



Instituto Superior de Gestão e Administração de Santarém

Curso de Mestrado em Engenharia De Tecnologias e Sistemas WEB

Staffing Tool

David Matias - 22006270

Orientador:

Professor Doutor Fernando Bento

Unidade Curricular: Dissertação

Santarém

Ano letivo 2021-2022

AGRADECIMENTOS

Na conclusão de mais uma grande etapa da minha vida, quero agradecer aos meus pais, Joaquim Matias e Lúdia Matias. Não há palavras suficientes para expressar o apoio incondicional que me deram. Esta etapa não é só minha, mas sim deles também, com muito orgulho. Nunca esquecerei o sacrifício e tudo o que abdicaram para que eu possa ser o que sou hoje e, sobretudo, por me terem acompanhado em todo este percurso até ao fim.

Agradeço à empresa *Do It Lean* e a todos os colegas de trabalho que me apoiaram, prestando toda a informação e apoio necessário para do desenvolvimento deste tema. Quero também agradecer à minha colega Mónica Furtado e ao meu grande amigo Pedro Vicente pelo apoio ao longo de todo o processo, não só pelo apoio prestado, mas também pelas sugestões e ideias partilhadas.

Dirijo um especial agradecimento ao Professor Doutor Fernando Bento pela paciência e profissionalismo com que me orientou, mostrando-se sempre disponível para me auxiliar na realização do presente documento.

Por fim, mas não menos importante, quero também agradecer a todos os docentes, funcionários e colegas do ISLA Santarém e ISLA Gaia, que em muito contribuíram para o enriquecimento do meu conhecimento, no decorrer do percurso académico.

Obrigado a todos por todo o apoio, atenção e companheirismo que demonstraram.

LISTA DE ACRÓNIMOS

API - *Application Programming Interface*

DSR - *Design Science Research*

RAD - *Rapid Application Development*

RH - Recursos Humanos

SI - Sistemas de Informação

ST - Staffing Tool

TI - Tecnologias de Informação

UI - *User Interface*

UML - *Unified Modeling Language*

XP - *Extreme Programming*

RESUMO

Cada vez mais, em ambientes organizacionais, os sistemas de informação são os instrumentos mais importantes para a sua sobrevivência no mercado, tanto em termos de crescimento como de gestão interna, desde os seus clientes até aos seus próprios recursos humanos. Este documento baseia-se na gestão de recursos humanos do estado da arte, incluindo a disponibilidade destes recursos no cliente, de forma a gerir a alocação e disponibilidade de recursos humanos para integrar em projeto. Foi feita uma pesquisa a nível de estado da arte, onde são referenciados os problemas que existem na gestão de alocação de recursos humanos nas organizações e de que forma é que esses recursos poderiam ser utilizados e geridos. Com base neste estudo, foi projetado um artefacto que pode resolver este problema das organizações, utilizando ferramentas de engenharia de *software* para desenhar toda a estrutura da solução, utilizando metodologias ágeis para a organização da mesma. Para a sua implementação foi utilizada uma tecnologia de *low-code* para desenvolver a solução orientada para organizações de grande gestão de recursos humanos. Além do desenho e da correspondente implementação da solução, este documento apresenta também cenários de solução, ilustrando assim a sua utilização e situações reais de forma a poder aproximar a realidade à organização. Conclui-se que este estudo cria valor através da investigação realizada pelo estado da arte, dando uma vista ao leitor dos pontos essenciais para a produção desta solução apresentada neste documento.

Palavras-chave: Gestão de Recursos Humanos; Alocação de Recursos; Low-Code, Desenvolvimento; OutSystems.

ABSTRACT

Increasingly, in organizational environments, information systems are the most important instruments for their survival in the market, both in terms of growth and internal management, from their customers to their own human resources. This document is based on state-of-the-art human resources management, including the availability of these resources at the client, to manage the allocation and availability of human resources to integrate into a project. State-of-the-art research was carried out, where the problems that exist in the management of the leasing of human resources in organizations are referenced and how these resources could be used and managed. Based on this study, an artifact was designed that can solve this problem of organizations, using software engineering tools to design the entire structure of the solution, using agile methodologies for the organization of the solution, and for its implementation, a low-code technology was used to develop the solution-oriented toward large human resource management organizations. In addition to the design and corresponding implementation of the solution, this document also presents solution scenarios, thus illustrating its use and real situations to bring reality closer to the organization. It is concluded that this study creates value through the investigation carried out by the state of the art, giving the reader a view of the essential points to produce this solution presented in this document.

Keywords: Human Resources Management; Resource Allocation; Low-Code, Development; OutSystems.

ÍNDICE GERAL

AGRADECIMENTOS.....	III
LISTA DE ACRÓNIMOS.....	IV
RESUMO.....	V
ABSTRACT.....	VI
ÍNDICE GERAL.....	VII
ÍNDICE DE TABELAS.....	X
ÍNDICE DE FIGURAS.....	XI
INTRODUÇÃO.....	13
ENQUADRAMENTO.....	13
QUESTÃO E OBJETIVOS DE INVESTIGAÇÃO.....	14
OBJETIVOS.....	14
PLANEAMENTO.....	16
ORGANIZAÇÃO DA DISSERTAÇÃO.....	17
ESTADO DA ARTE.....	19
RECURSOS HUMANOS.....	19
ALOCÇÃO DE RECURSOS HUMANOS.....	20
SISTEMAS DE INFORMAÇÃO.....	21
METODOLOGIAS ÁGEIS.....	22
SCRUM.....	23
ENGENHARIA DE SOFTWARE.....	25
MODELAÇÃO UML.....	27
ARMAZENAMENTO DA INFORMAÇÃO.....	27
LOW-CODE.....	27
NO-CODE.....	31
DIFERENÇA ENTRE LOW-CODE E NO-CODE.....	32
OUTSYSTEMS.....	33
ARCHITECTURE CANVAS.....	35
ABORDAGEM METODOLÓGICA.....	36

DESIGN SCIENCE RESEARCH.....	36
IMPLEMENTAÇÃO.....	39
REQUISITOS FUNCIONAIS	39
<i>Dashboard</i>	39
<i>Disponibilidade de recursos</i>	39
<i>Alocações de Recursos</i>	40
<i>Equipa de Recursos Humanos no Projeto</i>	40
CASOS DE USO.....	41
DIAGRAMA DE CLASSES.....	42
MODELO ER	43
ARQUITETURA	43
WIREFRAMES.....	45
DESENVOLVIMENTO DA SOLUÇÃO.....	45
<i>Criar um novo projeto</i>	45
<i>Criar perfis da equipa associados ao projeto no Staffing Tool</i>	46
<i>Selecionar recurso humano a um projeto</i>	46
<i>Criar um pedido de recursos “Request to HR” num determinado projeto com vários perfis</i>	47
<i>Consultar e submeter o pedido aos Recursos Humanos</i>	49
<i>Listagem de pedidos</i>	51
<i>Detalhe dos pedidos</i>	51
<i>Estado dos pedidos</i>	52
<i>Cancelamento dos pedidos</i>	52
<i>Cancelamento de pedidos por resolução interna</i>	53
<i>Notificação do cancelamento por resolução interna</i>	53
<i>Cancelamento de pedidos por perfil</i>	55
<i>Notificação do cancelamento por perfil</i>	56
<i>Criar candidaturas outsourcing</i>	56
<i>Acesso aos dados dos candidatos</i>	57
<i>Candidatos associados por projeto</i>	57

<i>Estados dos pedidos</i>	58
<i>Resumo dos pedidos</i>	59
CONCLUSÕES	61
BIBLIOGRAFIA	62
APÊNDICES	67
APÊNDICE I – WIREFRAMES DA APLICAÇÃO	67

ÍNDICE DE TABELAS

Tabela 1 Planeamento	16
Tabela 2 Low-code VS Tradicional (Adaptado)	30
Tabela 3 Comparação entre low-code e no-code (Adaptado)	32

ÍNDICE DE FIGURAS

Figura 1 AS-IS	15
Figura 2 TO-BE	16
Figura 3 - Metodologia Scrum	24
Figura 4 - Ciclo Scrum	25
Figura 5 Líderes do mercado de plataformas low-code	29
Figura 6 Arquitetura da plataforma OutSystems	34
Figura 7 Camadas da "Architecture Canvas"	35
Figura 8 Estrutura de investigação de sistemas de informação (Hevner et al., 2004)	37
Figura 9 Metodologia DSR (Hevner, 2007).....	38
Figura 10 Diagrama de Casos de Uso	41
Figura 11 Diagrama de Classes.....	42
Figura 12 Modelo de ER.....	43
Figura 13 Arquitetura do projeto.....	44
Figura 14 <i>Architecture Canvas</i> do projeto	44
Figura 15 Formulário para criar novo projeto	46
Figura 16 Editar recurso no projeto	47
Figura 17 Selecionar um recurso.....	47
Figura 18	48
Figura 19 Formulário para fazer pedidos de recursos ao RH.....	49
Figura 20 Listagem de pedidos para enviar aos RH	50
Figura 21 Email de pedido de recurso ao RH	50
Figura 22 Listagem de pedidos ao RH agrupado por projeto	51
Figura 23 Detalhe de um projeto e os seus respetivos pedidos.....	52
Figura 24 Listagem dos pedidos ao RH através do <i>Staffing Tool</i>	52

Figura 25 <i>Resolve Internal for Developer</i>	53
Figura 26 Listagem após atualizar o estado do pedido	53
Figura 27 Email a notificar que o pedido de recurso ao RH foi resolvido internamente....	54
Figura 28 Detalhe do pedido <i>Resolved Internal</i>	55
Figura 29 <i>Cancel for Developer</i>	55
Figura 30 Email a notificar que o pedido de recurso ao RH foi cancelado.....	56
Figura 31 Associar candidato do recrutamento ao projeto.....	57
Figura 32 Atualização da listagem após associar o recurso ao projeto no RH	57
Figura 33 Dashboard do Staffing Tool.....	58
Figura 34 Listagem de pedidos ao Outsourcing.....	58
Figura 35 Listagem de pedidos ao RH para o projeto “Demo Script 5”	59
Figura 36 Dashboard do Staffing Tool.....	59
Figura 37 <i>Wireframe</i> do projeto	67
Figura 38 <i>Wireframe</i> da equipa do projeto	68
Figura 39 <i>Wireframe</i> recurso do projeto	69
Figura 40 <i>Wireframe</i> para adicionar um recurso ao projeto	70
Figura 41 <i>Wireframe</i> da listagem dos pedidos RH	71
Figura 42 <i>Wireframe</i> do formulário de pedido aos RH.....	72
Figura 43 <i>Wireframe</i> do formulário para adicionar novo projeto	73
Figura 44 <i>Wireframe</i> da pesquisa de recursos para o projeto	74
Figura 45 <i>Wireframe</i> listagem dos pedidos ao RH	75
Figura 46 <i>Wireframe</i> do detalhe do recurso humano	76
Figura 47 <i>Wireframe</i> da pesquisa de recursos.....	77
Figura 48 <i>Wireframe</i> do dashboard.....	78
Figura 49 <i>Wireframe</i> da listagem dos projetos	78

Figura 50 <i>Wireframe</i> detalhe de um recurso humano e sua alocação	79
Figura 51 <i>Wireframe</i> do resumo do projeto	79
Figura 52 <i>Wireframe</i> do calendário no alocação dos recursos em projeto	80
Figura 53 <i>Wireframe</i> da alocação de recursos em projeto através do calendário	80

INTRODUÇÃO

O primeiro capítulo desta dissertação contempla o contexto e o enquadramento da temática do projeto realizado, onde é evidenciado a sua pertinência e importância, juntamente com as questões e objetivos de investigação.

Enquadramento

Na era digital em que vivemos, a organização e visualização dos dados que nos permitem quantificar ou avaliar o nosso comportamento num dado espectro torna-se cada vez mais relevante e recorrente.

Como a natureza do desenvolvimento de *software* é um processo centrado nas pessoas e nas suas interações, os fatores humanos são considerados como tendo grande relevância no desenvolvimento de software. Isto significa que as atividades realizadas pelas partes interessadas, dependendo da forma como são realizadas, podem afetar o produto final de diferentes formas. (Yanzer Cabral et al., 2014)

As gestões dos recursos humanos indicam diferentes aspetos do ser humano, que estão envolvidos e têm impacto no desenvolvimento de software. Estes fatores podem ser abordados em três categorias:

- Individual: Abrange questões humanas individuais relacionadas com a engenharia ou desenvolvimento de software, tais como características individuais, personalidade, cultura, entre outras.
- Interpessoal: Está relacionado com fatores humanos entre indivíduos que afetam ou são afetados por processos de engenharia ou desenvolvimento de software, tais como cooperação, aprendizagem em grupo, trabalho em equipa, entre outros.
- Organizacional: Inclui fatores humanos relacionados com organizações e ambientes de trabalho, tais como tomada de decisão em organizações, consultores, ambiente organizacional e outros.

Na atribuição de recursos humanos, algumas decisões têm de ser tomadas pelo gestor do projeto, entre elas, o gestor deve decidir a quem atribuir cada atividade (Arias et al., 2018; Chang et al., 2001; Zavyalova et al., 2020).

Para isso, é necessário determinar que competências e formação académica são necessárias para o desempenho de cada atividade, para olhar na organização que são os profissionais que têm a qualificação necessária e que estão disponíveis neste momento e, finalmente, atribuir o melhor possível aos profissionais para as atividades a realizar (Chiang & Lin, 2020; Kieling et al., 2021; Nestel et al., 2016).

Questão e objetivos de investigação

Dado o problema de investigação enunciado no ponto anterior e tendo em conta a solução que se pretende obter, surgem as seguintes questões de investigação:

- Que tipo de funcionalidades uma aplicação digital deve conter, para que seja possível gerir, analisar e monitorizar a alocação dos recursos humanos em projeto?
- Como é que vou saber se a equipa do projeto está completa?
- Como é que vou saber o estado de cada projeto, consoante o estado da alocação de cada recurso humano?
- Consultar a disponibilidade de alocação de cada recurso humano?
- Como é que vou consultar alocação de cada recurso humano?
- De que forma vou pedir recursos humanos e como os posso adquirir recorrendo ao outsourcing?
- Como é que vou consultar os pedidos na área dos recursos humanos por projeto?

Desta forma, de modo que se consiga responder às questões de investigação apresentadas, o objetivo geral inerente às mesmas define-se por: desenvolver uma plataforma digital (aplicação) que permita o registo, análise e controlo do processo de alocação dos recursos humanos e controlar os pedidos ao departamento de recursos humanos.

Objetivos

O presente documento de dissertação do mestrado em engenharia tecnológica e sistemas web consiste na criação de uma dissertação relacionada com o desenvolvimento de uma aplicação empresarial. Os principais objetivos da aplicação são ajudar no processo de atribuição de recursos a projetos, tornando-o mais rápido, simples e eficaz.

Esta aplicação será implementada numa grande empresa nos arredores de Lisboa, onde surgiu a necessidade de implementar um processo eficaz na alocação de recursos humanos aos projetos.

Uma vez que é uma aplicação essencial para o funcionamento diário da empresa, devido à sua grande importância torna-se imprescindível esta solução.

Com base nos principais objetivos, que são o desenvolvimento de uma aplicação utilizando boas práticas de programação, utilizando arquiteturas e quadros de desenvolvimento, com os dados existentes na empresa de forma a alimentar este novo SI.

O desenvolvimento da aplicação será realizado respeitando boas práticas de engenharia de software, identificando todos os requisitos, criando diagramas de alto e baixo nível, o próprio desenvolvimento, implementação e a sua manutenção.

O principal objetivo do trabalho final no mestrado é a realização de uma aplicação nomeada por *Staffing Tool*. O *Staffing Tool* é uma ferramenta que ajudará no processo de atribuição de recursos humanos aos projetos.

Pretende-se, com esta plataforma, passar do modelo presente na **Figura 1** para o modelo representado na **Figura 2**.



