Criar Ficha de Avaliação Diagnóstico.
Breve Descrição	O “Gestor de Caso”, após ter contactado com a Pssa, cria uma ficha de avaliação para essa pessoa com o diagnóstico da sua situação.
Atores	Gestor de Caso.
Trigger	n.a.
Pré-Condição	Login Válido.
Cenário de Sucesso	1. O ator seleciona a opção “Criar Ficha de Avaliação Diagnóstico”; 2. O sistema apresenta várias tabelas com os dados para preencher; 3. Preencher os dados e gravar; 4. O sistema mostra a notificação “Ficha de Avaliação Diagnóstico Criada com Sucesso”.
Fluxos Secundários	3. a) Tentar gravar os dados sem um dos itens obrigatórios estar preenchido; 3. b) O ator opta por cancelar a ação.
Pós-Condição	Sistema envia automaticamente mensagem e email ao gestor de caso a informar que foi criada uma ficha de avaliação diagnóstico para o processo x.
Requisitos Especiais	n.a.

A Ficha de Avaliação diagnóstico será a ferramenta que permitirá ao Gestor de Caso diagnosticar as necessidades de Pssa, influenciando as ações a serem tomadas no futuro relativas a esta pessoa.

Criar Plano de Intervenção

O passo seguinte depois do Gestor de Caso fazer a ficha de avaliação diagnóstico da pessoa, será criar um plano de intervenção para a mesma, onde define as medidas que se tentará tomar para melhorar a vida da pessoa. Na tabela seguinte iremos ver a descrição deste processo.

Tabela 13 - Descrição do caso de uso “Criar Plano de Intervenção”.

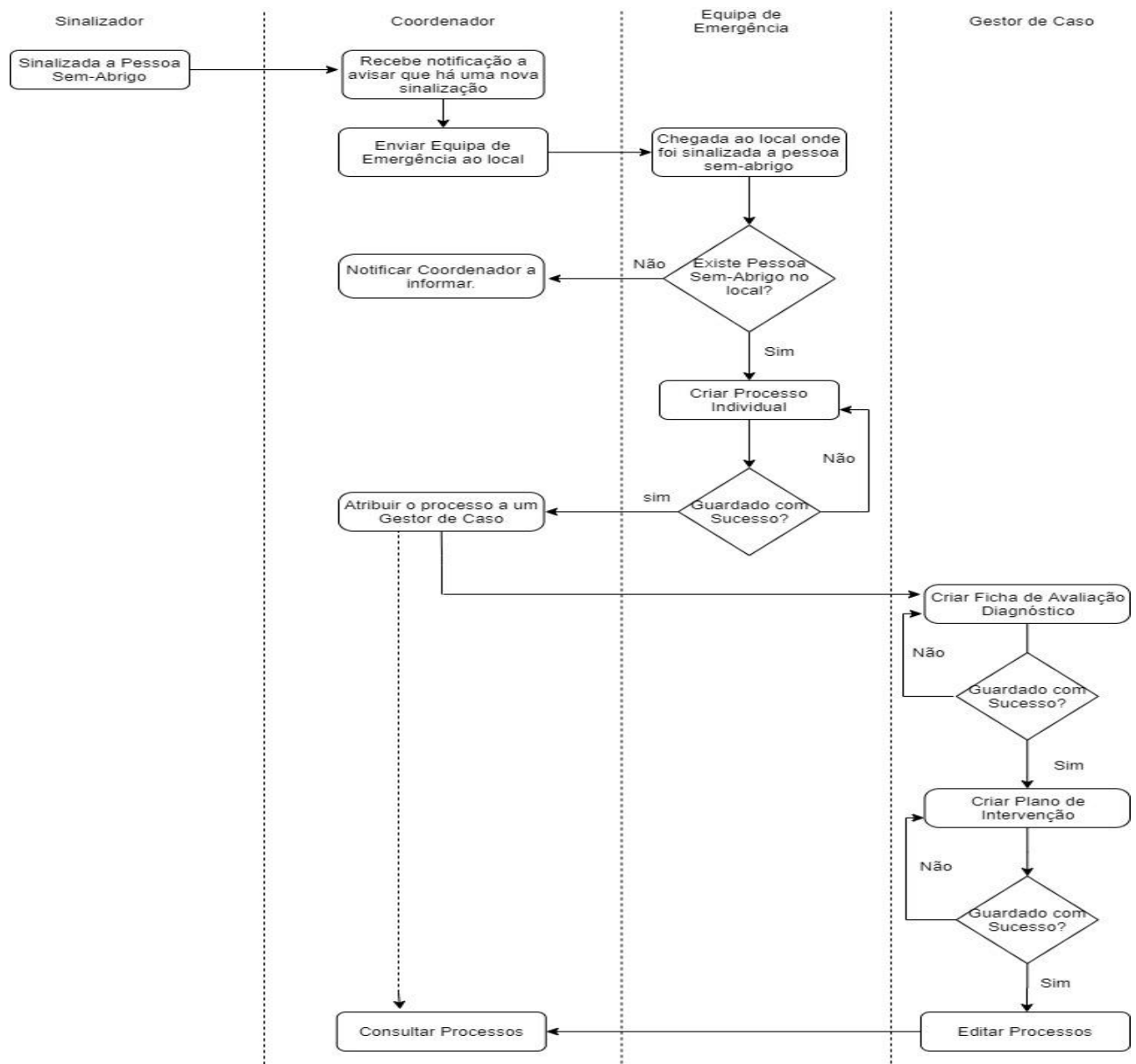
Nome	Criar Plano de Intervenção.
Breve Descrição	O “Gestor de Caso”, após ter realizado o diagnóstico à pessoa a interencionar e assim levantado as suas necessidades, cria um plano de intervenção para essa pessoa.
Atores	Gestor de Caso.
Trigger	n.a.
Pré-Condição	Login Válido.
Cenário de Sucesso	1. O ator seleciona a opção “Criar Plano de Intervenção”; 2. O sistema apresenta várias tabelas com os dados para preencher; 3. Preencher os dados e gravar; 4. O sistema mostra uma notificação “Plano de Intervenção criado com Sucesso”.
Fluxos Secundários	3. a) Tentar gravar os dados sem um dos itens obrigatórios estar preenchido; 3. b) O ator opta por cancelar a ação.
Pós-Condição	Sistema envia automaticamente mensagem e email ao gestor de caso a informar que foi criado um plano de intervenção para o processo “x”.
Requisitos Especiais	n.a.

O Plano de Intervenção deverá ser feito com objetivos concretos e atingíveis, e deverá ser possível acompanhar quais destes objetivos foram atingidos com sucesso e quando o foram.

4.6 Diagrama de Atividades

No diagrama de atividades são representados os diferentes processos pelo qual uma atividade passa até ser realizada. Os diagramas de atividades são destinados a mostrar as atividades que compõem um processo de sistema e o fluxo de controle de uma atividade para outra (Sommerville, 2019).

Figura 10 – Diagrama de Atividades.