Figura 1 AS-IS

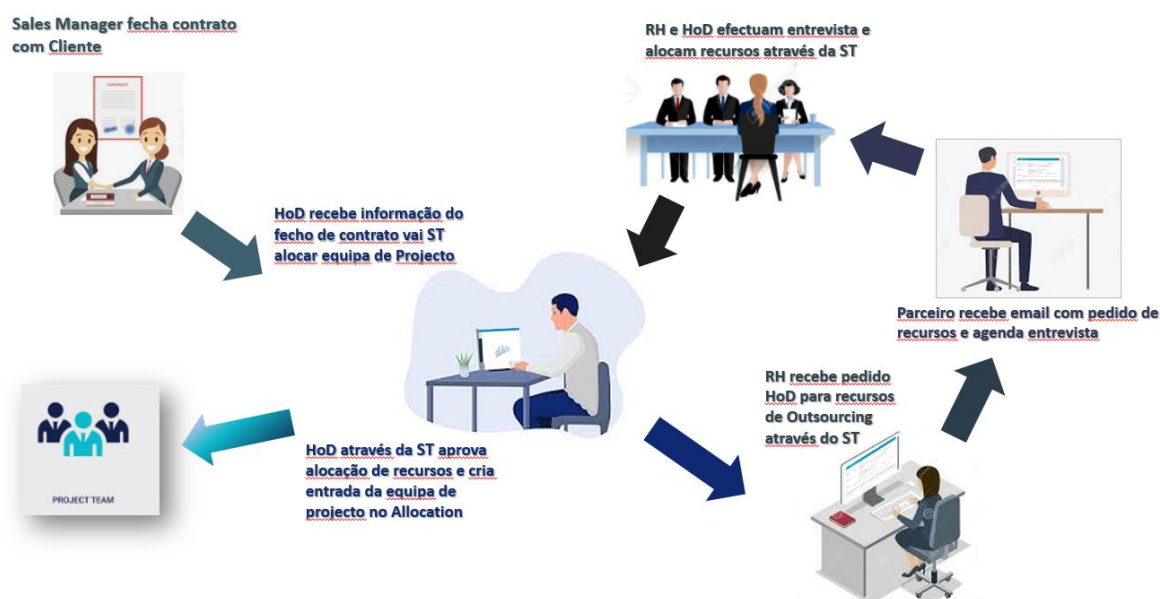


Figura 2 TO-BE

Assim, permite a colaboração das partes interessadas nos processos de criação de equipas de projetos na mesma base de informação e com integração com outras fontes de informação importantes para a tomada de decisões.

Planeamento

O planeamento para a produção da dissertação pode ser consultado na **Tabela 1**

Tabela 1 Planeamento

Tarefas	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Julh
Estado da Arte									
Req. funcionais									
Casos de Uso									
Modelo ER									
Diagrama de Classes									
Wireframes									
Arquitetura									
Implementação									
Defesa									

Organização da Dissertação

O presente documento está separado em capítulos que relatam todo o trabalho realizado para a produção da dissertação referente ao mestrado em engenharia de tecnologias e sistemas web.

O capítulo **Estado da Arte** contém uma análise das metodologias e tecnologias utilizadas do mais geral ao mais específico. Começando por uma abordagem dos recursos humanos, de forma a apresentar a sua importância no mundo empresarial, depois a alocação de recursos humanos no projeto, de forma a poder caracterizar e definir a melhor abordagem para sua gestão.

De seguida, uma análise de engenharia de *software*, os seus objetivos, processos e modelos. Com a apresentação das fases de engenharia de *software*, é feita a apresentação das abordagens à gestão de projeto, com um foco maior na abordagem *Agile*.

Apresentação da tecnologia UML, que é a base de todas as modelações utilizadas ao longo deste documento, sendo também utilizada para modelação pelos autores que fundamentaram este capítulo. Em seguida, uma breve apresentação sobre o armazenamento de informação, uma abordagem ao *low-code* e *no-code*, apresentando cada uma delas e referindo a sua diferença.

Por fim, a apresentação de algumas tecnologias e ferramentas que complementam o desenvolvimento dos sistemas de informação, nomeadamente a tecnologia *OutSystems* e a definição da sua arquitetura como plataforma.

O capítulo **Abordagem Metodológica** apresenta e explica a metodologia de investigação *Design Science Research*, adotada para a produção do estado da arte deste documento.

O capítulo da **Implementação** apresenta o trabalho realizado dividindo em vários subcapítulos:

- Requisitos Funcionais
- Caso de Uso
- Diagrama de Classes
- Modelo ER
- Arquitetura
- *Wireframes*
- Desenvolvimento da Solução

No capítulo **Conclusões**, existe numa reflexão durante e após a conclusão da investigação realizada durante a dissertação, expectativas das tecnologias, metodologias e a implementação da solução.

No capítulo **Bibliografia**, é apresentada a lista de todas as fontes de conhecimento e as informações utilizadas para atingir os objetivos propostos.

No capítulo **Apêndices**, está compilada toda a documentação produzida que serve de apoio a este documento.

ESTADO DA ARTE

Antes de iniciar o desenvolvimento de uma solução, um dos principais aspetos para garantir o seu sucesso e evitar a dúvida técnica no futuro é desenhar uma boa arquitetura que, juntamente com a modularização da aplicação, permite desenvolver novas funcionalidades e manter o código de forma mais fácil e rápida.

Recursos Humanos

A premissa da utilização da metodologia *Scrum* em RH é, precisamente, trazer mais agilidade aos processos, como foi dito na introdução deste conteúdo, o tempo é um dos recursos que as empresas não se podem dar ao luxo de desperdiçar em tempos de transformação digital e aumento da competitividade no espectro mais diverso do mercado (Kavitha & Suresh, 2021; Lepak et al., 2006).

A melhor parte é que os requisitos são entregues mais rapidamente, a qualidade dos processos não é deixada de lado. Por outras palavras, quando são utilizadas técnicas eficientes os colaboradores podem desempenhar as suas funções em menos tempo, o que, consequentemente, aumenta a capacidade da equipa e proporciona crescimento ao sector como um todo, segundo (Huemann et al., 2007; X. Wang et al., 2012).

Utilizando a metodologia *Scrum* para gerar eficiência em RH e fornecer os resultados desejados, cada ação deve ser atribuída a um profissional devidamente treinado, bem como um prazo anexado a ele. Desta forma, cada membro da equipa será encarregue de cuidar de todos os detalhes durante a execução das atividades, bem como do progresso dos projetos de outros departamentos (Ichniowski et al., 1995; Yanzer Cabral et al., 2014).

Por último, mas não menos importante, é possível alinhar as expectativas da empresa com as dos profissionais de RH na fixação de objetivos para a equipa. Se considerarmos que um dos fundamentos do *Scrum* é a divisão do projeto em diferentes ciclos com uma determinada duração, os objetivos podem estar relacionados com cada *Sprint* (Britto et al., 2012; Lin, 2011).

Mas para que isto funcione na prática, cada objetivo deve ser estabelecido com base em critérios realistas como a capacidade de execução da equipa, os recursos disponíveis e o tempo médio de entrega. A fixação de prazos curtos para que um grande número de responsabilidades seja cumprido só fará com que os colaboradores se sintam sobrecarregados, o que pode ter impactos negativos no clima organizacional (Aslam & Ijaz, 2018; Huzooree & Ramdoo, 2015).

Alocação de Recursos Humanos

O problema da alocação de recursos humanos envolve um grupo de profissionais e um conjunto de atividades. Cada profissional tem muitas características:

- Competências;
- Conhecimento;
- Experiências;
- Formação acadêmica;
- Certificações.

Além disso os profissionais estão indisponíveis em diferentes períodos. Não estar disponível num período significa que o profissional não pode ser atribuído a qualquer atividade de qualquer projeto nesse período. (Nankervis et al., 2019)

Esta indisponibilidade pode ser devido a várias razões, tais como férias, alocação para outra atividade no mesmo período, entre outros. O custo das horas de trabalho e o número de horas de trabalho por dia por um profissional são também informações relevantes para o problema de atribuição, uma vez que esta informação influencia respetivamente, o custo do projeto e a disponibilidade do profissional para a missão (Mellentien & Trautmann, 2001).

Cada atividade por sua vez, tem um conjunto de características necessárias, isto são requisitos que um profissional precisa de ter para realizar a tarefa. E além disso, uma atividade indica o tempo que é necessário para ser realizado e quantas horas por dia precisam ser dedicado à sua execução nesse período (Engwall & Jerbrant, 2003).

Considerando que uma gestão dos recursos humanos em projeto de *software*, consiste basicamente na seleção e alocação dos mesmos de acordo com as competências, conhecimentos, formação académica e experiência necessárias para a realização de cada atividade em projeto. O seu principal objetivo é selecionar os profissionais mais adequados, além disso, a gestão, a monitorização e análise contínua, de forma a minimizar o impacto de desempenhos potencialmente negativos que estão associados ao sucesso do projeto como um todo (Jaselskis & Ashley, 1991).

O processo consiste nos seguintes métodos:

- Planeamento da Alocação de Recursos Humanos, que visa planejar a atribuição de profissionais necessários para executar o projeto de desenvolvimento de *software* em causa;
- Monitorização da Alocação de Recursos Humanos indica, gerir e monitorizar a alocação de recursos humanos que participam num projeto de desenvolvimento de *software*;
- Avaliação de Recursos Humanos, que visa avaliar os profissionais em relação às competências e conhecimentos, além de registar as experiências adquiridas durante a execução das suas atividades em projeto.

O conjunto de projetos é gerido de forma estruturada e os projetos devem ser entregues de acordo com os objetivos da organização (H. Wang et al., 2018).

Os auxílios da gestão dos recursos humanos nas organizações são utilizados sistemas de informação. Esses SI contêm várias funcionalidades da área do RH, como armazenamento de dados dos funcionários, gestão de vencimentos, processos de recrutamento, entre outros (Ankrah et al., 2012). A nível de mercado, existem vários Sistemas de informação para recursos humanos, essas soluções podem ser aplicações isoladas ou módulos que integram outros SI. Como por exemplo a *Primavera Software*¹, é um sistema de gestão interna empresarial (*ERP - Enterprise Resource Planning*), que tem um módulo específico para a gestão de recursos humanos, com a possibilidade de controlar as alocações. E outro exemplo é aplicação com o nome de *Monday.com*², que é uma solução isolada, onde também contempla inúmeras funcionalidades a gestão da área do RH.

Sistemas de Informação

Um sistema de informação é um padrão representativo do negócio, que demonstra de forma consciente, o conhecimento do funcionamento e dos seus objetivos.

A gestão de sistemas de informação, segundo (Varajão, 2005), é um conjunto de processos, que são essenciais no negócio para a gestão de sistemas de informação e a filiação de tecnologias a suportar.

¹ Disponível em <https://pt.primaverabss.com/pt/>

² Disponível em <https://monday.com>

A gestão de sistemas de informação é um processo contínuo e interativo, desenvolvido através de três atividades principais:

- Planeamento;
- Desenvolvimento;
- Exploração.

O Planeamento de sistemas de informação é a parte integrante geral do negócio, que identifica os objetivos e prioridades que devem guiar os planos às diferentes áreas.

O Desenvolvimento dos sistemas de informação refere-se a todas as atividades envolvidas de engenharia de *software*, para a construção de sistemas de informação que apoiem a organização. No decorrer da aplicação dos processos de engenharia de *software*, podem ser utilizados diferentes modelos de processos, técnicas e tecnologias.

A Exploração é responsável pela administração de TI, como também será preponderante para a definição de futuras estratégias, restringindo ou facilitando as outras atividades.

Metodologias Ágeis

A expressão “Metodologias Ágeis” tornou-se conhecida em 2001, quando especialistas em processos de desenvolvimento de *software* representado, os métodos Scrum e XP, fundiram princípios e características comuns destes métodos. Foi assim criada a “Aliança Ágil” e efetuou-se o estabelecimento do “Manifesto Ágil”. A partir do Manifesto Agile, que os métodos *Agile* se tornaram práticas recorrentes no desenvolvimento de *software* num ambiente de rápida evolução, existindo vários métodos ou *frameworks Agile* que têm muito em comum como as iterações curtas, retorno frequente e rápido dos clientes e aprendizagem contínua, sendo o *Scrum* um dos mais usados, segundo (X. Wang et al., 2012)

O “*Manifesto for Agile Software Development*” é composto por 4 valores essenciais no desenvolvimento de *software* (Beck et al., 2001):

- Pessoas e interação entre eles, isto é, mais do que processos e ferramentas;
- *Software* a funcionar, mais que documentação;
- Colaboração com o cliente, mais que negociação de contratos;
- Responder a mudanças, em vez seguir um plano fixo.

Esta metodologia assenta em 12 princípios fundamentais:

1. Prioridade em satisfazer o cliente através da entrega adiantada e contínua de *software*;
2. Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento;
3. Entregar *software* a funcionar com frequência, na escala de semanas ou até meses, com preferência em períodos mais curtos;
4. As pessoas relacionadas com os negócios e pessoas ligadas ao desenvolvimento devem trabalhar em conjunto e diariamente, durante todo o projeto;
5. Construir projetos ao redor de pessoas motivadas;
6. O método mais eficiente e eficaz de transmitir informações é através de uma conversa;
7. *Software* funcional é a medida primária de progresso;
8. Processos ágeis promovem um ambiente sustentável;
9. Contínua atenção à excelência técnica e bom *design*, aumenta a agilidade;
10. Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito;
11. As melhores arquiteturas, requisitos e *designs* emergem de equipas auto-organizáveis;
12. Em intervalos regulares, a equipa reflete em como ficar mais produtivo, ajusta-se e otimiza o comportamento de acordo com o que foi refletido.

Scrum

A metodologia *Scrum* ilustrada na **Figura 3** e na **Figura 4**, disponibiliza uma *framework* e um conjunto de *guidelines* para o desenvolvimento de um projeto. Este é realizado em ciclos (tipicamente com a duração de 2 semanas a 1 mês) designados por *Sprints*, em cada um dos quais existe um conjunto de atividades a serem cumpridas.

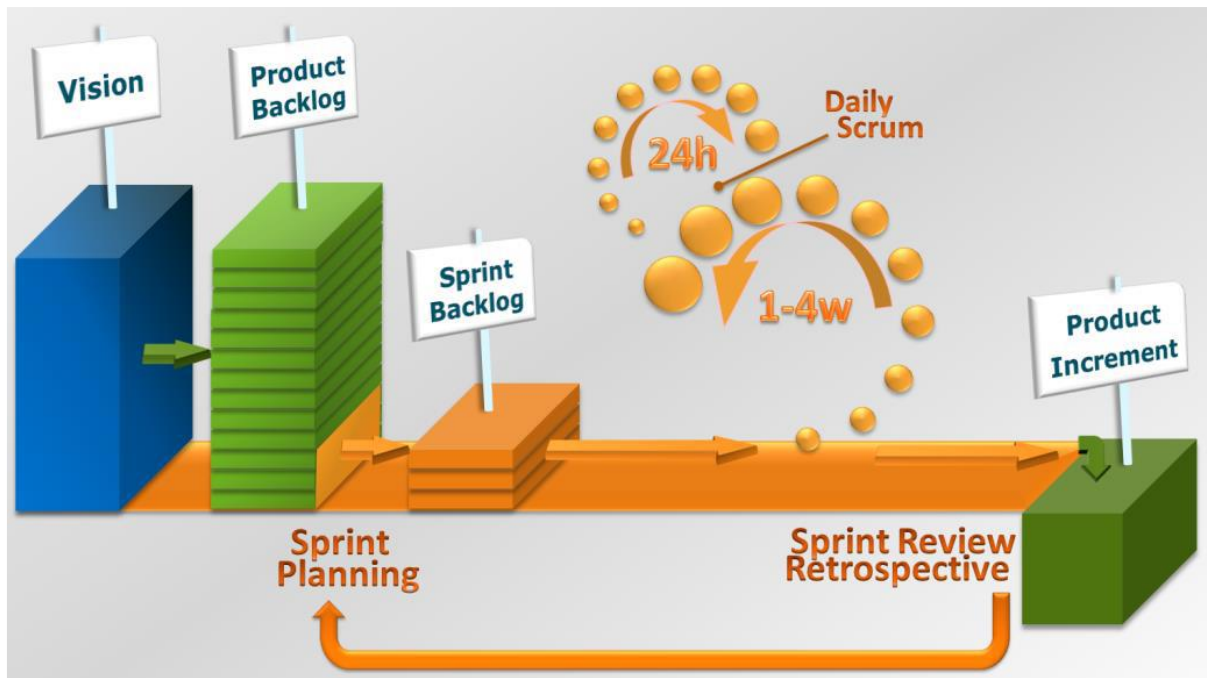


Figura 3 - Metodologia Scrum³

Em cada *Sprint* é desenvolvido um produto designado por *Product Increment*, o qual produz uma aplicação a funcionar, mas implementando apenas um subconjunto da totalidade de requisitos enumerados no *Product Backlog* na forma de *User Stories*. Em cada *Sprint*, vai sendo acrescentado valor ao projeto de forma incremental (adicionando requisitos), que junto com os testes e demonstrações (*Sprint Reviews*) no fim de cada *Sprint*, facilitam o entendimento e comunicação entre a equipa de desenvolvimento do projeto e o cliente final (Stellman & Greene, 2014).

³ Disponível em <https://www.capgemini.com/ch-en/2011/05/the-scrum-perspective-to-agile/>



Figura 4 - Ciclo Scrum

- ***Sprint Planning*** – O que vamos fazer a seguir?
- ***Scrum Daily*** – Como está a correr hoje?
- ***Sprint Review*** – Como é que fizemos?
- ***Sprint Retrospective*** – Como é que melhoramos?

Isto torna muito mais fácil identificar estrangulamentos, falhas ou pontos de melhoria antes que seja tarde demais e a operação seja afetada por impactos negativos, danos ou quaisquer outros problemas. (Soares, 2004)

Engenharia de Software

A engenharia de *software* insere-se na atividade de Desenvolvimento e passa por um conjunto de atividades e resultados que produzem um *software*, por vezes o próprio *hardware*. A engenharia abrange todas as atividades incluindo a gestão de projeto, desenvolvimento, teorias, metodologias e tecnologias para encontrar as soluções para os problemas, dentro das restrições organizacionais e financeiras.