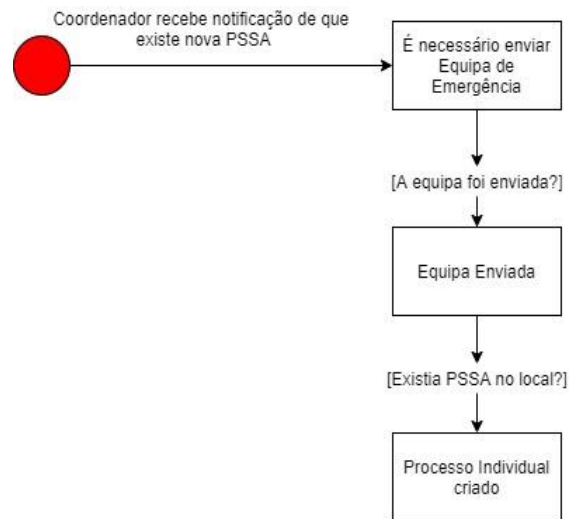
A figura 10 apresenta os intervenientes do processo (Sinalizador, Coordenador, Equipa de Emergência, Gestor de Caso) no sentido de caracterizar as diferentes interações existentes entre os intervenientes no processo.

4.7. Diagrama de Estados

Segundo Booch et al. (2012), um diagrama de estados é usado para mostrar a dinâmica de um sistema, uma vez que dão ênfase ao comportamento de um objeto, quando há interação entre ele e eventos. Este diagrama mostra uma máquina de estados, que é constituída por estados, transições, eventos e atividades.

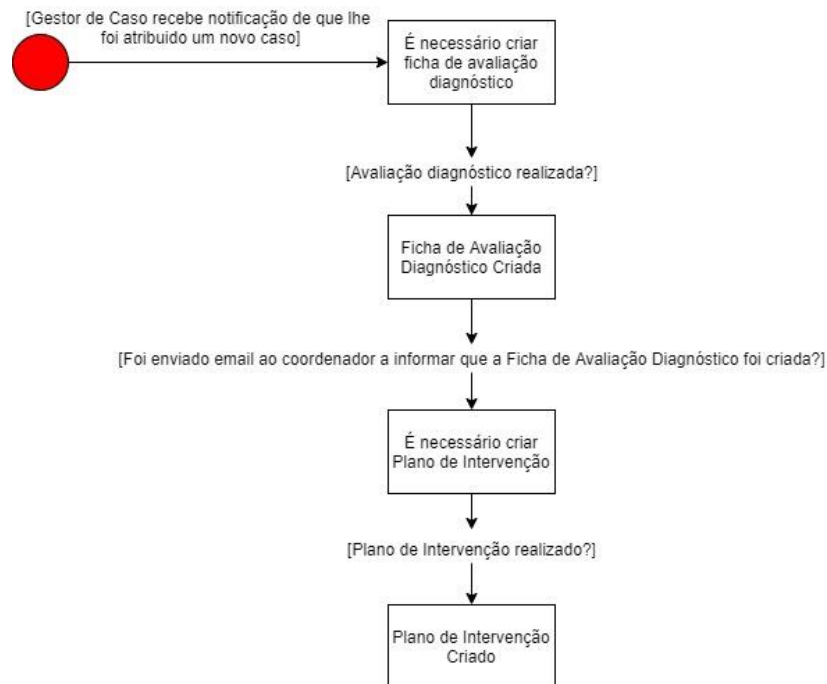
De seguida apresentar-se-ão dois exemplos de diagramas de estados, um para a criação do processo individual, outro para a criação da Ficha de Avaliação Diagnóstico e do Plano de Intervenção.

Figura 11 – Diagrama de Estados do Processo Individual



O coordenador ao receber a notificação de que existe uma nova Pssa, o Identificador Único (ID) dessa Pssa transita para o estado “É necessário enviar Equipa de Emergência”. Quando a equipa é enviada, o ID passa para o estado “Equipa enviada”. Depois da Equipa de Emergência chegar ao local e estabelecer contacto com a Pssa, quando criar o processo individual, o ID passa para “Processo Individual Criado”.

Figura 12 – Diagrama de Estados da Ficha de Avaliação Diagnóstico e do Plano de Intervenção



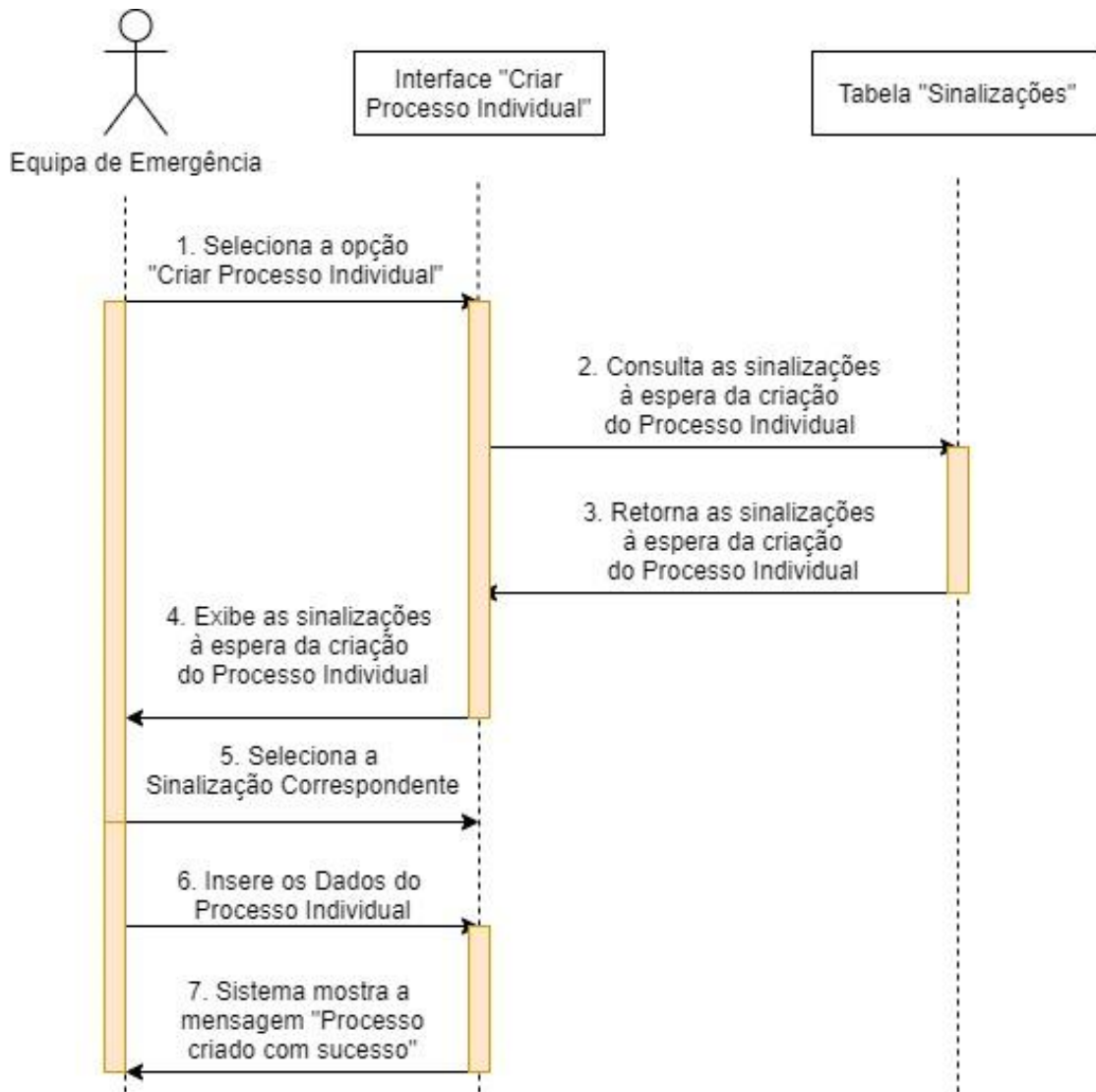
O gestor de caso ao receber a notificação de que lhe foi entregue um novo caso de Pssa, o ID dessa Pssa transita para o estado “É necessário criar ficha de avaliação diagnóstico”. Quando essa ficha é realizada, o ID passa para o estado “Ficha de Avaliação Diagnóstico criada”. Depois do Gestor de Caso criar a Ficha de Avaliação Diagnóstico, o sistema automaticamente envia um email e uma notificação de sistema ao Coordenador a informar, e aí o ID passa para “É necessário criar plano de intervenção”. Quando o Plano de Intervenção é criado, o ID passa para “Plano de Intervenção Criado”.