Os processos de *software* organizam as atividades de engenharia de *software*. Os diferentes processos de *software* organizam essas atividades de formas distintas e são descritas em níveis de detalhes diferenciados, como sugere (Stephens, 2015).

No entanto, há algumas atividades básicas comuns à grande parte dos processos existentes (Stephens, 2015):

- Levantamento de requisitos;
- Desenho de alto nível;
- Desenho de baixo nível;
- Desenvolvimento;
- Testes;
- Implementação;
- Manutenção.

O Levantamento de Requisitos consiste na descrição de todas as propriedades e as restrições ou condicionantes da solução, indicando todos os métodos, funcionalidades e o aspeto da interface que o sistema deve ter.

O Desenho de Alto Nível fornece uma visão do sistema num nível abstrato, indicando quais as principais partes da aplicação final e reconhecendo as principais ações e os seus intervenientes.

O Desenho de Baixo Nível tem com objetivo a descrição detalhada de todas as funções, métodos, interfaces e também toda a informação que responde às necessidades do utilizador.

O Desenvolvimento é o coração de engenharia de *software*. Esta atividade é também chamada de construção ou codificação do *software*. A codificação enfatiza que o *software* deve ser construído utilizando linguagens de programação.

Ao realizar os Testes ao *software* é possível identificar erros a partir da execução do programa, de acordo com a recolha de requisitos.

A Implementação é um processo de instalação do *software* e também pelo método de formação aos seus futuros utilizadores.

A Manutenção é o método de melhoria e otimização de um *software* já desenvolvido e também reparação de defeitos. A Manutenção do *software* é uma das fases do processo de Desenvolvimento de *software*, que ocorre a seguir à Implementação de *software*.

Modelação UML

A abstração do sistema de *software* através de modelos que o descrevem é uma ferramenta poderosa para compreender e comunicar o produto final que será desenvolvido. Segundo (O'Docherty, 2005), UML é uma linguagem padrão para a modelação de sistemas de informação.

A UML é uma linguagem representada através de diagramas, cada um dos quais é composto por elementos que possuem relação entre si, de forma a padronizar e melhorar o controlo da engenharia de *software*.

Armazenamento da informação

Não há SI sem base de dados. Quando existe a necessidade de desenvolver de raiz uma estrutura para uma nova base de dados, existem várias alternativas metodológicas. Uma das hipóteses aponta para os seguintes passos (Damas, 2017):

- Identificação dos atributos;
- Normalização;
- Implementação.

Quando o nível de complexidade da informação a extrair obriga a usar como origem um conjunto grande de tabelas ligadas entre si, é preferível recorrer a *Views*, para que o código das consultas não se torne muito complexo (Damas, 2017)

É muito versátil na sua aplicabilidade, podendo ser dimensionada tanto para pequenas empresas ou organizações com uma estrutura maior e mais exigente, prevendo soluções tanto a longo como a curto prazo.

Low-code

Entende-se cada vez mais que o sucesso de qualquer negócio envolve a satisfação do cliente. Tendo em atenção essa máxima, *low-code* é sem dúvida, uma das tendências tecnológicas dos últimos anos, estas ferramentas estão claramente a cumprir este objetivo.

As plataformas de *low-code* permitem a criação de aplicações com programação visual, segundo (Morgan, Lisa, 2019). Este sistema facilita e acelera a sua produção permitindo uma

rápida entrega de soluções de negócio com uma menor necessidade de código escrito. Uma proposta que resolve rapidamente os processos de negócio respondendo depressa às alterações solicitadas pelos clientes ou erros que possam ocorrer.

Permite também que profissionais experientes na área entreguem trabalho de forma mais rápida e fiável. Os modelos visuais são utilizados numa interface gráfica para construir e configurar aplicações. Os desenvolvedores avançam com toda a infraestrutura e a reimplementação de padrões que podem dificultar o desenvolvimento, permitindo-lhes assim desconcentrarem-se nas partes únicas de cada aplicação.

Estas plataformas descrevem uma família de ferramentas que ajuda os desenvolvedores a criar aplicações completas de forma visual, usando interfaces de *drag-and-drop* ao invés de escrever milhares de linhas complexas de código e sintaxe. As plataformas *low-code* possibilitam aos utilizadores a construção de aplicações completas com interfaces modernos, integrações, data e lógica rapidamente e visualmente, (Wong et al., 2021).

Tem a vantagem de serem plataformas de fácil compreensão sendo preciso poucos dias de treino para formar a equipa de um projeto, iniciar rapidamente e concluir tudo em menos tempo, com ajuda do acesso remoto e até do *home office*, o que diminui muito os riscos e custos.

Analisando o mercado de plataformas *low-code*, é possível observar-se que certas entidades se distinguem das demais não só em termos de oferta e estratégia, como também em termos de presença no mercado.

Em agosto de 2021, a *Gartner* apontou a *Mendix* e a *OutSystems* como líderes do mercado de plataformas *low-code*, pois segundo a empresa de consultoria norte americana, estas eram aquelas que possuíam a maior capacidade de execução dos seus recursos para que a sua respetiva visão fosse atingida, **Figura 5**.

Figure 1: Magic Quadrant for Enterprise Low-Code Application Platforms



Figura 5 Líderes do mercado de plataformas low-code

Ao analisar o mercado de plataformas *low-code*, é possível observar-se que certas entidades se distinguem das demais, não só em termos de oferta e estratégia, como também em termos de presença no mercado.

Em comparação ao desenvolvimento tradicional, onde os desenvolvedores necessitam de uma formação específica em programação, nomeadamente requerida em cursos dedicados à programação tradicional. No *low-code* os desenvolvedores não precisam de fazer programação manual, isso acelera o processo de implementação do *software* em produção, segundo (Säfsten, 2022). A **Tabela 2** apresenta uma comparação entre o desenvolvimento tradicional e o desenvolvimento em *low-code*.

Tabela 2 Low-code VS Tradicional (Adaptado)

Características	Desenvolvimento em <i>low-code</i>	Desenvolvimento Tradicional
<i>Definição</i>	O desenvolvimento em <i>low-code</i> , é criar aplicações de <i>software</i> usando uma interface gráfica, ou seja, modelagem visual.	O desenvolvimento tradicional é construir aplicações de <i>software</i> utilizando codificação manual.
<i>Ferramentas</i>	Ferramentas de desenvolvimento de aplicações rápidas como <i>Mendix</i> e <i>OutSystems</i> são usadas para o desenvolvimento de aplicações <i>web</i> de <i>low-code</i> .	As estruturas <i>web</i> e as linguagens de programação são usadas por desenvolvedores para aplicações de código tradicional.
<i>Conhecimento de Codificação</i>	Mínimo de conhecimento necessário. Os desenvolvedores de <i>low-code</i> requerem conhecimentos básicos de codificação.	Desenvolvedores profissionais de <i>web</i> com habilidades avançadas de codificação e conhecimento de múltiplas estruturas <i>web</i> tradicionais.
<i>Velocidade de Desenvolvimento</i>	As plataformas de desenvolvimento em <i>low-code</i> , são conhecidas pela sua rápida evolução. É possível desenvolver uma aplicação <i>web</i> em apenas 1-3 semanas.	O desenvolvimento tradicional requer codificação e assim, pode levar 2-8 meses para ser concluído.
<i>Personalização</i>	As plataformas <i>low-code</i> pode adicionar uma personalização muito limitada à aplicação <i>web</i> .	A codificação tradicional de aplicações <i>web</i> , pode ser adaptada de forma a satisfazer os seus requisitos específicos.
<i>Agilidade</i>	As aplicações de <i>low-code</i> são mais ágeis porque pode fazer alterações rapidamente. As suas novas versões de aplicações serão mais rápidas e quaisquer erros podem ser desviados e resolvidos imediatamente.	O desenvolvimento personalizado tradicional, as mudanças podem ser lentas. Mas seguindo práticas ágeis de desenvolvimento, as aplicações <i>web</i> tradicionais podem ser ágeis.

Implementação	A implementação de aplicações de <i>low-code</i> é rápida. Mas tem acesso a plataformas limitadas. Dependendo das ferramentas de <i>low-code</i> que está a utilizar, só pode ser executado em plataformas suportadas.	Com aplicações <i>web</i> tradicionais, você é livre de implementar para todas e quaisquer plataformas.
Qualidade	A integridade é limitada. A escalabilidade é limitada. Desempenho e velocidade é padrão. O <i>debugging</i> ao vivo, a aplicação está livre de erros.	Integra com todos os sistemas. Alta escalabilidade. Desempenho e velocidade é extremamente alto. A testagem e o <i>debugging</i> requer tempo.
Manutenção	Aplicações de <i>low-code</i> são fáceis de manter comparativamente. Quando a empresa atualizar a plataforma de baixo código, as suas aplicações <i>web</i> de baixo código serão automaticamente atualizadas. Então gasta-se recursos mínimos na manutenção regular.	Aplicações personalizadas na <i>Web</i> podem ser difíceis de atualizar e manter regularmente. Sendo necessário uma equipa dedicada para atualizar o código.

No-Code

As plataformas *no-code*, são construídas para pessoas sem conhecimentos em engenharia informática, que não necessitam de saber linguagens de programação para usar as plataformas.

Todas as características nelas implementadas são criadas pelo seu desenvolvedor, segundo (Hurlburt, 2021).

Estas soluções são idênticas às plataformas de *blogs* e a empresas de *design* de sites de comércio eletrónico que têm páginas pré-construídas, que podem ser usadas e colocadas em produção em minutos.

No entanto, estas ferramentas de *no-code*, normalmente, são muito difíceis de personalizar e não têm nenhuma funcionalidade exclusiva, o que atualmente, se torna um problema com os desafios da modernização digital. Além disso, a maioria destas plataformas foram desenvolvidas inicialmente para resolver um único problema de negócio.

Diferença entre low-code e no-code

Embora as ferramentas de *low-code* como as de *no-code* compartilhem o mesmo princípio de abstração de código para fornecer os benefícios da modelação visual, há uma diferença fundamental na escala e no tipo de aplicações que podem ser construídos por meio das duas abordagens.

As ferramentas *no-code* podem ser ótimas para utilizadores que desejam resolver casos básicos de uso funcional. Estas ferramentas são simples e ideais para construir pequenas aplicações com uma finalidade muito específica. O maior problema que encontramos na utilização das ferramentas *no-code* é a escalabilidade e a capacidade de integração com outros sistemas, como pode ser analisado na **Tabela 3**, uma base de comparação entre o *low-code* e *no-code* (Morgan, Lisa, 2019).

Tabela 3 Comparação entre low-code e no-code (Adaptado)

Características	<i>Low-Code</i>	<i>No-Code</i>
<i>Público-alvo</i>	Desenvolvedores profissionais.	Não desenvolvedores com conhecimento de domínio de negócios que têm uma compreensão profunda de processos e tarefas relacionadas.
<i>Benefícios</i>	Tempo mais rápido para o mercado. Pode ser complementado com código personalizado facilmente. Visibilidade no desenvolvimento de aplicações. Pode ligar-se a ativos empresariais.	Velocidade. Não tenho que esperar pelos desenvolvedores. Visibilidade no desenvolvimento de aplicações. Pode ligar-se a ativos empresariais.

Utilizar casos	Prototipagem. Aplicações que não requerem codificação personalizada completa. Aplicações previamente construídas com ferramentas RAD que precisam de maior funcionalidade e escalabilidade. Aplicações potencialmente complexas e críticas à missão.	Tarefas, processos, aplicações que não requerem codificação personalizada. Aplicações previamente construídas com ferramentas RAD.
Potenciais inconvenientes	Escalabilidade, limitações funcionais sem codificação. Limitações de UI sem codificação. Segurança. Pode não ser viável para aplicações muito complexas.	Escalabilidade, limitações funcionais, limitações de UI. Segurança. Provavelmente não é bom para aplicações muito complexas.
Tipo de fornecedor	Plataforma de baixo código ou fornecedor de ferramentas Fornecedor de ferramentas de desenvolvimento tradicional.	Plataforma sem código ou fornecedor de ferramentas. Fornecedor de ferramentas de desenvolvimento tradicional.

Por outro lado, as plataformas *low-code* oferecem arquiteturas escaláveis, a possibilidade de estender as capacidades da plataforma com *API*'s reutilizáveis e são mais flexíveis no que trata às implementações mesmo em arquiteturas de nuvem (*cloud*). Permitem também que os desenvolvedores façam o teste da aplicação, teste de qualidade e ferramentas de desempenho incorporando técnicas de alta produtividade vistas também em soluções *no-code* para acelerar o desenvolvimento por uma interface gráfica (Hurlburt, 2021).

OutSystems

Segundo (OutSystems, 2022), a tecnologia OutSystems concilia a produtividade oferecida por ferramentas de modelação visual com o uso de plataformas de desenvolvimento. Esta plataforma faculta o que é necessário para criar e disponibilizar rapidamente as aplicações. Tem, também, a vantagem de se poder mudar as mesmas em qualquer altura do seu ciclo de vida.

Caracteriza-se por ser uma plataforma de desenvolvimento ágil onde trabalhamos num ambiente visual altamente intuitivo e com uma tecnologia de publicação que ajuda na entrega

das aplicações e serviços em semanas, contrariamente a outras que poderão levar meses ou anos.

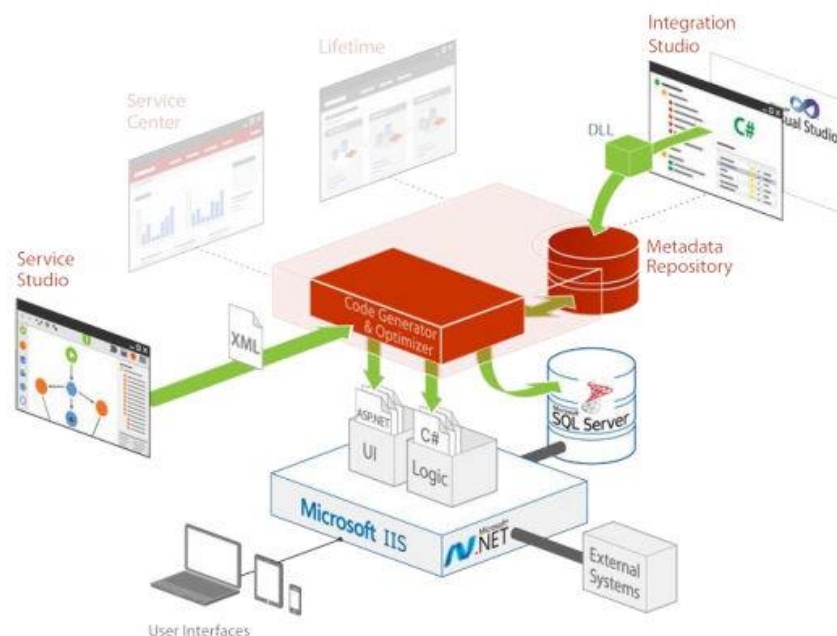


Figura 6 Arquitetura da plataforma OutSystems⁴

A sua arquitetura ilustrada na **Figura 6**, tem como principal componente o servidor *Platform Server*, que trata de todas as etapas necessárias para gerar, otimizar, compilar e fazer *deploy* num servidor aplicacional web. Tem o seguinte conjunto de serviços especializados:

Code generator – serviço responsável por gerar todos os componentes nativos da aplicação, a partir do modelo aplicacional desenvolvido;

Deployment services – serviços responsáveis por fazer *deploy* do código gerado para uma aplicação NET num servidor *front-end* específico;

Application services – serviços responsáveis por gerir as aplicações em tempo de execução, tais como o *Scheduler service* e o *Log service*.

OutSystems são geradas com código *JavaScript* otimizado, envolvidas com código nativo para lidar com interações com o dispositivo, dados e lógica *offline* (OutSystems,2022).

⁴ Disponível em <https://www.outsystems.com/evaluation-guide/platform-runtime/> acessido em 20 de fevereiro de 2022

Architecture Canvas

Em *OutSystems*, um projeto é construído tendo como base uma arquitetura com várias camadas de abstração. Assim, existe uma ferramenta chamada “*Architecture Canvas*” que simplifica o *design* de arquiteturas orientadas a serviços (SOA) que promove uma correta abstração de (micro) serviços reutilizáveis, bem como o isolamento de módulos funcionais distintos.

A “*Architecture Canvas*” da *OutSystems* é composta por 3 camadas e cada uma delas representa o tipo de funcionalidade que é esperado encontrar-se em módulos presentes nessas mesmas camadas. Neste sentido, a primeira camada é dominada de “*End User*”, a segunda de “*Core*” e a terceira de “*Library* (ou *Foundation*)” **Figura 7**.

End-user Modules	Services	UI and processes That provide functionality to the end users
Core Modules	Reusable	Business services Services around business concepts
Foundation Modules	Services	Non-functional requirements Services to connect to external systems or to extend your framework

Figura 7 Camadas da “*Architecture Canvas*”⁵

Nas duas camadas inferiores, os módulos devem conter serviços reutilizáveis. A camada “*Library*” deve englobar serviços que possibilitem a conexão a sistemas externos. Por outro lado, a camada “*Core*” deve abranger serviços que se foquem à volta de conceitos de negócio, visto que é nesta camada que se constrói o modelo de dados e que se define todas as regras de negócio a ser usadas na camada “*End User*”.

Por último, a camada superior “*End User*” não abrange nenhum serviço em específico, uma vez que estão presentes todas as *user stories* e *user interfaces* necessárias para suportar a interação do utilizador com a aplicação (*OutSystems*, 2022) .

⁵Disponível em https://success.outsystems.com/Documentation/Best_Practices/Architecture/Designing_the_Architecture_of_Your_OutSystems_Applications/The_Architecture_Canvas acedido em 23 de fevereiro de 2022

ABORDAGEM METODOLÓGICA

Neste capítulo vai ser descrita a metodologia de investigação usada. Dada as características do presente documento, foi adotado como metodologia de investigação o Modelo *Design Science Research*. Esta abordagem considera dois objetivos que se complementam e ocorrem simultaneamente: o desenvolvimento de um produto em resposta a um problema prático e a criação de conhecimentos científicos e técnicos.

Design Science Research

Em SI, grande parte das investigações possuem um caracter bastante prático, onde é frequente a aplicação de teorias normalmente usadas em áreas como economia, gestão e ciências da computação com o intuito de solucionar problemas que conjugam as TI e as organizações (Peffer et al., 2007)

O DSR é um paradigma que se define como abordagem epistemológica e metodológica para pensar e fazer, tem vindo a afirmar-se na área dos sistemas de informação. Este modelo de investigação apresenta dois objetivos essenciais e complementares:

- Desenvolver um produto para responder a um problema prático num contexto específico;
- Criar novos conhecimentos científicos e técnicos.

O conhecimento científico e técnico necessário à construção de um produto difere do conhecimento científico, uma vez que assenta no conhecimento sobre a arte de fazer.

Na investigação - *design* (produto) verifica-se a articulação entre conhecimento técnico e o conhecimento científico, ou seja, entre produto e teoria.

O processo do DSR pode ser estruturado e definido de várias formas. Contudo, a estrutura que é correntemente aceite e aplicada, foi desenvolvida por (Hevner et al., 2004) e mais tarde atualizada pelo mesmo, facilitando apreender não só a natureza do DSR, mas também apropriar-nos do vocabulário descrito do modelo. A estrutura está exibida na **Figura 8**, consiste em 3 ciclos:

- Ciclo Relevância;
- Ciclo Rigor;
- Ciclo *Design*.

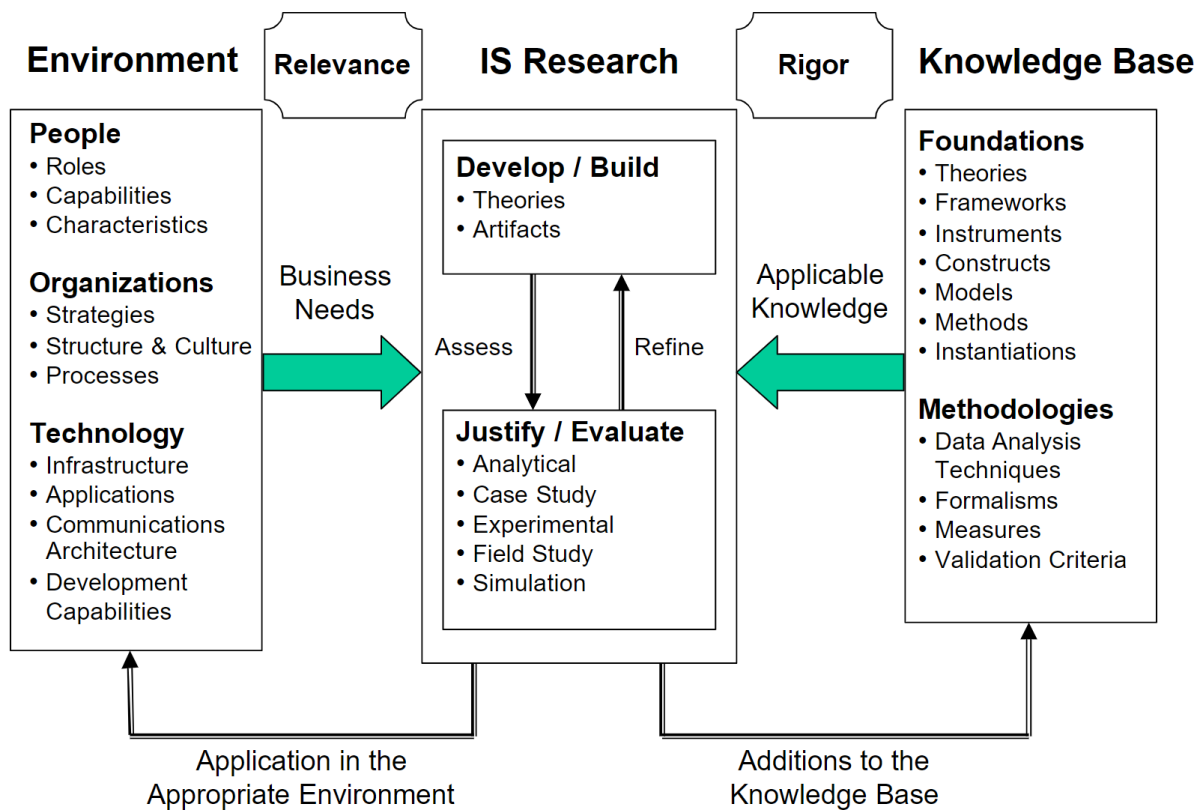


Figura 8 Estrutura de investigação de sistemas de informação (Hevner et al., 2004)

O Ciclo Relevância contém a inter-relação entre atividades de *design* e a correspondente aplicação ambiental que permite a emergência de novos produtos no contexto.

O Ciclo Rigor liga atividades de *design* a conhecimentos e investigação existentes, integrando ou evoluindo esse conhecimento, ao adicionar os resultados do processo de investigação.

O Ciclo *Design* consiste na interação entre a construção e a avaliação do produto a conceber, sendo que a avaliação do produto se pode concretizar de diferentes formas:

- Observação;
- Análise;
- Experimentação;
- Testagem;
- Descrição.

No que se refere ao presente documento de investigação, adotou-se o modelo da DSR de investigação por se apresentar como a metodologia mais adequada à concepção e construção de uma solução, **Figura 9**.

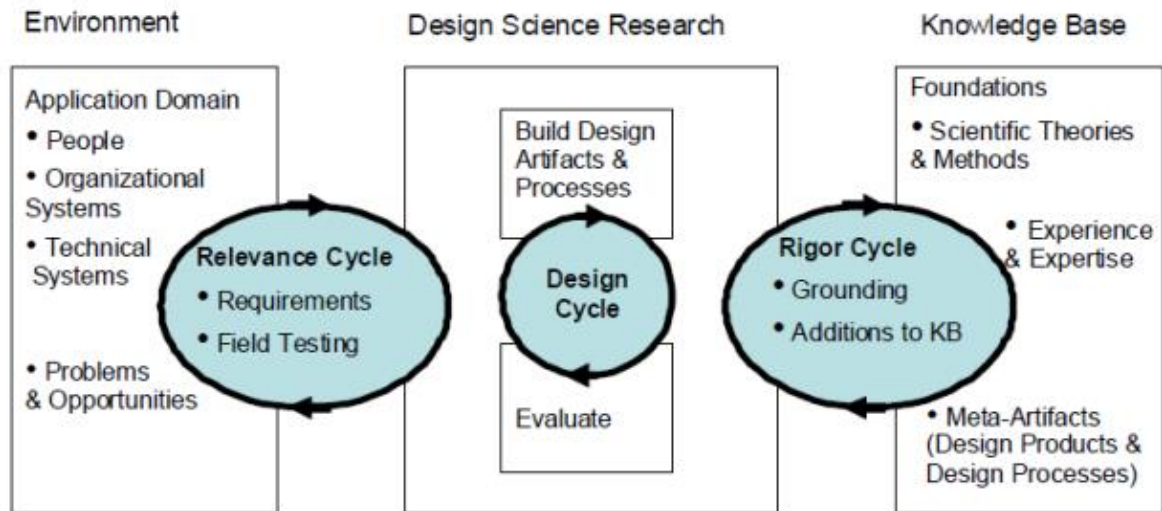


Figura 9 Metodologia DSR (Hevner, 2007)

IMPLEMENTAÇÃO

Este capítulo envolve uma descrição pormenorizada de todos os processos inerentes à construção de uma solução tendo como base a investigação acima apresentada e o estado da arte apresentado no capítulo anterior.

Requisitos Funcionais

Os requisitos funcionais, são os requisitos necessários para que a solução tenha a maior eficácia na conceção dos dados e na sua agilidade funcional.

Dashboard

Os principais requisitos funcionais que serão desenvolvidos na área do *dashboard*:

- Funcionalidade para o utilizador poder ver a aprovação pendente dos recursos atribuídos ou reservados à sua região de negócio. Aprovação do pedido de reserva ou atribuição efetuada pelo utilizador da sua região.
- Funcionalidade para o utilizador pode consultar os pedidos de recursos solicitados à HR no último mês.

Disponibilidade de recursos

Os principais requisitos funcionais que serão desenvolvidos na aplicação atual na disponibilidade de recursos:

- Tem de permitir que o utilizador consulte os recursos através do calendário, selecione o recurso e associe a um projeto.
- Tem de permitir que o utilizador reserve ou atribua um recurso e veja a equipa ou outras informações de projetos onde esteve alocado.
- Após alocação, estará pendente de aprovação se o recurso pertencer a outra região de negócio, caso o recurso pertença a quem está a fazer a alocação.
- Funcionalidade solicitada de um recurso HR quando a empresa não tem recursos disponíveis e solicitar ajuda de RH para contratação, seja através de parceiros ou de contratação direta.

Alocações de Recursos

Os principais requisitos funcionais que serão desenvolvidos na aplicação atual e nas alocações de recursos:

- Funcionalidade para o utilizador poder ver os seus recursos e a sua respetiva alocação e reserva do projeto, férias ou absentismo.

Equipa de Recursos Humanos no Projeto

Os principais requisitos funcionais que serão desenvolvidos na aplicação atual na equipa de projeto:

- Funcionalidade para o utilizador poder alocar uma equipa a um projeto, um ou mais recursos.
- Primeiro é necessário criar o projeto, detalhar o projeto e a respetiva equipa que se destina ao projeto.
- Depois de ter o projeto criado e discriminado, podemos selecionar os recursos e alocar ao respetivo projeto.
- Quando a empresa não tem recursos e efetua um pedido ao RH para contratação, através de parceiros ou de contratação direta.

Casos de Uso

A descrição dos casos de uso está presente na **Figura 10**, nesse documento estão presentes os casos de utilização a considerar no desenvolvimento da aplicação *Staffing Tool*.

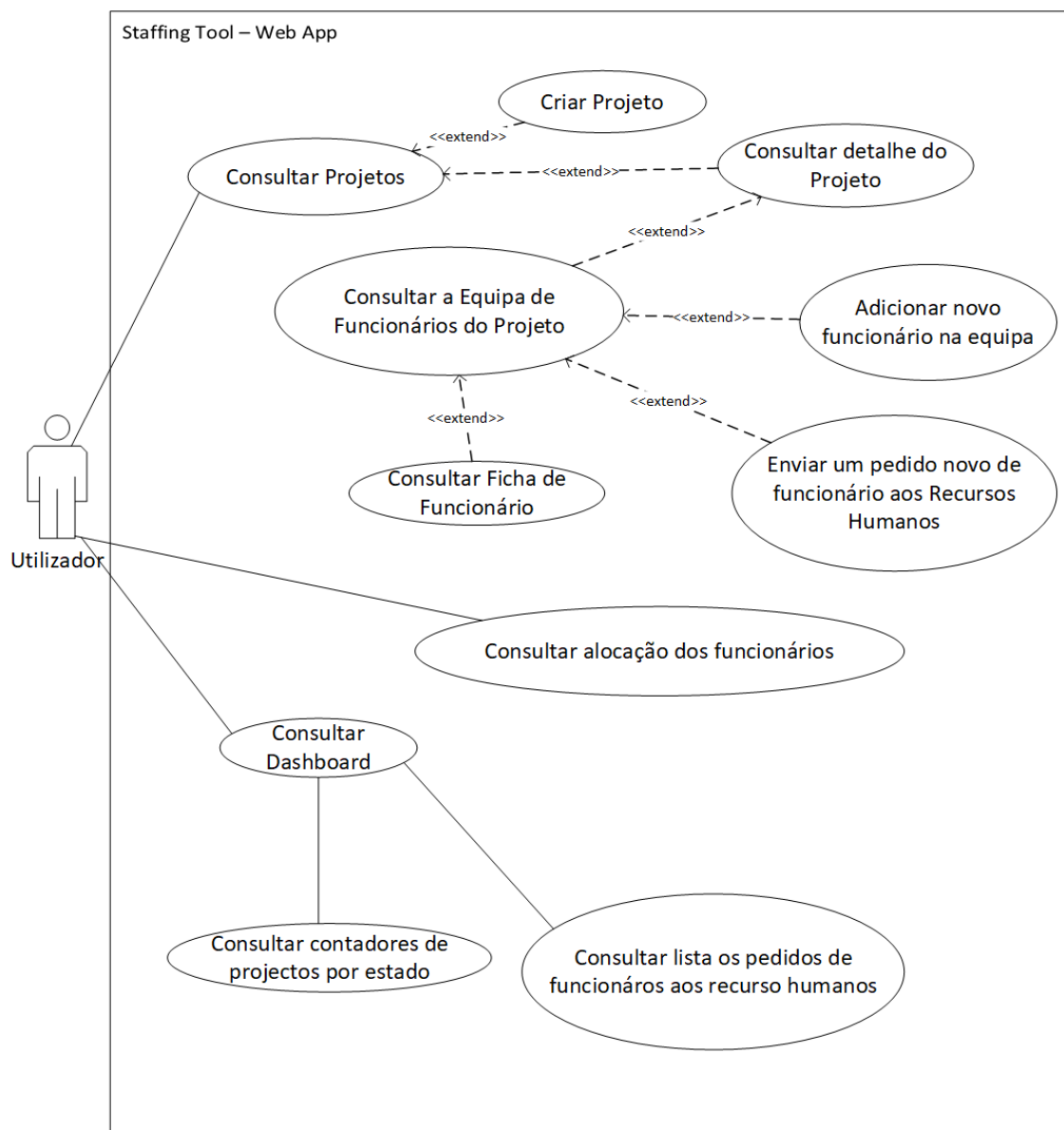


Figura 10 Diagrama de Casos de Uso

Diagrama de Classes

O diagrama representa as classes necessárias para responder ao modelo de negócio, na **Figura 11**, é possível verificar o diagrama de classes produzido para no documento técnico.

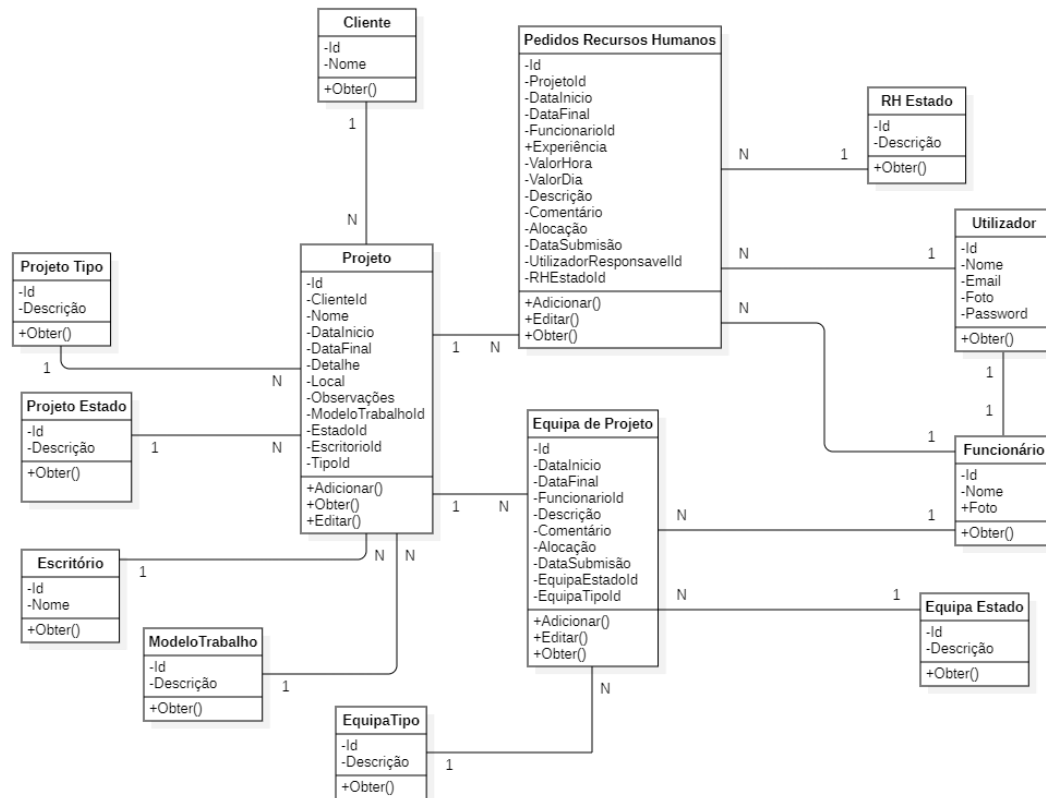


Figura 11 Diagrama de Classes

Modelo ER

O modelo ER representa a base de dados da plataforma *Staffing Tool*. Este modelo contém todas as entidades e seus atributos, representado na **Figura 12**.