4.8. Diagrama de Sequência

Segundo Booch et al. (2012), um diagrama de sequência é construído a partir do caso de uso, e descreve interações que existem entre os objetos, num determinado comportamento, em determinado período de tempo. Ele exhibe as interações entre os objetos e as mensagens que são geradas entre eles num caso de uso. Este diagrama é, portanto, uma ferramenta que é usada para representar as interações que surgem entre os objetos de um cenário, ordenadas temporalmente, o que permite representar essa informação de maneira mais clara e sequenciada.

As figuras seguintes representam o diagrama de sequência da criação de processo individual, criação de ficha de avaliação diagnóstica, criação de plano de intervenção e consultar processo.

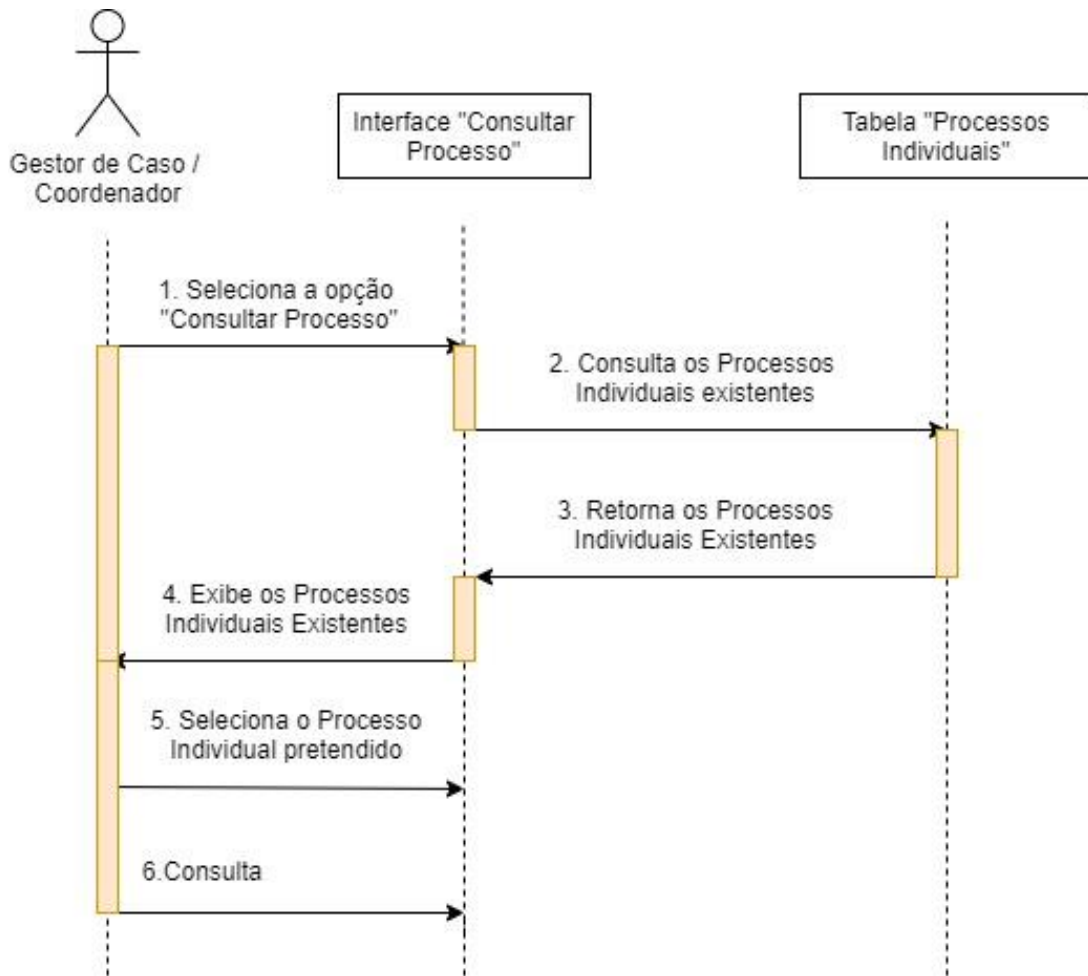
Figura 13 – Diagrama de Sequência “Criar Processo Individual”



Quando a sinalização de uma nova Pssa é feita, esse registo fica no sistema a aguardar que seja criado um Processo Individual para ele. Após a Equipa de Emergência se deslocar ao local e confirmar a existência de Pssa sinalizada, tem de seleccionar esse registo e criar um Processo Individual com os dados pessoais que recolheu junto do individuo sinalizado.

Após o processo individual ser criado pela equipa de emergência, ele passa a estar disponível para consulta pelo coordenador, e após ser atribuído por este a um gestor de caso, também poderá ser consultado pelo respetivo gestor de caso. Assim, o ator que quer consultar o processo terá que o seleccionar de entre os processos disponíveis (para o coordenador todos os processos individuais, para o gestor de caso apenas os processos que lhe estão atribuídos) e consultar.