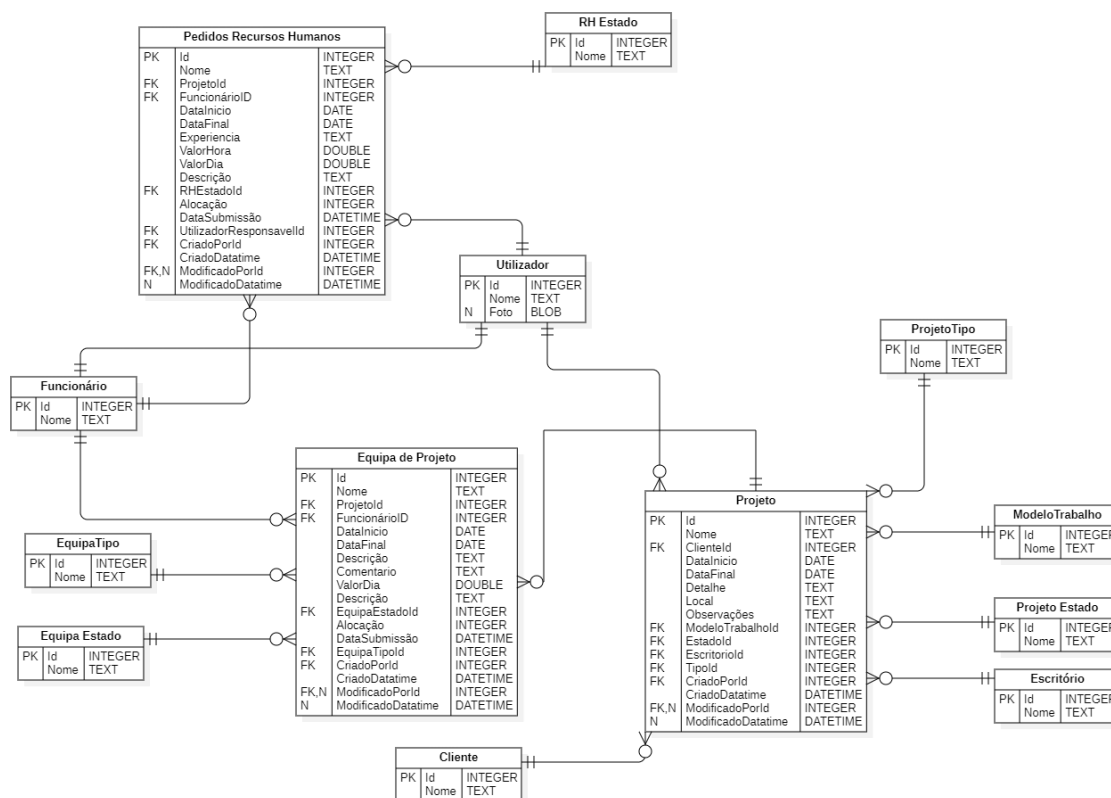


Figura 12 Modelo de ER

Arquitetura

A definição da arquitetura permite estabelecer a ligação entre o projeto e a engenharia de requisitos, envolvendo uma compreensão de alto-nível do funcionamento de um determinado sistema e dos componentes constituintes de forma a integrar entre si. Tal como indica na **Figura 13**, a aplicação *Staffing Tool* baseia-se em outras aplicações para poder dispor das informações necessárias, com a intenção de complementar a consistência dos dados.

As aplicações que partilham os dados, também são desenvolvidas em *OutSystems*, ou seja, a aplicação é determinada em um módulo integral que enriquece todo o ecossistema envolvido.

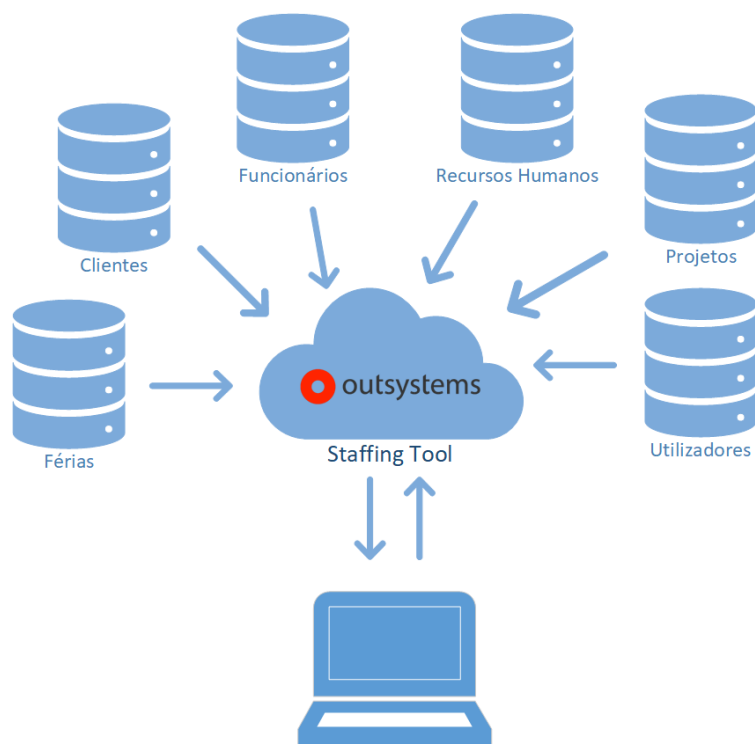


Figura 13 Arquitetura do projeto

O desenvolvimento da “*Architecture Canvas*” passou pela identificação de todos os conceitos inerentes a cada camada. Desta forma, todas as funcionalidades de interação entre utilizador e aplicação foram enumeradas na camada “*End-User*”, assim como todos os conceitos de negócio na camada “*Core*”, **Figura 14**.

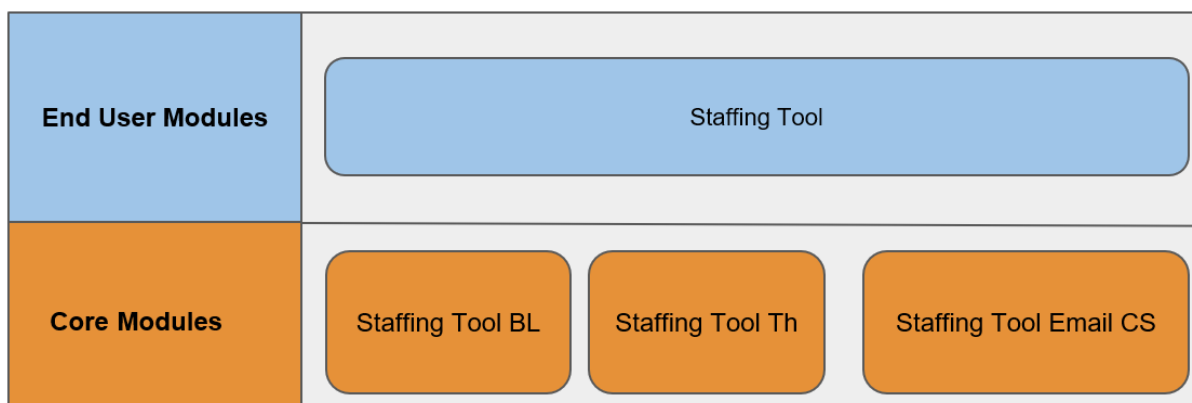


Figura 14 *Architecture Canvas* do projeto

Wireframes

Nos projetos de TI, nomeadamente no desenvolvimento de aplicações, é comum realizar o desenho da solução que se pretende construir. Uma vez que este procedimento é realizado nas fases iniciais de um projeto, o cliente tem a possibilidade de observar antecipadamente a versão avançada do seu produto e verificar se satisfaz eficazmente as suas necessidades, transmitindo esse *mesmo feedback* à equipa de desenvolvimento.

Tendo como base os requisitos funcionais, foram desenhados vários *Wireframes* da aplicação no programa *Balsamiq*. Este processo revelou-se bastante útil, uma vez que possibilitou a aceleração do desenvolvimento de *software*.

Neste seguimento, serão apresentados, na próxima secção, alguns dos *mockups* de cada vertente (*web*), no total foram desenhados 17 ecrãs, os restantes podem ser consultados no **Apêndice I**.

Desenvolvimento da Solução

Criar um novo projeto

Na opção menu *Project Team* – clicar no botão “*New Project Team*” e preencher o seguinte formulário:

- *Customer* – Selecionar o cliente que pretende, campo obrigatório. Ao escolher um cliente por defeito preenche a *label Business Unit* e a *Industry* do cliente.
- *Project* – Deverá escrever o nome do projeto, campo obrigatório.
- *Start Date* – Preencher a data de início do projeto, campo obrigatório.
- *End Date* – Preencher a data fim do projeto, campo obrigatório.
- *Project Technology* – selecionar o(s) tipos de tecnologias que vão estar associadas ao projeto, campo obrigatório.
- *Project Details* – poderá escrever uma descrição sobre o projeto.
- *Location* – deverá escrever a localização do projeto, campo obrigatório.
- *Office* – selecionar o escritório, campo obrigatório.
- *Work Mode* - selecionar o modo de trabalho da equipa do projeto, campo obrigatório.

- *Observations*- poderá escrever uma observação sobre o projeto.

Ao clicar no botão “Save”. O projeto é adicionado à lista de projetos no *Staffing Tool*,
Figura 15.

The screenshot shows the 'New Project Team' form in the Staffing Tool. The form is titled 'New Project Team' and has tabs for 'Project Detail', 'Team', and 'Resource Request to HR'. The 'Project Detail' tab is active. The form includes the following fields:

- Customer ***: A dropdown menu.
- Project ***: A text input field containing 'Demo sprint 6 test'.
- Business Unit**: A dropdown menu containing 'AMERICAS'.
- Industry**: A dropdown menu containing 'Entertainment'.
- Start Date ***: A date picker showing '1 Feb 2022'.
- End Date ***: A date picker showing '1 Apr 2022'.
- Project Technology ***: A dropdown menu containing 'Mobile, Reactive'.
- Project Detail**: A text input field containing 'Demo sprint 6 test, project details'.
- Location ***: A dropdown menu containing 'Lisbon'.
- Office ***: A dropdown menu containing 'PT - Alges Office'.
- Work Model ***: A dropdown menu containing 'Hybrid'.
- Observations**: A text input field containing 'Demo sprint 6 test observation'.

At the bottom right of the form, there are two buttons: a blue 'Save' button and a blue 'Add Team Member' button.

Figura 15 Formulário para criar novo projeto

Criar perfis da equipa associados ao projeto no Staffing Tool.

Criar perfis da equipa associados ao projeto no *Staffing Tool*. Ao clicar no botão “Save” quando está a criar um projeto, automaticamente cria os perfis que estão configurados por *default* na *configuration*. Mas se for necessário adicionar mais um perfil à equipa do projeto, existe um botão “Add Team Member”.

Selecionar recurso humano a um projeto

Em cada perfil (DEV, Tech Lead) temos a uma ação “Edit”. Ao clicar no “Edit” vamos poder selecionar um colaborador que irá integrar à equipa de recursos do projeto, Erro! A origem da referência não foi encontrada..

Name	Delivery Unit	Role	Allocation	Allocation/Reserved	Status	Days	Actions
D Developer		Developer	100%	1 Nov 2021 to 26 Dec 2021	Not Selected	4	Edit Request to HR Remove
D Developer		Developer	100%	1 Nov 2021 to 26 Dec 2021	Not Selected	4	
TL Tech Lead		Tech Lead	100%	1 Nov 2021 to 26 Dec 2021	Not Selected	4	

Figura 16 Editar recurso no projeto

Ao selecionar um RH, pode de ficar de uma cor diferente por exemplo: Se o recurso não esteja na mesma região do projeto, o recurso vai ficar no estado “*Waiting for Approval*”, caso contrário fica com o estado “*Approved*”, representado na **Figura 17**. Mas também é possível eliminar aquele perfil ou fazer um pedido ao RH.

Name	November 29 Mon	December 06 Mon	13 Mon	20 Mon	27 Mon	January 03 Mon	10 Mon	17 Mon	24 Mon	31 Mon	February 07 Mon	14 Mon
Abhishek Hayran	test project	test project	test project	test project	test project	test project	test project	test project	test project	test project	test project	test project
Amit Kumar Verma	Available	Available	Available	Available	Available	Available	Available	Available	Available	Available	Available	Available
Amol Tupe	test 123	test 123	test 123	test 123	test 123	test 123	test 123	test 123	test 123	test 123	test 123	test 123
Ana Rita Nunes	test 123	test 123	test 123	test 123	test 123	test 123	test 123	test 123	test 123	test 123	test 123	test 123
Ananya Shukla	Support	Support	Support	Support	Support	Support	Support	Support	Support	Support	Support	Support
André Costa	Available	Available	Available	Available	Available	Available	Available	Available	Available	Available	Available	Available

Figura 17 Selecionar um recurso

Criar um pedido de recursos “*Request to HR*” num determinado projeto com vários perfis.

Aos Perfis “*Solutions Architect*”, “*Engagement Manager*” e “*Developer*” criados por default no *Team* do projeto vou realizar ação de fazer um pedido aos recursos humanos, **Figura 18**.

The screenshot shows the 'Staffing Tool' interface. At the top, there's a navigation bar with links: Staffing Tool, Dashboard, Resource Allocation, Project Team, and Configuration. The user 'David Matias' is logged in. The main section is titled 'Demo sprint 6 test'. Below this, there's a filter bar with 'Client', 'Project' (set to 'Demo sprint 6 test'), 'Start Date' (1 Feb 2022), and 'End Date' (1 Apr 2022). A tabbed interface shows 'Project Detail', 'Team' (selected), and 'Resource Request to HR'. The 'Team' tab displays a table with columns: Name, Delivery Unit, Role, Allocation, Allocation/Reserved, Status, Days, and Actions. The table lists five team members: an Engagement Manager, a Solutions Architect, and three Developers. Each member has a 100% allocation and a status of 'Not Selected'. A context menu is open over the first Developer, showing options: Edit, Request to HR, and Remove.

Name	Delivery Unit	Role	Allocation	Allocation/Reserved	Status	Days	Actions
Engagement Manager		Engagement Manager	100%	1 Feb 2022 to 1 Apr 2022	Not Selected	44	...
Solutions Architect	Technology and Knowledge	Solutions Architect	100%	1 Feb 2022 to 1 Apr 2022	Not Selected	44	...
Developer		Developer	100%	1 Feb 2022 to 1 Apr 2022	Not Selected	4	...
Developer		Developer	100%	1 Feb 2022 to 1 Apr 2022	Not Selected	4	...
Developer		Developer	100%	1 Feb 2022 to 1 Apr 2022	Not Selected	4	...

Figura 18

Ao clicar no link “*Request to HR Team*” dentro de cada perfil definido num determinado projeto temos de preencher o seguinte formulário:

- *Job* – o campo já vem selecionado com *Job*, campo obrigatório.
- *Quantity* – número de pessoas que precisamos para este tipo de perfil, campo obrigatório e por defeito vem sempre a 1.
- *OutSystems Experience* - é um campo para selecionar os anos de experiência em OutSystems que o candidato terá de ter. As opções apresentadas são: 1 year, 2 years, 3 years, 4 years, + 5 years. Campo obrigatório.
- *Languages* - é um campo para selecionar a língua que o candidato tem de comunicar no projeto. As opções são: *English, French, Spanish*.
- *Allocation* - o campo já vem preenchido por defeito, mas pode ser alterado. Este campo representa a alocação do perfil no projeto. Campo Obrigatório.
- *Start Date* - o campo já vem preenchido com a data de início do projeto por defeito, mas pode ser alterada. Campo obrigatório.
- *End Date* – o campo já vem preenchido com a data de fim do projeto por defeito, mas pode ser alterada. Campo obrigatório.
- *Target Rate*– o campo tem de ser preenchido com um valor por hora e depois é feito o cálculo do dia. Campo obrigatório.

- *Skills* – é um campo onde podemos fazer uma descrição do tipo de conhecimentos que pretendemos para um determinado perfil. Campo Obrigatório.
- *Comments* – é um campo que podemos fazer breves comentários sobre o perfil que estamos a pedir na área dos recursos humanos. Não é um campo obrigatório.
- *Owner* - é um campo que identifica o responsável daquele pedido ao HR. Campo obrigatório.

Ao clicar no botão “Save”. O pedido após ser gravado fica no estado guardado, mas ainda não foi enviado para a área dos recursos humanos, **Figura 19**.

Figura 19 Formulário para fazer pedidos de recursos ao RH.

Consultar e submeter o pedido aos Recursos Humanos.

Consultar e submeter o pedido ao RH. Todos os pedidos ao RH gravado ou submetidos e o respetivo estado deles estão na tabela *Resource Request to HR* com o respetivo estado do pedido. Ao clicar no botão “Request to HR Submit” os pedidos no estado “HR Request Open” passam para estado “HR Request Submitted” e são enviados por email para a área de recrutamento e esses mesmos pedidos podem ser consultados na aplicação recrutamento no menu *Outsourcing Request*, como está exibido na **Figura 20** a listagem de pedido ao RH.

Staffing Tool	Dashboard	Resource Allocation	Project Team	Configuration	David Matias	
---------------	-----------	---------------------	--------------	---------------	--------------	--

< Demo sprint 6 test		Client	Project Demo sprint 6 test	Start Date 1 Feb 2022	End Date 1 Apr 2022
----------------------	--	--------	-------------------------------	--------------------------	------------------------

Project Detail	Team	Resource Request to HR
----------------	------	------------------------

HR Requests						Requests to HR Submit
Job	Rate	OutSystems Experience	Allocation	Allocation/Reserved	Status	Actions
Developer	150€/Hour 1200€/Day	1 Year	100%	1 Feb 2022 to 1 Apr 2022	HR Request Open	...
Developer	350€/Hour 2800€/Day	1 Year	100%	1 Feb 2022 to 1 Apr 2022	HR Request Open	...
Engagement Manager	500€/Hour 4000€/Day	4 Years	100%	1 Feb 2022 to 1 Apr 2022	HR Request Open	...

Figura 20 Listagem de pedidos para enviar aos RH

Ao submeter os pedidos, é enviado um email através da aplicação *Staffing Tool* com os pedidos submetidos num determinado projeto para área responsável pelo recrutamento de Outsourcing, **Figura 21**.

 **Staffing Tool**

Customer:

Project: Demo sprint 6 test

OutSystems Technology: Mobile,
Reactive

Business Unit: AMERICAS

Project Duration: 9 weeks

Job	OutSystems Experience	Allocation	Duration	Request Date
Developer #52	1 Year	100%	9 weeks	2022-02-08
Developer #54	1 Year	100%	9 weeks	2022-02-08
Engagement Manager #53	4 Years	100%	9 weeks	2022-02-08

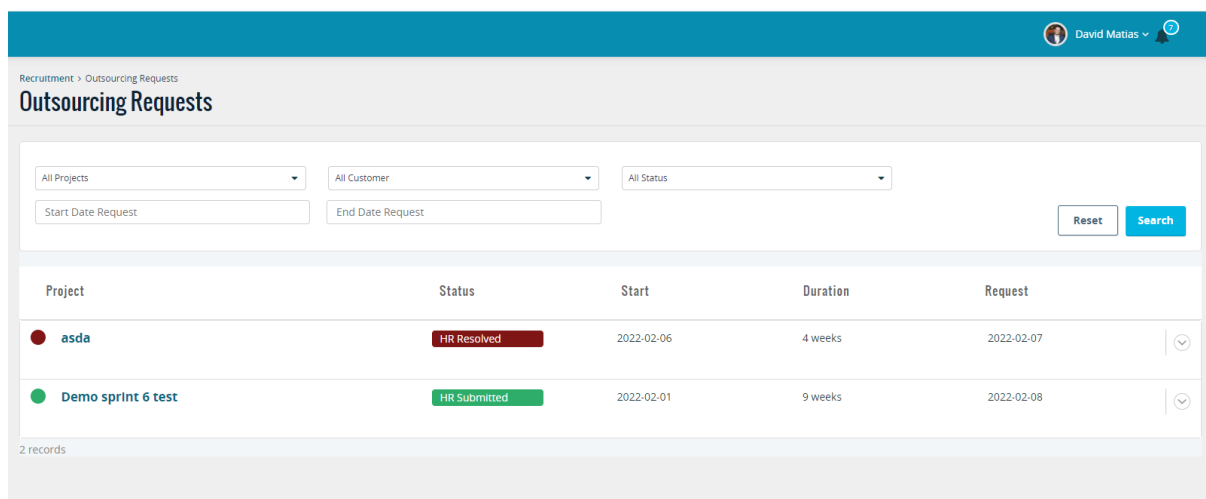
[Go to Recruitment](#)

Figura 21 Email de pedido de recurso ao RH

Os pedidos submetidos podem ser consultados na aplicação RH para a área do *Outsourcing*, como está representado na **Figura 22**.

Listagem de pedidos

Temos uma listagem com todos os pedidos efetuados pela aplicação *Staffing Tool* ao RH, indicando o nome do projeto, cliente, perfil, data de início, duração e a data do pedido.



Project	Status	Start	Duration	Request
asda	HR Resolved	2022-02-06	4 weeks	2022-02-07
Demo sprint 6 test	HR Submitted	2022-02-01	9 weeks	2022-02-08

2 records

Figura 22 Listagem de pedidos ao RH agrupado por projeto

Detalhe dos pedidos

O detalhe dos pedidos submetidos, toda a informação do pedido que foi preenchida no formulário na aplicação *Staffing Tool* é visualizada na aplicação do *Recruitment*. O estado do pedido quando a área de recrutamento começa a contactar as empresas de *outsourcing* passa de “Request to Submitted” para “In Progress” e alteração do estado do pedido faz-se no detalhe do mesmo, presente na **Figura 23**.

Recruitment > Outsourcing Requests > Demo sprint 6 test

< Demo sprint 6 test

Developer: 2, Engagement Ma...: 1

Customer

Business Unit: **AMERICAS**

Industry: **Entertainment**

Project Technology: **Mobile**, **Reactive**

Start: **1 Feb 2022**, End: **1 Apr 2022**

Duration: **9 weeks**

Location: **Lisbon**

Office: **PT - Algés Office**

Work Model: **Hybrid**

Observations: **Demo sprint 6 test observations**

Developer #52 In Progress

Start / End: **1 Feb 2022-1 Apr 2022**, Duration: **9 weeks**, Allocation: **100%**, Status: **HR Submitted**

Target Rate: **15€ Hour - 120€ Day**, Outsystems Experience: **1 Year**, Languages: **English**, Owner: **Monica Furtado**

Skills: **Formação Superior em Engenharia Informática ou similar; Experiência igual ou superior a 2 anos em desenvolvimento de software; Experiência em Jira, Confluence, BitBucket, Jenkins, Groovy, Java e Python; Experiência em Windows e Linux; Guidelines de boas práticas de desenvolvimento.**

Comments: **comments**

Developer #54 In Progress

Start / End: **1 Feb 2022-1 Apr 2022**, Duration: **9 weeks**, Allocation: **100%**, Status: **HR Submitted**

Target Rate: **350€ Hour - 2800€ Day**, Outsystems Experience: **1 Year**, Languages: **English**, Owner: **Vera Pereira**

Figura 23 Detalhe de um projeto e os seus respetivos pedidos

Estado dos pedidos

O estado do pedido por perfil quando é alterado para “*In Progress*” no recrutamento essa informação também é visível no *Staffing Tool* **Figura 24**.

Staffing Tool | Dashboard | Resource Allocation | Project Team | Configuration

< Demo sprint 6 test

Client: Demo sprint 6 test, Project: Demo sprint 6 test, Start Date: 1 Feb 2022, End Date: 1 Apr 2022

Project Detail | Team | Resource Request to HR

HR Requests

Job	Rate	OutSystems Experience	Allocation	Allocation/Reserved	Status	Actions
Developer	15€/Hour 120€/Day	1 Year	100%	1 Feb 2022 to 1 Apr 2022	HR Request In Progress	...
Developer	350€/Hour 2800€/Day	1 Year	100%	1 Feb 2022 to 1 Apr 2022	HR Request Submitted	...
Engagement Manager	500€/Hour 4000€/Day	4 Years	100%	1 Feb 2022 to 1 Apr 2022	HR Request Submitted	...

Figura 24 Listagem dos pedidos ao RH através do *Staffing Tool*

Cancelamento dos pedidos

No *Staffing Tool*, ao fazermos pedidos aos recursos humanos, existe a possibilidade de cancelar o pedido por perfil, ou conseguir resolver esse pedido utilizando um recurso interno.

Em ambas as situações temos de escrever o motivo pelo qual o pedido fica sem efeito. Essa mudança de estado no pedido por perfil é enviada por email para os recursos humanos e também pode ser consultada na aplicação de Recrutamento.

Cancelamento de pedidos por resolução interna

Nas imagens apresentadas de seguida estamos a executar um “*Resolve Internal*” a um pedido ao RH, a explicar o motivo sem efeito, como está representado na **Figura 25** e **Figura 26**.

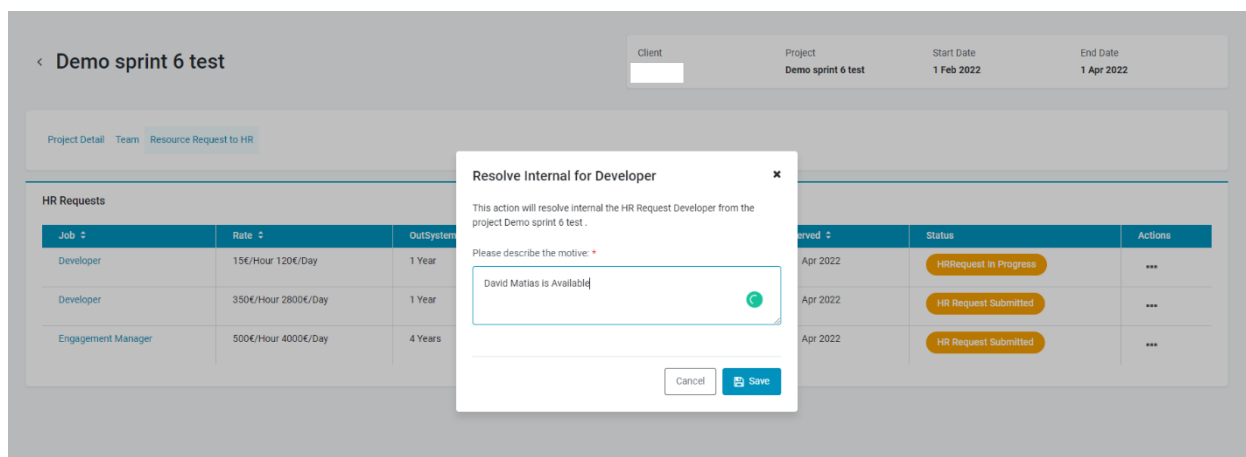


Figura 25 *Resolve Internal for Developer*

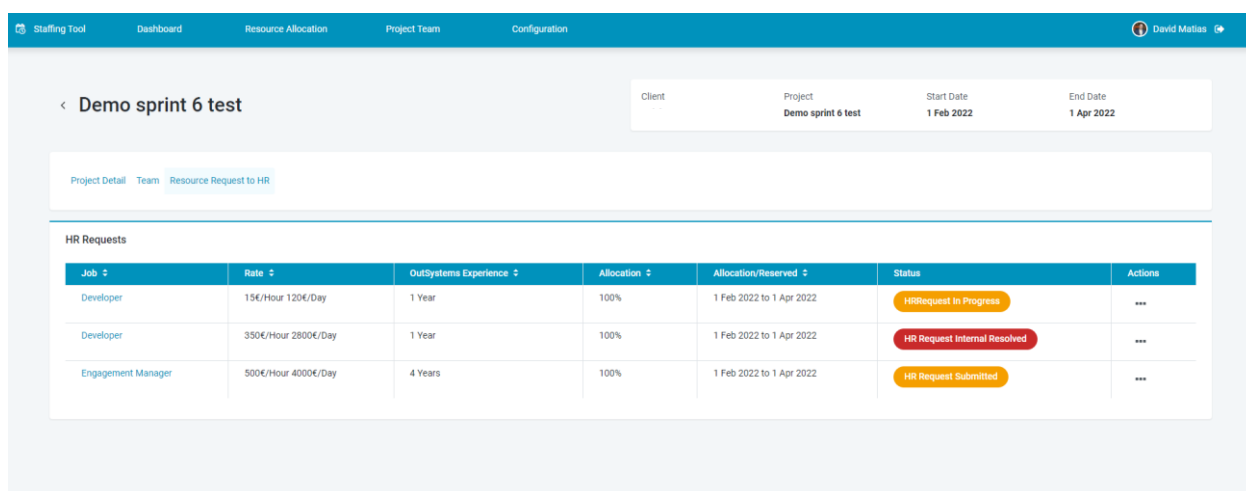


Figura 26 Listagem após atualizar o estado do pedido

Notificação do cancelamento por resolução interna

O email que é enviado para a área dos recursos humanos, irá identificar o pedido e o projeto respetivo. A informação do cliente com o motivo da resolução interna, representado na **Figura 27**.

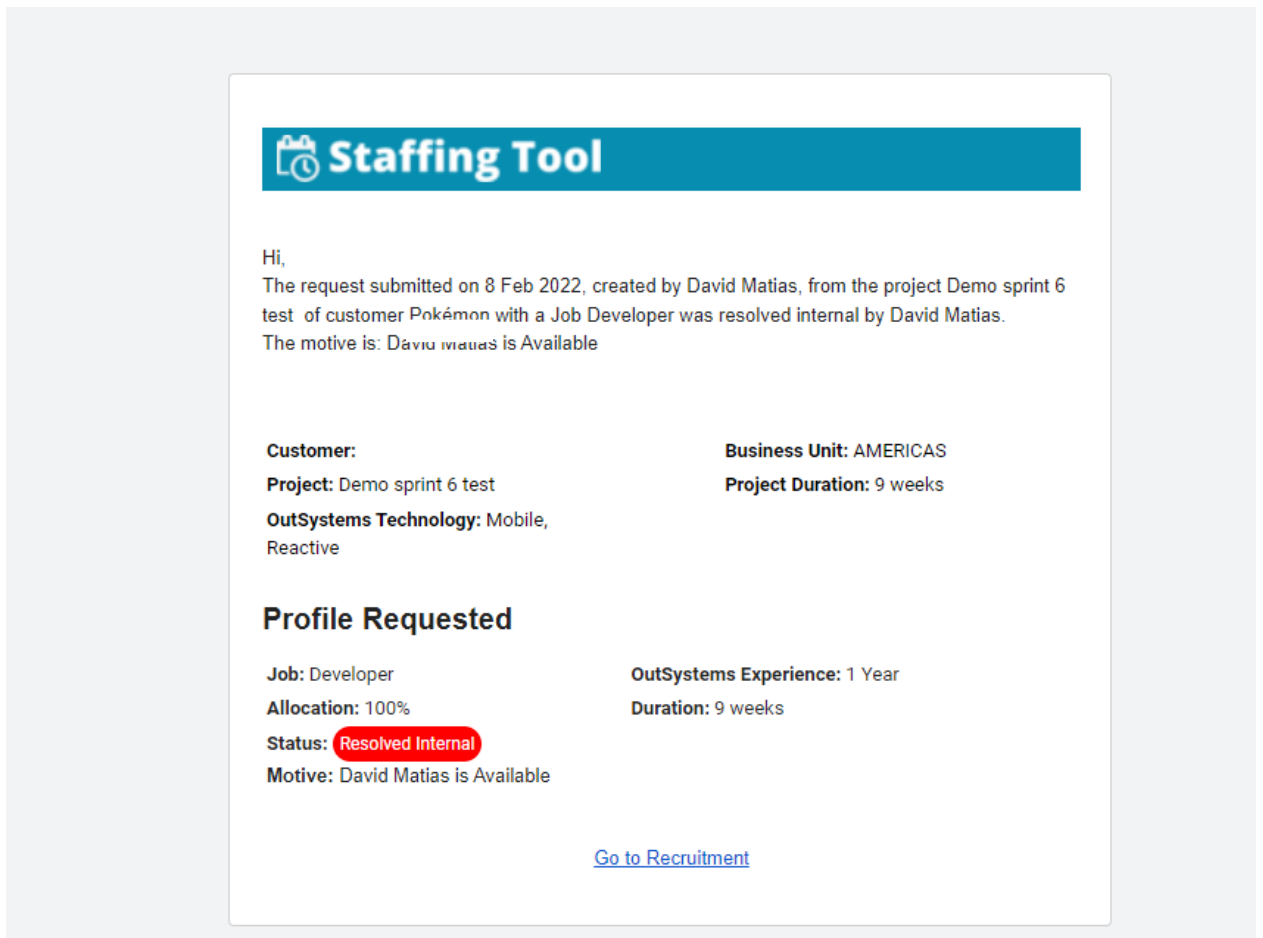


Figura 27 Email a notificar que o pedido de recurso ao RH foi resolvido internamente.