Figura 14 - Diagrama de Sequência “Consultar Processo”

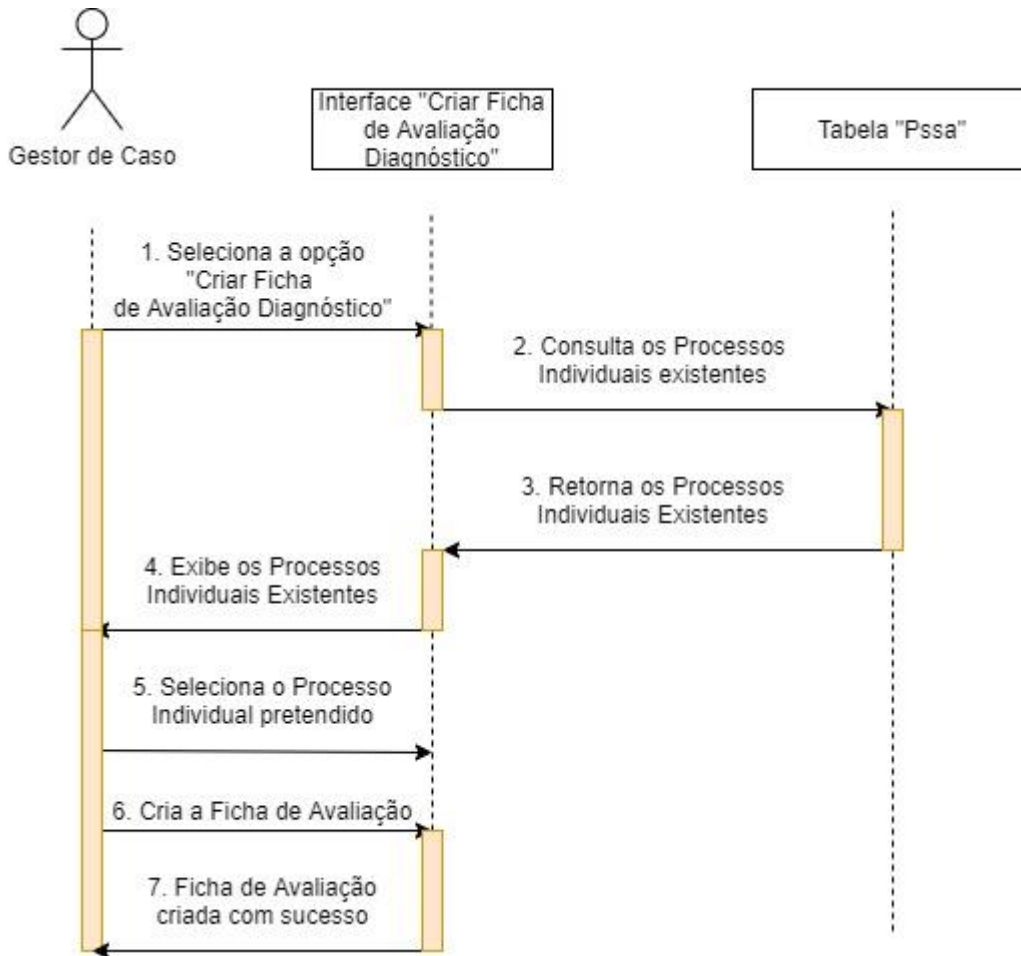


Neste diagrama de sequência o coordenador/gestor de caso, através da opção “consultar processo”, acede a toda a informação contida na base de dados relativa à Pssa pretendida e a todo o seu processo.

Após o processo individual ser criado pela equipa de emergência, passa a ser possível adicionar-lhe uma ficha de avaliação diagnóstico, realizada pelo gestor de caso depois de se encontrar com a Pssa. O gestor de caso irá à opção “criar ficha de avaliação diagnóstico”, irá seleccionar o processo individual para o qual quer criar a ficha e quando terminar, guarda.

Esta ficha será composta por diversos parâmetros de diagnóstico de Pssa, que serão complementares à informação contida na tabela “Pssa”. Será assim criada a tabela “Ficha de Avaliação diagnóstico”.

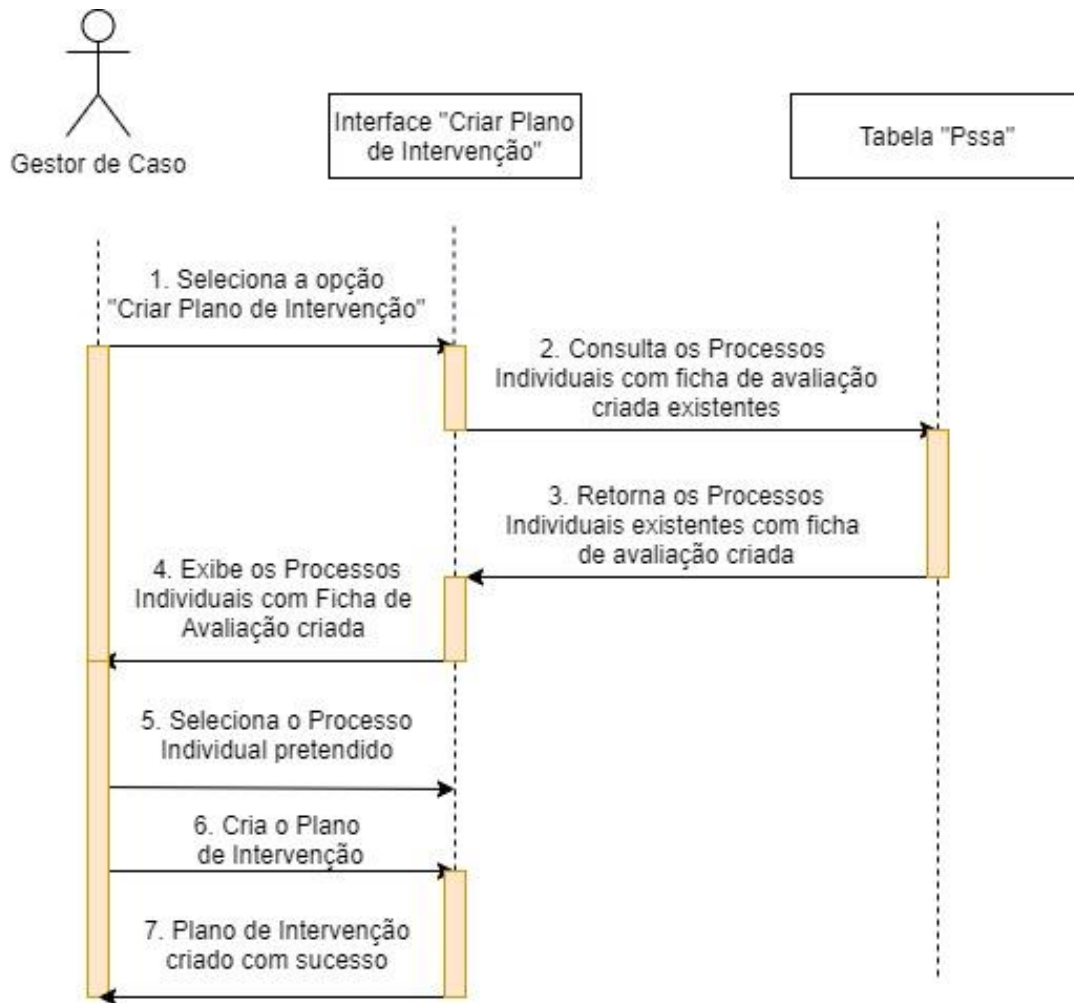
Figura 15 – Diagrama de Sequência “Criar Ficha de Avaliação Diagnóstico”



Neste diagrama de sequência o coordenador/gestor de caso, através da opção “Criar Ficha de Avaliação Diagnóstico”, acede à informação contida na entidade “Pssa” da base de dados. No final de criada a ficha de avaliação diagnóstico com sucesso, é criada na base de dados uma nova entrada na entidade “Ficha de Avaliação Diagnóstico”, que está ligada a um e só um processo individual específico, ou seja, uma Pssa, pois cada ficha de avaliação diagnóstico apenas poderá estar ligada a uma Pssa.

Depois da ficha de avaliação diagnóstico ser criada pelo gestor de caso, passa a ser possível adicionar ao processo individual o plano de intervenção, realizado pelo gestor de caso após chegar a acordo com a Pssa. O gestor de caso irá à opção “Criar Plano de Intervenção”, irá seleccionar o processo individual para o qual quer criar o plano de intervenção, guarda.

Figura 16 – Diagrama de Sequência “Criar Plano de Intervenção”

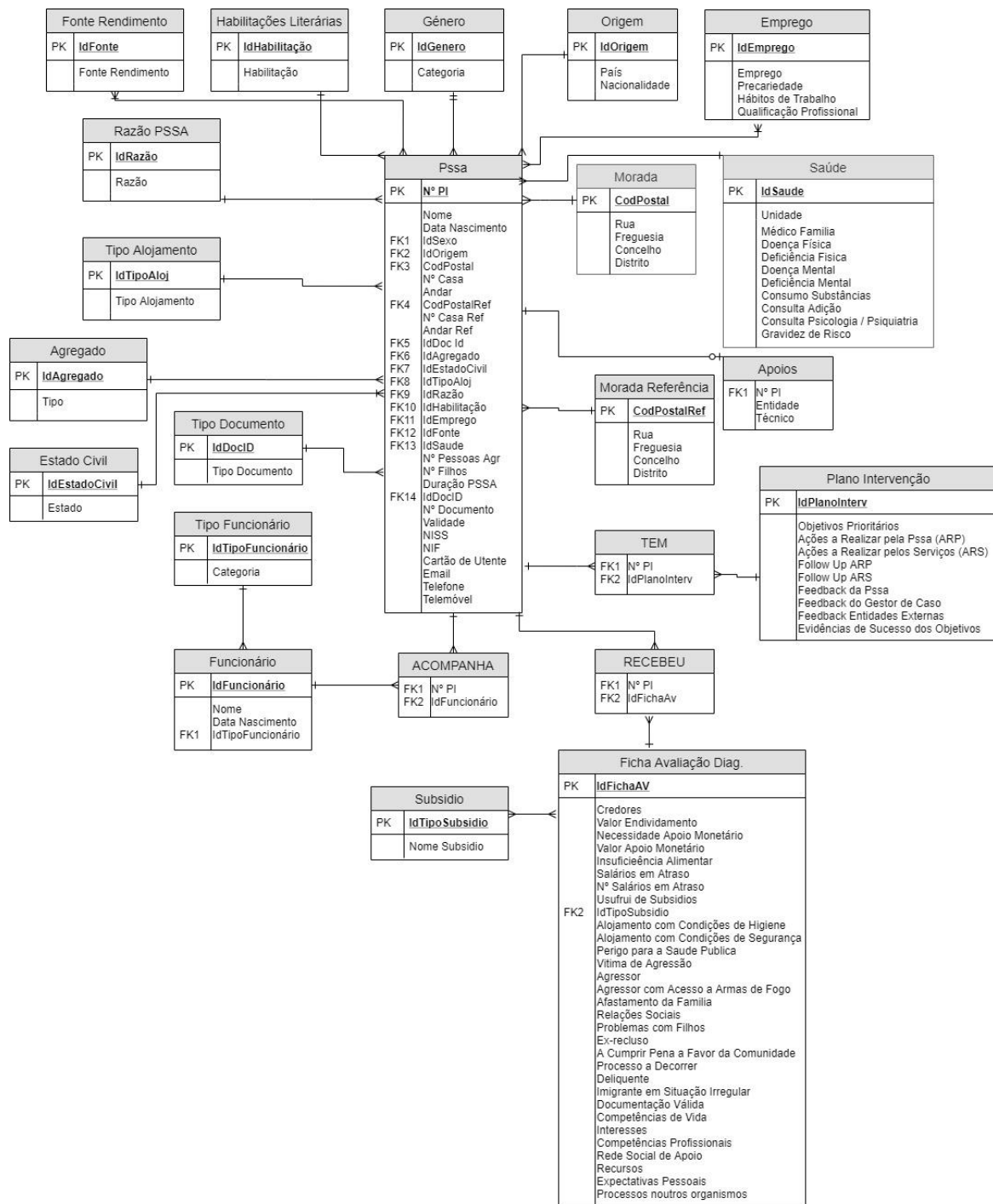


Neste diagrama de sequência o coordenador/gestor de caso, através da opção “Criar Plano de Intervenção”, acede à informação contida nas entidades “Pssa” e “Ficha de Avaliação Diagnóstica” da base de dados. No final de criado o plano de intervenção com sucesso, é criada na base de dados uma nova entrada na entidade “Plano de Intervenção”, que está ligada a um e só um processo individual específico, ou seja, uma Pssa, pois cada plano de intervenção apenas poderá estar ligado a uma Pssa.

4.9. Diagrama Entidade - Relação

Segundo Pressman & Maxim (2019), um diagrama de Entidade - Relação representa todos os objetos que são inseridos, armazenados, transformados e produzidos dentro de uma aplicação. No modelo relacional desenvolvido para a problemática de Pssa, a entidade “Pssa” é aquela com mais relações e à volta da qual irá ser feito o modelo.

Figura 17 – Diagrama Entidade - Relação



Para Pressman (2019), classe é um conceito que agrega dados e procedimentos que descrevem o conteúdo e o comportamento de entidades do mundo real, representadas por objetos. Estas classes são representadas por um retângulo (tabela), que contém dentro dele diferentes atributos. Estes atributos são os valores que serão inseridos na base de dados e podem além disso ser chave primária (pk) de uma tabela (é um atributo único que permite a associação de tabelas e que não pode ter valor nulo) e chave estrangeira (fk) (quando um dos atributos da tabela é chave primária de outra tabela). As relações entre as diferentes classes estão representadas na figura 17. A base de dados que será construída terá como principal fonte de dados a entidade “Pssa”, onde serão inseridos todos os dados relativos à Pssa, seja pela Equipa de Emergência, seja pelo Gestor de Caso. Cada Pssa terá relacionado uma ficha de avaliação diagnóstico e um plano de intervenção, que serão apenas personalizados para cada uma.

5. Análise comparativa de plataformas

Para implementar o sistema desenhado, foram avaliadas algumas plataformas de *low-code*.

Segundo Waszkowski (2019), plataformas de programação *low-code* são um conjunto de ferramentas para programadores e não programadores, que permitem a criação rápida de aplicações empresariais, com um esforço mínimo para escrever numa linguagem de código, requerendo um esforço reduzido para instalar e configurar no ambiente de trabalho, assim como para formar os utilizadores e implementar o programa. Estas plataformas baseiam-se na interface do utilizador para desenhar a aplicação, por oposição às técnicas de programação tradicionais. Estas plataformas focam-se no desenvolvimento da base de dados, dos processos de negócio e da interface do utilizador (Waszkowski, 2019).