A mudança de estado do pedido para “*Resolved Internal*” no lado do *Staffing Tool* também pode ser consultado na aplicação recrutamento, no menu *Outsourcing Request* no detalhe de pedido com respetivo motivo para esta modificação de estado, como pode está presenta na **Figura 28**.

Project Technology

Mobile Reactive

Start

1 Feb 2022

End

1 Apr 2022

Duration

9 weeks

Location

Lisbon

Office

PT - Alges Office

Work Model

Hybrid

Observations

Demo sprint 6 test observations

Developer #54

Start / End

1 Feb 2022-1 Apr 2022

Duration

9 weeks

Allocation

100%

Status

Resolved Internal

Target Rate

350€ Hour - 2800€ Day

Outsystems Experience

1 Year

Languages

English

Owner

Vera Pereira

Skills

Required qualifications to be successful in this role

Advanced knowledge and hands-on technical experience with OutSystems in an Agile development;

Experience with databases Oracle DB or SQL Server;

Experience with JavaScript;

An ability to quickly translate and navigate through new code from different development styles;

Communication skills in English, speaking and writing.

Desired Qualifications

More than 2 Years Experience;

Certification in Professional Web Traditional;

Experience as a Developer in other technologies is a plus;

Independent worker that requires little supervision beyond initial scope expectation and transfer of knowledge;

Aptitude for application level support, migration and deployment activities

Professional Attributes;

Results-oriented with a strong track record of successful delivery

Participate and network externally in industry forums to share best practices and gain competitive intelligence.

Education and Certification

Bachelor's degree with a focus on Information Technology/Communications, or equivalent.

Comments

Motive

David Matias is Available

Figura 28 Detalhe do pedido *Resolved Internal*

Cancelamento de pedidos por perfil

No *Staffing Tool* ao fazermos pedidos aos recursos humanos. Existe a possibilidade de cancelar o pedido por perfil. Ao cancelar o pedido é necessário escrever o motivo pelo qual o pedido fica sem efeito. Essa mudança de estado no pedido por perfil é enviada por email para o RH e também pode ser consultada na aplicação de recrutamento. Na imagem apresentada, estamos a dar sem efeito um pedido ao RH, a explicar o motivo de cancelamento, neste caso escolhemos “*Cancel*” **Figura 29**.

Staffing Tool

Dashboard

Resource Allocation

Project Team

Configuration

David Matias

< Demo Script 5

Client

Project Demo Script 5

Start Date 3 Jan 2022

End Date 22 Apr 2022

Project Detail

Team

Resource Request to HR

HR Requests

Job	Quantity	Rate	On
Developer	1	30€/Hour 160€/Day	21
Engagement Manager	1	50€/Hour 400€/Day	5
Tech Lead	1	40€/Hour 320€/Day	5

Cancel Request for Developer

This action will cancel the HR Request Developer from the project Demo Script 5.

Please describe the motive: *

We don't need another Developer.

Cancel

Save

Reserved	Status	Actions
2 to 22 Apr 2022	HR Request Submitted	...
2 to 22 Apr 2022	HR Request Internal Resolved	...
2 to 22 Apr 2022	HR Request In Progress	...

Figura 29 *Cancel for Developer*

Notificação do cancelamento por perfil

O email que é enviado para a área dos recursos humanos, identifica o pedido de um determinado projeto e a respetiva informação do cliente, com o motivo do cancelamento, **Figura 30**.

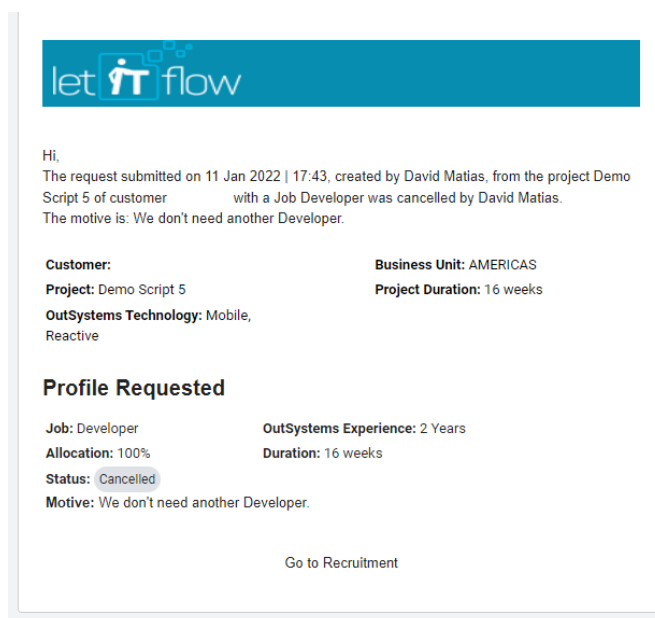


Figura 30 Email a notificar que o pedido de recurso ao RH foi cancelado.

A mudança do estado do pedido para “*Cancelled*” no lado do *Staffing Tool* também pode ser consultado na aplicação de Recrutamento, no menu *Outsourcing Request* no detalhe do pedido com respetivo motivo para esta mudança de estado.

Criar candidaturas outsourcing

Ao criar ou editar uma candidatura no lado do Recrutamento, quando temos de associar um candidato ao projeto do *Staffing Tool*, conseguimos controlar o processo de recrutamento na aplicação. Para isso foi criado um campo com a possibilidade de escolher um projeto de *Staffing Tool*, **Figura 31**.

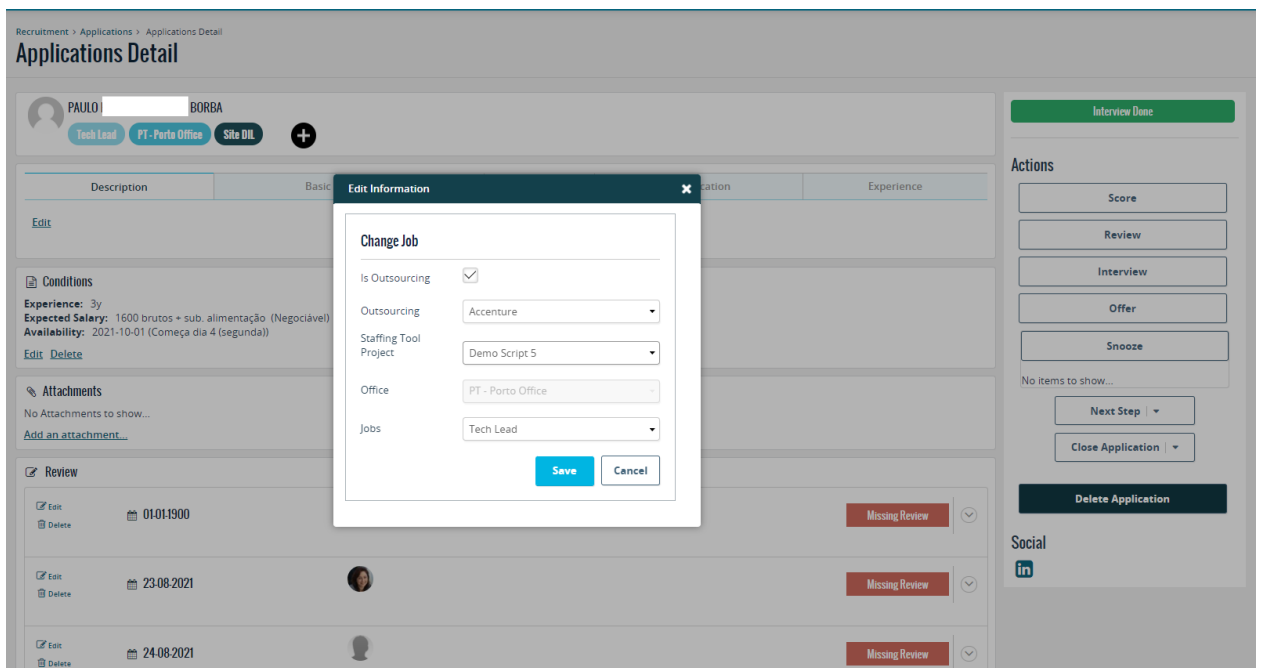


Figura 31 Associar candidato do recrutamento ao projeto

Acesso aos dados dos candidatos

Após a associação do candidato ao projeto do *Staffing Tool*, conseguimos consultar o menu *Outsourcing Request* na listagem dos pedidos, ao clicar na seta temos acesso a informação sobre candidato e o estado do processo no recrutamento, **Figura 32**.

Recruitment > Outsourcing Requests

Outsourcing Requests

All Projects

All Customer

All Status

Start Date Request

End Date Request

Reset

Search

Project	Status	Customer	Job	Start	Duration	Request
Demo Script 5	In Progress	I	Tech Lead	2022-01-03	16 weeks	2022-03-11

Name	Status	Office	Source	Date	
PAULO	BORBA	Interview Done	PT - Porto Office	Accenture	2021-08-12

1 record

Figura 32 Atualização da listagem após associar o recurso ao projeto no RH

Candidatos associados por projeto

Na aplicação *Staffing Tool* também conseguimos consultar o candidato(s) associado no recrutamento a um determinado projeto.

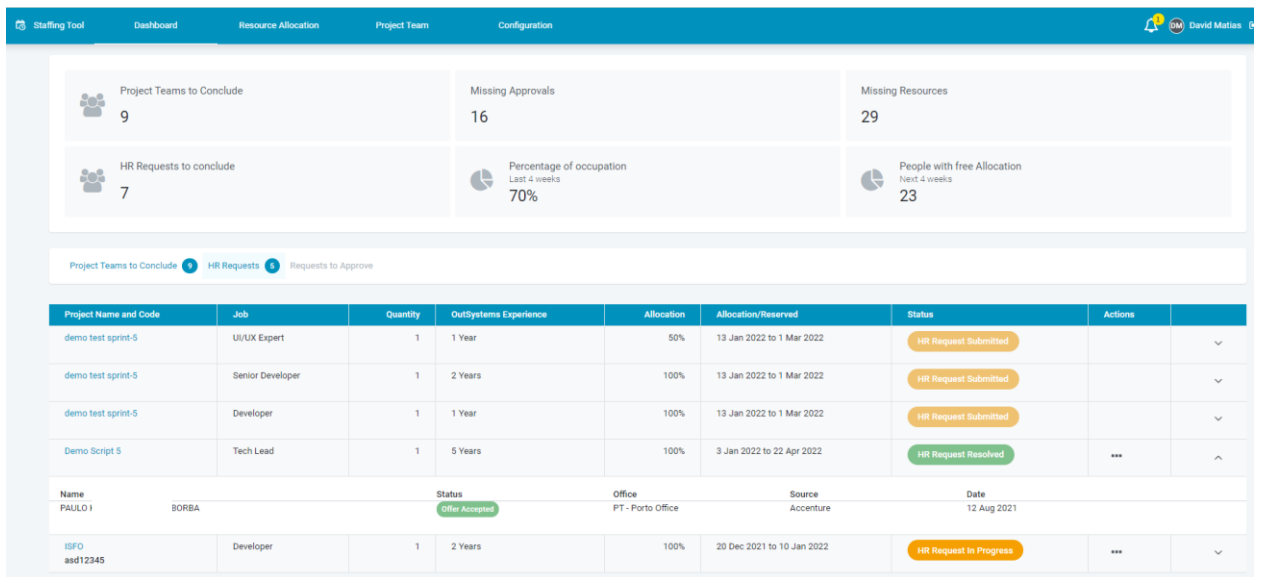


Figura 33 Dashboard do Staffing Tool

Estados dos pedidos

Quando um candidato de *outsourcing* aceita a oferta, automaticamente o pedido fica no estado “*HR Request Resolved*”, **Figura 33**. Na aplicação de recrutamento de um determinado o candidato, ao escolher a opção “*Offer Accepted*” se estiver associado ao projeto *Staffing Tool*. Nas imagens a seguir temos um exemplo da situação de colocar um candidato no estado “*Offer Accepted*”, **Figura 34**.

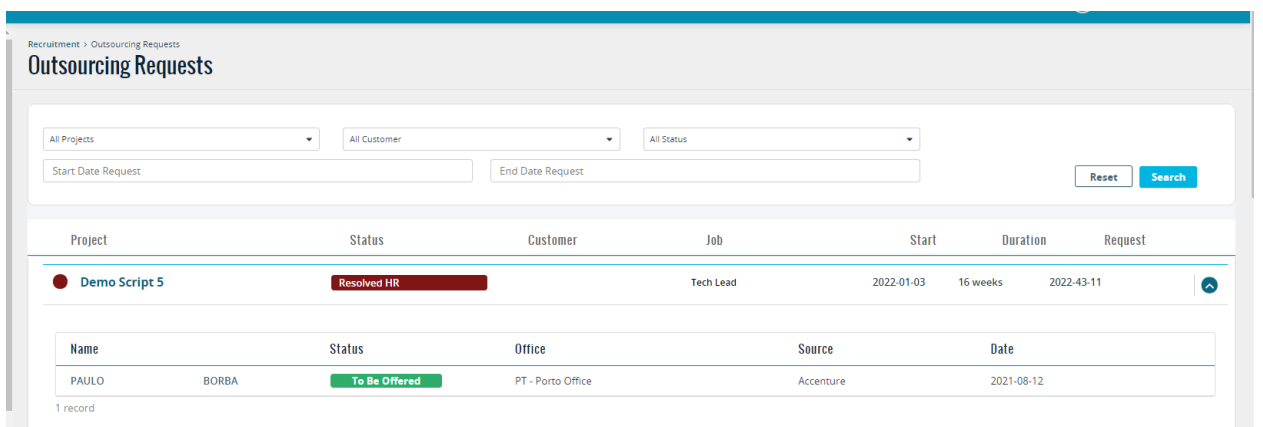


Figura 34 Listagem de pedidos ao Outsourcing

Pedidos nos estados “*HR Request Internal*” e “*HR Request Resolved*”. O projeto está na situação incompleta e temos de associar os colaboradores ao pedido, presente na **Figura 35**.

Staffing Tool Dashboard Resource Allocation Project Team Configuration							
Demo Script 5				Client	Project Demo Script 5	Start Date 3 Jan 2022	End Date 22 Apr 2022
Project Detail Team Resource Request to HR							
Team							
Name	Delivery Unit	Role	Allocation	Allocation/Reserved	Status	Days	Actions
TL Tech Lead		Tech Lead	100%	3 Jan 2022 to 22 Apr 2022	HR Request in Progress	80	
EM Engagement Manager		Engagement Manager	50%	3 Jan 2022 to 22 Apr 2022	HR Request Internal Resolved	80	Edit
SD Senior Developer		Senior Developer	100%	3 Jan 2022 to 22 Apr 2022	Not Selected	80	...
D Developer		Developer	100%	3 Jan 2022 to 22 Apr 2022	HR Request Submitted	80	
D Developer		Developer	100%	3 Jan 2022 to 22 Apr 2022	Not Selected	80	...
SA Solutions Architect	Technology and Knowledge	Solutions Architect	20%	3 Jan 2022 to 22 Apr 2022	Not Selected	80	...
UE UI/UX Expert		UI/UX Expert	50%	3 Jan 2022 to 22 Apr 2022	Not Selected	80	...
QE QA Engineer		QA Engineer	50%	3 Jan 2022 to 22 Apr 2022	Not Selected	80	...

Figura 35 Listagem de pedidos ao RH para o projeto “Demo Script 5”

Resumo dos pedidos

Na **Figura 36**, está representado o *dashboard* onde indica o número de projetos completos, o número de pedidos ao departamento de recursos humanos, a percentagem de alocação de todos os funcionários e a percentagem de disponibilidade de alocação também de todos os funcionários.

Staffing Tool

Dashboard

Resource Allocation

Project Team

Configuration

9

Project Teams to Conclude

7

HR Requests to conclude

16

Missing Approvals

27

Missing Resources

70%

Percentage of occupation
Last 4 weeks

23

People with free Allocation
Last 4 weeks

Project Teams to Conclude 9

HR Requests 7

Requests to Approve 0

Project Name and Code	Job	Quantity	OutSystems Experience	Allocation	Allocation/Reserved	Status	Actions	
demo test sprint-5	Developer	1	1 Year	100%	13 Jan 2022 to 1 Mar 2022	HR Request in Progress	...	▼
Demo Script 5	Tech Lead	1	5 Years	100%	3 Jan 2022 to 22 Apr 2022	HR Request Resolved	...	▲
Name PAULO	BORBA	Status Offer Accepted		Office PT - Porto Office	Source Accenture	Date 12 Aug 2021	Edit	
ISFO asd12345	Developer	1	2 Years	100%	20 Dec 2021 to 10 Jan 2022	HR Request in Progress	...	▼

Figura 36 Dashboard do Staffing Tool

Os testes foram realizados à medida que os métodos e funções foram desenvolvidos, assim todas as funções foram testadas e algumas pequenas correções efetuadas à medida que o desenvolvimento estava a decorrer.