Tabela 14 – Comparação entre as diferentes plataformas de *low-code*

	Mendix	Powerapps	Outsystem
Preço	Gratuito	8,40\$ a 33,70\$ por mês	6,5 \$ a 15\$ por mês
Linguagens suportadas	- Inglês	- Inglês, Chinês, Alemão, Japonês, Espanhol, Russo, Italiano, Português, Polaco	- Inglês
Disponível para	- Windows - Linux - Android - Iphone/Ipad	- Windows - Android - Iphone/Ipad - Mac	- Windows - Linux - Android - Iphone/Ipad - Mac
Integrações	- GitHub (sistema que permite aos utilizadores acompanharem as mudanças do código) - Microsoft Excel - Tableau Software (ferramenta na área de BI)	- SQL do Azure - Dropbox - One Drive - DB2- Base de dados da IBM. - PowerBI	- SAP - Google Drive - DropBox - Amazon

Dentre as plataformas de *low-code* existentes, foram analisadas:

- *Microsoft Powerapps,*
- *Outsystems*
- *Mendix.*

A tabela 14 sistematiza uma análise comparativa entre as diferentes plataformas *low-code*.

Nesta tabela é efetuada a comparação entre as três plataformas, sendo que a primeira característica analisada foi a característica determinante para escolher a plataforma *Mendix*. Uma vez que este é um projeto académico de desenvolvimento de uma aplicação para uma ONG, o facto de não haver custos associados à criação da aplicação é uma condição que tem de ser cumprida, pelo que o *Mendix* é a única que satisfaz esta condição por ser grátis. Nos outros parâmetros, a linguagem suportada é apenas o inglês, que não é algo problemático. Está disponível para quatro sistemas operativos (SO) (*Windows, Linux, Android, Iphone/Ipad*), perdendo neste campo apenas para a *Outsystems*, que está disponível para todos estes sistemas mais *Mac*, mas novamente, não é algo problemático, uma vez que os SO usados pela *Cáritas* englobam-se nestes quatro.

A nível de integração esta plataforma é bem mais limitada que as rivais, mas adequa-se ao pretendido, uma vez que é possível ser integrada com o *Microsoft Excel*, ferramenta usada pela associação atualmente para armazenar a informação existente sobre Pssa. Depois de escolhida esta opção, verificou-se que outra das vantagens que tem é a existência de uma academia de formação, oferecida pela plataforma, o que é bastante positivo para aprender a programar neste formato, e como pontos negativos verificou-se alguns bugs e a pouca documentação de apoio existente fora do website da *Mendix*.

A criação de uma aplicação na plataforma *Mendix* segue cinco passos:

1. Verificar os requisitos do negocio (*Capture*);
2. Criar um plano para a aplicação (*Plan*);
3. Desenvolver a aplicação (*Develop*);
4. Fazer *deploy* da aplicação (*Deploy*);
5. Fazer testes (*Review*).

Figura 18 – Ciclo de Vida de uma aplicação no *Mendix*



Fonte: Ciclo de Vida de uma aplicação no *Mendix*, (*Mendix*, 2018)

Devido ao tempo de execução do mestrado e às condições pandémicas que se têm vivido desde o início do ano transato, a criação da aplicação não foi exequível, tendo como principais fatores limitadores um background de formação na área das ciências socioeconómicas e reduzida experiência em programação. A elaboração desta aplicação tornou-se complexa, contudo foram encetadas um conjunto de tentativas e foi analisada a possibilidade de desenvolver a aplicação em *Outsystems*, tentativa essa que se revelou sem sucesso, tendo-se analisado a possibilidade de desenvolver o projeto com o *Mendix*. Face a essa possibilidade foi desenvolvido um modelo de dados, que servirá de base à conceção da aplicação. O desenvolvimento da aplicação nesta plataforma revelou-se complicada, em grande parte devido à dificuldade de adaptar a base de dados imaginada para o que é permitido e suportado pela plataforma. Não foi possível reproduzir a base de dados idealizada devido à impossibilidade de definir as pk e fk nesta plataforma, tendo esta se revelado demasiado simples para o modelo de dados idealizado.

6. Conclusões e perspectivas de trabalho futuro

Após o trabalho desenvolvido neste projeto, é possível fazer uma análise com as conclusões que se podem retirar, assim como trabalho a desenvolver no futuro, no seguimento do que foi abordado neste projeto.

6.1. Conclusões

Este projeto teve como objetivo estudar a problemática de Pssa, resultando no desenvolvimento dum protótipo não funcional que permitisse a agregação da informação recolhida junto de Pssa, permitindo lançar as bases para futuramente se desenvolver uma plataforma que contenha a informação recolhida junto de Pssa e que permita utilizar esta informação no terreno e atualizar a mesma online. O número de Pssa no nosso país tem tido um grande aumento nos últimos anos, com um relatório da OCDE a indicar que apenas entre 2014 e 2018 o número de Pssa em Portugal aumentou em 157% (Porfírio, 2020). Se a tendência nos anos anteriores já era muito negativa, com o surgimento da pandemia de Covid-19 no ano de 2020 e os problemas económicos e sociais que daí advieram, é expectável que a situação das pessoas em situação mais frágil na nossa sociedade seja afetada negativamente, o que pode fazer com que este fenómeno aumente. Face a este cenário, e à importância de se mitigar este fenómeno para que seja possível construir uma sociedade mais justa e com melhores condições de vida para todos, este projeto revela-se bastante oportuno. Olhando para a crescente digitalização do mundo, e os desenvolvimentos existentes na área da análise de dados e gestão de informação nos últimos anos, será pertinente otimizar o SI da organização. Sendo uma organização que trabalha diretamente com Pssa e cuja missão é ajudá-las a melhorar as suas condições de vida, os dados que recolhe junto dessas pessoas são dos ativos mais importantes que tem para conseguir realizar a sua missão com sucesso, o que torna o SI uma parte fulcral para concretizar essa missão com sucesso. É no SI que são armazenados todos os dados referentes às Pssa: quem são, de onde vêm, o porquê de estarem nesta situação, onde trabalham ou trabalharam, se têm antecedentes de doença ou vício e muitos outros que permitem aos trabalhadores da Cáritas compreender melhor a pessoa em questão e as suas necessidades. Um SI eficaz, que ajude a tratar esses dados e a oferecer informação relevante aos trabalhadores é por isso uma ferramenta que vai aumentar a eficácia das suas ações junto de Pssa.

O objetivo do trabalho foi o estudo da problemática, tendo depois sido feita a análise de requisitos e o desenvolvimento do protótipo. Dado o volume de informação e o trabalho

subjacente à temática para o projeto desenvolveu-se a fase da análise de requisitos. O projeto ficou nesta fase devido à abundância de material para desenvolver nas fases iniciais do projeto e, por outro lado, a incapacidade de conseguir desenvolver o artefacto com a plataforma de *low-code Mendix*, aliada ao pouco conhecimento técnico na área e escassez de tempo para terminar tudo dentro dos prazos existentes. Entre as mais valias que se podem retirar deste estudo para a organização, podem-se identificar a relativa simplicidade do software a desenvolver, que permitirá não só baixos custos no seu desenvolvimento como também baixa necessidade de formação para os utilizadores, a redução da utilização de papel e o fim da necessidade de existir um arquivo físico nesse formato, a possibilidade do gestor de caso aceder a uma grande quantidade de informação remotamente e a funcionalidade de qualquer pessoa da comunidade poder sinalizar uma Pssa existente na zona de maneira fácil e rápida.

6.2. Perspetivas de trabalho futuro

Uma vez que neste projeto se desenvolveu um protótipo não funcional para servir de base ao desenvolvimento futuro de uma solução que contenha uma base de dados com a informação recolhida junto de Pssa, assim como a sua versão móvel, fica aberta a possibilidade de desenvolver e implementar o projeto. Para concretizar este projeto deverá ser definida toda a parte tecnológica necessária para o projeto funcionar (hardware, software e telecomunicações envolvidas), assim como a maneira como a informação será gerada, recolhida e armazenada. Considera-se que o desenvolvimento da solução deverá ser suportada numa plataforma *low-code* ou criando uma base de dados, por exemplo em *SQL* e ligada a uma aplicação web desenvolvida em *JavaScript*. Para essa aplicação será necessário desenvolver ainda o sistema de permissões e autenticações de utilizadores, assim como a definição de diferentes perfis para os mesmos, que não foi abordado neste projeto. As instituições de solidariedade social são pela sua natureza instituições com inúmeras carências financeiras e ao nível de recursos humanos, assim, sugere-se uma análise de viabilidade da implementação de soluções que contemplem fatores de sustentabilidade e exequibilidade ao longo do tempo.

Referências

- Aguilar, J., Tripp-Barba, C., Misra, S., & Zamudio, L. (2017). A Requirements Engineering Techniques Review in Agile Software Development Methods. *Computational Science and Its Applications – ICCSA 2017* (pp. 683-698). Springer. doi:https://doi.org/10.1007/978-3-319-62404-4_50
- Albertão, S. E. (2005). *ERP - Sistemas de Gestão Empresarial: metodologia para avaliação, seleção e implantação*. Iglu.
- Anderson, I. (Março de 2003). Synthesizing Homelessness Research: Trends, Lessons and Prospects. *Journal of Community & Applied Social Psychology*, 3(2), pp. 197-205. doi:10.1002/casp.721
- Araújo, M. H. (2019). *Sistemas de Informação*. Senac.
- Aurum, A., & Wohlin, C. (2005). *Engineering and Managing Software Requirements*. Springer.
- Baptista, I., & Knutagård, M. (2019). *HOUSING FIRST IN EUROPE An Overview of Implementation*. Obtido de Strategy and Fidelity, Nicholas Pleace: www.enipssa.pt/documents/10180/12068/Sumário+executivo+-+Estudo+Europeu+sobre+o+modelo+Housing+Firs
- Basden, A. (2017). *The Foundations of Information Systems - Resarch and Practice*. Routledge.
- Baskerville, R., Baiyere, A., Gregor, S., Hevner, A., & Rossi, M. (2018). Design Science Research Contributions: Finding a Balance between Artifact and Theory. *Journal of the Association for Information Systems*, 19(5), 358-376.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2012). *UML: guia do usuário* (2ª ed.). Elsevier.
- Bourgeois, D. T., Smith, J. L., Wang, S., & Mortati, J. (2019). *Information Systems for Business and Beyond*. Open Textbooks. Obtido de <https://digitalcommons.biola.edu/open-textbooks/1>
- Caldas Jr., J., & Masiero, P. C. (1999). Obtenção de Modelos de Análise a partir de um Modelo de Requisito Baseado em Cenários. *Memorias de Jornadas Jornadas Iberoamericanas de Ingeniería de Requisitos y Ambientes de Software - IDEAS'99. 1*, pp. 24-35. Centro de Información Tecnológica.
- Castells, M. (2012). *A Sociedade em Rede* (4ª ed., Vol. 1). Fundação Calouste Gulbenkian.
- Chen, S.-P. (2020). *Fundamentals of Information and Communication Technologies*. Cambridge Scholars Publishing.

- Cheng, B. H., & Atlee, J. M. (2007). Research Directions in Requirements Engineering. Future of Software. *FOSE '07: 2007 Future of Software Engineering* (pp. 285-303). IEEE Computer Society.
- Christel, M., & Kang, K. (1992). *Issues in requirements elicitation*. Software Engineering Institute. Carnegie Mellon University. doi:[http://doi.org/10.1016/S0140-6736\(08\)60207-X](http://doi.org/10.1016/S0140-6736(08)60207-X)
- Cockburn, A. (2000). *Writing Effective Use Cases*. Addison-Wesley Longman.
- Coimbra, R. (18 de Maio de 2020). *Agile Methods: LSD/Agile/Projetos e TI*. Obtido em 4 de Janeiro de 2021, de Projetos e TI: <https://projetoseti.com.br/agile-methods-bsd/>
- Côrtes, P. L. (2012). *Administração de Sistemas de Informação*. Saraiva.
- Costa, A. B. (1998). *Cadernos Democráticos - Exclusões Sociais*. Gradiva Publicações.
- Davies, A. (2005). *Just enough requirements management: Where Software Development Meets Marketing*. Dorset House Publishing.
- Engholm Júnior, H. (2010). *Engenharia de software na prática*. Novatec.
- Estratégia Nacional para a Integração das Pessoas em Situação de Sem-Abrigo/Conceito de Pessoa em Situação de Sem-Abrigo*. (2017). Obtido em 26 de Janeiro de 2020, de Site oficial ENIPSSA: <http://www.enipssa.pt/conceito-de-pessoa-em-situacao-de-sem-abrigo>
- Estratégia Nacional para a Integração das Pessoas em Situação de Sem-Abrigo/Enquadramento*. (2017). Obtido em 26 de Janeiro de 2020, de Site Oficial ENIPSSA: <http://www.enipssa.pt/enquadramento>
- Estratégia Nacional para a Integração das Pessoas em Situação de Sem-Abrigo/NPISA*. (2017). Obtido em 26 de Janeiro de 2020, de Site oficial ENIPSSA: <http://www.enipssa.pt/npisa>
- Estratégia Nacional para a Integração das Pessoas em Situação de Sem-Abrigo/Sobre a ENIPSSA*. (2017). Obtido em 09 de Abril de 2020, de Site oficial ENIPSSA: <http://www.enipssa.pt/sobre-a-enipssa>
- Estratégia Nacional para a Integração de Pessoas Sem Abrigo 2009 – 2015 (ENIPSA)*. (s.d.). Obtido em 26 de Janeiro de 2020, de Site Oficial da Segurança Social: http://www.seg-social.pt/documents/10152/13334/enipsa_2009_2015
- Ferguson, P., Humphrey, W. S., Khajenoori, S., Macke, S., & Matvya, A. (Maio de 1997). Results of applying the Personal Software Process. *IEEE Computer*, 30(5), pp. 24-31. doi:10.1109/2.589907

- Fernandes, J. M., & Machado, R. J. (2017). *Requisitos em projetos de software e de sistemas de informação*. Novatec.
- Fernandes, M. M. (2006). *Fechados no Silêncio – Os Sem Abrigo*. Tese de Mestrado, Universidade Aberta. Obtido de <http://hdl.handle.net/10400.2/619>
- Fortin, M. F. (2009). *Fundamentos e Etapas do Processo de Investigação*. Lusodidacta.
- Hevner, A., March, S., Park, J., & Ram, S. (Março de 2004). Design science in information systems research. *MIS Quarterly*, 28(1), pp. 75-105.
- IEEE. (1997). *IEEE Software Standards Collection*. IEEE Computer Society Press.
- ISO/IEC 27002. (2013). *Information Technology - SecurityTechniques - Information Security Management Systems - Requirements*. ISO.
- Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The Unified Software Development Process*. Addison-Wesley.
- Lacerda, D. P., Dresch, A., Proença, A., & Antunes Junior, J. A. (26 de Novembro de 2013). Design Science Research: método de pesquisa para a engenharia de produção. *Gestão e Produção*, 20(4), pp. 741-761. doi:<http://dx.doi.org/10.1590/S0104-530X2013005000014>
- Laudon, K. C., & Laudon, J. P. (2020). *Essentials of MIS* (14^a ed.). Pearson.
- Leffingwell, D., & Widrig, D. (2000). *Managing Software Requirements - A Unified Approach*. Addison-Wesley Longman.
- Lojkine, J. (1995). *A Revolução Informacional*. Cortez.
- Madureira, V. (2010). *Modelo de Gestão de Identidades e Acessos*. Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto. Obtido de <https://hdl.handle.net/10216/67888>
- March, S. T., & Smith, G. F. (Dezembro de 1995). Design and Natural Science Research on Information Technology. *Decision Support Systems*, 15(4), pp. 251-266. doi:[https://doi.org/10.1016/0167-9236\(94\)00041-2](https://doi.org/10.1016/0167-9236(94)00041-2)
- Marcuschi, L. (2006). *Análise da Conversação*. Ática.
- Mendix. (2020). *Evaluation guide/Mendix*. Obtido em 20 de Outubro de 2020, de Mendix: <https://www.mendix.com/evaluation-guide/mendix-difference>
- Mertens, D. M. (2019). *Research and Evaluation in Education and Psychology - Integrating Diversity With Quantitative, Qualitative, and Mixed Methods* (5^a ed.). SAGE Publications.

- Moresi, E. (2003). *Metodologia da Pesquisa*. Universidade Católica de Brasília. Obtido em 15 de Setembro de 2020, de <http://www.inf.ufes.br/~pdcosta/ensino/2010-2-metodologia-de-pesquisa/MetodologiaPesquisa-Moresi2003.pdf>
- O'Brien, J. A., & Marakas, G. M. (2012). *Administração de sistemas de informação - uma introdução* (15ª ed.). McGraw-Hill.
- Pandey, D., Suman, U., & Ramani, A. K. (2010). An Effective Requirement Engineering Process Model for Software Development and Requirements Management. *International Conference on Advances in Recent Technologies in Communication and Computing (ARTcom)* (pp. 287-291). IEEE Xplore. doi:10.1109/ARTCom.2010.24
- Parlamento Europeu, Conselho Europeu. (27 de abril de 2016). Regulamento (UE) 2016/679 do Parlamento Europeu e do Conselho, de 27 de abril de 2016, relativo à proteção das pessoas singulares no que diz respeito ao tratamento de dados pessoais e à livre circulação desses dados e que revoga a Diretiva 95/46/CE (RGPD). *Jornal Oficial da União Europeia*, L (119) 1-88. Obtido de Web site da União Europeia: <https://eur-lex.europa.eu/legal-content/PT/TXT/HTML/?uri=CELEX:32016R0679#d1e40-1-1>
- Pimenta, M. (1992). *Os Sem-Abrigo na cidade de Lisboa*. Editores Centro de Estudos para a Intervenção Social.
- Polloni, E. (2000). *Administrando Sistemas de Informação*. Futura.
- Poppendieck, M., & Poppendieck, T. (2013). *The Lean Mindset: Ask the Right Questions*. Addison-Wesley.
- Porfírio, J. (2020). Em quatro anos, número de sem-abrigo em Portugal sobe 157%. *Observador*. Obtido em 21 de outubro de 2020, de <https://observador.pt/2020/01/29/em-quatro-anos-numero-de-sem-abrigo-em-portugal-sobe-157/>
- Portugal. (2 de Abril de 1976). *Constituição da República Portuguesa*. Obtido em 31 de Janeiro de 2020, de Portal do Parlamento: https://dre.pt/web/guest/legislacao-consolidada/-/lc/337/202002091333/exportPdf/normal/1/cacheLevelPage?_LegislacaoConsolidada_WAR_drefrontofficeportlet_rp=indice
- Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner's Approach* (9ª ed.). McGraw-Hill.
- Quatrani, T. (23 de Junho de 2003). *Introduction to the unified modeling language*. Obtido em 5 de Novembro de 2020, de http://www.nyu.edu/classes/jcf/g22.2440-001_sp09/handouts/intro_rdn.pdf

- Rainer, R. K., Turban, E., & Potter, R. E. (2005). *Administração De Tecnologia Da Informação*. Rio de Janeiro: Campus.
- Rainer, R. K., Turban, E., & Potter, R. E. (2016). *Introduction to Information Systems – Supporting and Transforming Business* (4ª ed.). John Wiley.
- Schwartz, M., & Machulak, M. (2018). *Securing the Perimeter: Deploying Identity and Access Management with Free Open Source Software*. Apress.
- Software, R. (Novembro de 2001). *Rational Unified Process - Best Practices for Software Development Teams*. Obtido em 6 de Abril de 2020, de IBM: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
- Sommerville, I. (2019). *Engenharia de Software* (10ª ed.). Pearson Brasil.
- Stallings, W., & Brown, L. (2018). *Computer Security: Principles and Practice* (4ª ed.). Pearson.
- Stephens, M., & Rosenberg, D. (2003). *Extreme Programming Refactored*. Apress.
- Vaishnavi, V., Kuechler, W., & Petter, S. (30 de Junho de 2019). *Design Science Research in Information Systems*. (V. Vaishnavi, W. Kuechler, & S. Petter, Edits.) Obtido em 30 de Março de 2020, de www.desrist.org: <http://www.desrist.org/design-research-in-information-systems>
- Vazquez, C., & Simões, G. (2016). *Engenharia de Requisitos: Software Orientado ao Negócio*. Brasport.
- Waszkowski, R. (Janeiro de 2019). Low-code platform for automating business processes in manufacturing. *IFAC Papers Online*, pp. 376-381. doi:10.1016/j.ifacol.2019.10.060
- Whitman, M. E., & Mattord, H. J. (2017). *Principles of Information Security* (6ª ed.). Cengage Learning.
- Wirfs-Brock, R. (1993). Designing Scenarios. *Smaltalk Report*, Vol.3, No. 3.
- Zwass, V. (1997). *Foundations of Information Systems*. McGraw-Hill.