Sem dúvida, a fase de teste desempenha um papel fundamental no *design* de qualquer projeto, pois permite verificar as funcionalidades dos processos implementados, revelando possíveis erros que possam estar a ocorrer.

CONCLUSÕES

Neste trabalho de dissertação foi conceptualizado e implementado um sistema de gestão da alocação, através de uma aplicação *web*, num ambiente de negócios utilizando os recursos dos sistemas de informação já implementados, com o objetivo de gerir os recursos humanos no projeto, controlando a disponibilidade e a sua alocação.

Através da aplicação *web*, facilmente disponibilizamos o acesso à aplicação através de um *browser*, seja em equipamentos móveis ou através de um computador. A sua capacidade de resposta é uma mais-valia para todos os tipos de tamanho do ecrã, ajustando-se automaticamente a cada ecrã, de modo que a informação seja exibida de forma igual.

Através da aplicação, evitamos a utilização de folhas de cálculo, que normalmente são utilizadas para resolver este tipo de problemas. Em que a informação não é centralizada e a consistência dos dados não é a valorizada, devido à ausência de centralização e partilha, torna-se um problema num ambiente organizacional.

Outro dos aspetos positivos deste documento é que estamos a resolver um problema atual existente nas organizações, onde o fluxo de trabalho se torna complexo devido à ausência no sistema de gestão de alocação de RH. A principal vantagem é a reutilização dos dados existentes através da integração numa única aplicação, é uma das formas de criar valor dentro de uma organização, reduzindo o tempo de gestão do projeto e a sua acessibilidade a todos os utilizadores.

Em termos de limitações, este sistema depende de outros SI, nomeadamente o acesso aos dados dos clientes, colaboradores, entre outros. Atualmente uma organização que não tenha um sistema de gestão interna, é quase impossível sobreviver nos dias de hoje.

Este documento apresenta uma solução que é uma mais-valia para a sobrevivência de uma organização que necessita de gerir o RH em projeto, neste caso o sistema foi desenvolvido usando tecnologia *OutSystems*, por outro lado, podia ter seguido por uma abordagem de desenvolvimento mais tradicional, nomeadamente através de linguagens de programação *Open Source*. Contudo o custo existente na utilização da tecnologia *OutSystems* é muito superior comparado com outras tecnologias mais tradicionais. Isto pode tornar a adoção desta plataforma de *low-code* inacessível a organizações mais pequenas com um orçamento mais limitado.

BIBLIOGRAFIA

- Ankrah, E., Sokro, E., Sokro, E., & Sokro, E. (2012). HUMAN RESOURCE INFORMATION SYSTEM AS A STRATEGIC TOOL IN HUMAN RESOURCE MANAGEMENT. *Problems of Management in the 21st Century*, 5. <https://doi.org/10.33225/pmc/12.05.06>
- Arias, M., Saavedra, R., Marques, M. R., Munoz-Gama, J., & Sepúlveda, M. (2018). Human resource allocation in business process management and process mining: A systematic mapping study. *Management Decision*, 56(2), 376–405. <https://doi.org/10.1108/MD-05-2017-0476>
- Aslam, W., & Ijaz, F. (2018). A Quantitative Framework for Task Allocation in Distributed Agile Software Development. *Nan*, 6(nan), 15380-15390-15380–15390. <https://doi.org/10.1109/ACCESS.2018.2803685>
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., & others. (2001). *Manifesto for agile software development*.
- Britto, R., Neto, P. S., Rabelo, R., Ayala, W., & Soares, T. (2012). A hybrid approach to solve the agile team allocation problem. *2012 IEEE Congress on Evolutionary Computation*, 1–8. <https://doi.org/10.1109/CEC.2012.6252999>
- Chang, C. K., Christensen, M. J., & Zhang, T. (2001). Genetic Algorithms for Project Management. *Annals of Software Engineering*, 11(1), 107–139. <https://doi.org/10.1023/A:1012543203763>
- Chiang, H. Y., & Lin, B. M. T. (2020). A Decision Model for Human Resource Allocation in Project Management of Software Development. *IEEE Access*, 8, 38073–38081. <https://doi.org/10.1109/ACCESS.2020.2975829>
- Damas, L. (2017). *SQL - Structured Query Language* (2017th ed.). FCA.

- Engwall, M., & Jerbrant, A. (2003). The resource allocation syndrome: The prime challenge of multi-project management? *International Journal of Project Management*, 21(6), 403–409. [https://doi.org/10.1016/S0263-7863\(02\)00113-8](https://doi.org/10.1016/S0263-7863(02)00113-8)
- Hevner, A. R. (2007). *A Three Cycle View of Design Science Research*. 19, 7.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75–105. <https://doi.org/10.2307/25148625>
- Huemann, M., Keegan, A., & Turner, J. R. (2007). Human resource management in the project-oriented company: A review. *International Journal of Project Management*, 25(3), 315–323. <https://doi.org/10.1016/j.ijproman.2006.10.001>
- Hurlburt, G. F. (2021). Low-Code, No-Code, What’s Under the Hood? *IT Professional*, 23(6), 4–7. <https://doi.org/10.1109/MITP.2021.3123415>
- Huzooree, G., & Ramdoo, V. (2015). Review of Effective Human Resource Management Techniques in Agile Software Project Management. *International Journal of Computer Applications*, 114, 975–8887. <https://doi.org/10.5120/19972-1860>
- Ichniowski, C., Shaw, K., & Prennushi, G. (1995). *The Effects of Human Resource Management Practices on Productivity* (Working Paper No. 5333; Working Paper Series). National Bureau of Economic Research. <https://doi.org/10.3386/w5333>
- Jaselskis, E. J., & Ashley, D. B. (1991). Optimal Allocation of Project Management Resources for Achieving Success. *Journal of Construction Engineering and Management*, 117(2), 321–340. [https://doi.org/10.1061/\(ASCE\)0733-9364\(1991\)117:2\(321\)](https://doi.org/10.1061/(ASCE)0733-9364(1991)117:2(321))
- Kavitha, R., & Suresh, M. (2021). Agile Practices in Human Resource Management. In G. Kumaresan, N. S. Shanmugam, & V. Dhinakaran (Eds.), *Advances in Materials Research* (pp. 713–721). Springer. https://doi.org/10.1007/978-981-15-8319-3_71

- Kieling, E. J., Rodrigues, F. C., Filippetto, A., & Barbosa, J. L. V. (2021). Human resource allocation in projects: A systematic mapping study. *International Journal of Business Information Systems*, 37(4), 505–521. <https://doi.org/10.1504/IJBIS.2021.116999>
- Lepak, D. P., Liao, H., Chung, Y., & Harden, E. E. (2006). A Conceptual Review of Human Resource Management Systems in Strategic Human Resource Management Research. In J. J. Martocchio (Ed.), *Research in Personnel and Human Resources Management* (Vol. 25, pp. 217–271). Emerald Group Publishing Limited. [https://doi.org/10.1016/S0742-7301\(06\)25006-0](https://doi.org/10.1016/S0742-7301(06)25006-0)
- Lin, K.-L. (2011). Human Resource Allocation for Remote Construction Projects. *Journal of Management in Engineering*, 27(1), 13–20. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000032](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000032)
- Mellentien, C., & Trautmann, N. (2001). Resource allocation with project management software. *OR-Spektrum*, 23(3), 383–394. <https://doi.org/10.1007/PL00013358>
- Morgan, Lisa. (2019, July 1). Low-code/no-code development heats up. *SD Times*. <https://sdtimes.com/lowcode/low-code-no-code-development-heats-up/>
- Nankervis, A., Baird, M., Coffey, D. J., & Shields, J. (2019). *Human Resource Management*. Cengage AU.
- Nestel, D., Papadopoulos, N. T., Pascacio-Villafán, C., Righini, N., Altuzar-Molina, A. R., & Aluja, M. (2016). Resource allocation and compensation during development in holometabolous insects. *Journal of Insect Physiology*, 95, 78–88. <https://doi.org/10.1016/j.jinsphys.2016.09.010>
- O’Docherty, M. (2005). *Object-Oriented Analysis and Design: Understanding System Development with UML 2.0*. Wiley.
- OutSystems. (2022). Build Applications Fast, Right and For the Future. <https://www.outsystems.com/pt-br/>

- Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>
- Säfsten, V. (2022). *Low-code vs traditional code* Application of low-code solution in ERP systems. <http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-478456>
- Soares, M. dos S. (2004). Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software. *Revista Eletrônica de Sistemas de Informação*, 3(1), Article 1. <https://doi.org/10.21529/RESI.2004.0301006>
- Stellman, A., & Greene, J. (2014). *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*. O'Reilly Media, Inc.
- Stephens, R. (2015). *Beginning software engineering*. John Wiley & Sons.
- Varajão, J. (2005). *A Arquitetura da Gestão de Sistemas de Informação* (2005th ed.). <https://www.fca.pt/pt/catalogo/informatica/sistemas-de-informacao-engenharia-de-software/a-arquitetura-da-gestao-de-sistemas-de-informacao/>
- Wang, H., Liang, G., & Zhang, X. (2018). Feature Regularization and Deep Learning for Human Resource Recommendation. *Nan*, 6(nan), 39415-39421-39415–39421. <https://doi.org/10.1109/ACCESS.2018.2854887>
- Wang, X., Conboy, K., & Cawley, O. (2012). “Leagile” software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software*, 85(6), 1287–1299. <https://doi.org/10.1016/j.jss.2012.01.061>
- Wong, J., Iijima, K., Leow, A., Jain, A., & Vincent, P. (2021, September 20). *Magic Quadrant for Enterprise Low-Code Application Platforms*. Magic Quadrant for Enterprise Low-Code Application Platforms. <https://www.gartner.com/doc/reprints?id=1-295WSI86&ct=220217&st=sb>

- Yanzer Cabral, A. R., Ribeiro, M. B., & Noll, R. P. (2014). Knowledge Management in Agile Software Projects: A Systematic Review. *Journal of Information & Knowledge Management*, 13(01), 1450010. <https://doi.org/10.1142/S0219649214500105>
- Zavyalova, E., Sokolov, D., & Lisovskaya, A. (2020). Agile vs traditional project management approaches: Comparing human resource management architectures. *International Journal of Organizational Analysis*, 28(5), 1095–1112. <https://doi.org/10.1108/IJOA-08-2019-1857>

APÊNDICES

Apêndice I – Wireframes da Aplicação

Nas seguintes figuras estão representadas *Wireframes* produzidos para desenhar os ecrãs de aplicação.

The wireframe illustrates the 'Staffing Tool' application interface. It features a top navigation bar with a 'Team Project' title and a user profile 'David Mullins'. A left sidebar contains navigation links: 'Dashboard', 'Resource availability', 'Resource Allocation/Reserve', and 'Team Project'. The main content area is titled 'Team Project' and includes a 'New Team Project' button. Below this is a search section with a text input 'Search by Project Name or Resource Name' and several dropdown menus for 'Customer' (Cofidis), 'Project' (FeedBack 360), 'Business Unit' (AMERICA, APAC, EMEA), 'Type Date' (Start Date, End Date), 'Allocation' (Complete, Incomplete), and 'Status' (Waiting for Approval, Approved, Not Approved). A 'Reset' and 'Search' button are also present. The central part of the interface displays a table with three rows of project data:

Code	Project Name	Business Unit	Team	Status	Actions
202103021	STAFFING TOOL	Business Unit EMEA	Team 6	Status Not Approved	▼
202103022	FEEDBACK 360	Business Unit EMEA	Team 7	Status Approved	▼
2021004521	BIG PROJECT	Business Unit EMEA	Team 7	Status Waiting for Approval	Not Approved Approved ▲

Below the table, there is a section for 'Customer: COFIDIS' and 'Industry: Financials'. It includes an 'Allocation' section with radio buttons for 'Complete' and 'Incomplete' (selected). A grid shows resource allocation for various roles: EM, TL, DE, UX/UI, FRONT, BA, SA, and QA. The 'SA' role is currently assigned to 'Fabrice Rosa'. At the bottom, there is a pagination indicator '1 to 26 of 26 items' and a page navigation control showing '1' and '2'.

Figura 37 Wireframe do projeto

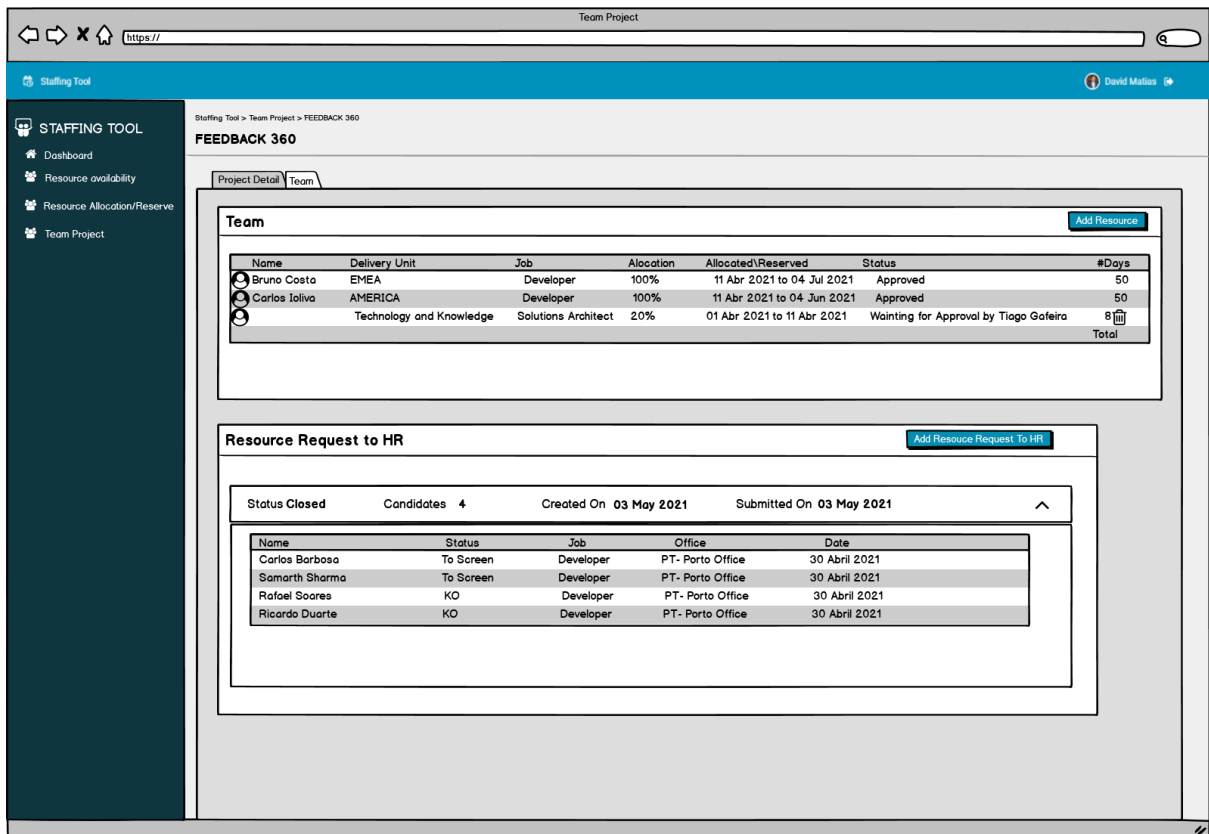


Figura 38 Wireframe da equipa do projeto

Team Project

https://

Staffing Tool

David Matias

STAFFING TOOL

Dashboard

Resource availability

Resource Allocation/Reserve

Team Project

Staffing Tool > Team Project > FEEDBACK 360

FEEDBACK 360

Project Detail

Team

Te

Add Resource

Job

Developer

Job Grade

Rookie

Intermediate

Experienced

Role Model

Company

Academia de código

Adentis

Affinity

DIL

Delivery Unit

AMERICA

APAC

EMEA

Executive Team

IN - BA

IN - DL

IN - HF

IN - QA

INDIA

Pt - MW Team

Shared Services

Technology and K

Office

PT- Porto Office

Start Date

01/04/2021

End Date

30/04/2021

Skills

Outsystems

English

Mobile

Job Level

4

2

1

Add Role

Reset

Search

Select Resource

☐	Name	Delivery Unit	Week 1	Week 2	Week 3	Week 4	Week 5	Job
☒	Bruno Costa	EMEA	Allocated 60%	60%	60%	60%	50%	Developer
☒	Carlos Ioliva	EMEA	Reserved		100%	100%		Developer
☐	Hugo Dias	EMEA			100%	100%		Developer
☒	Igor Khodak	EMEA	100%	100%	100%	100%	100%	Developer
☒	Carolina Martins	EMEA	Vacations	100%	100%	100%	100%	Developer

1 to 26 of 26 item

<

1

2

>

Cancel

Figura 40 Wireframe para adicionar um recurso ao projeto

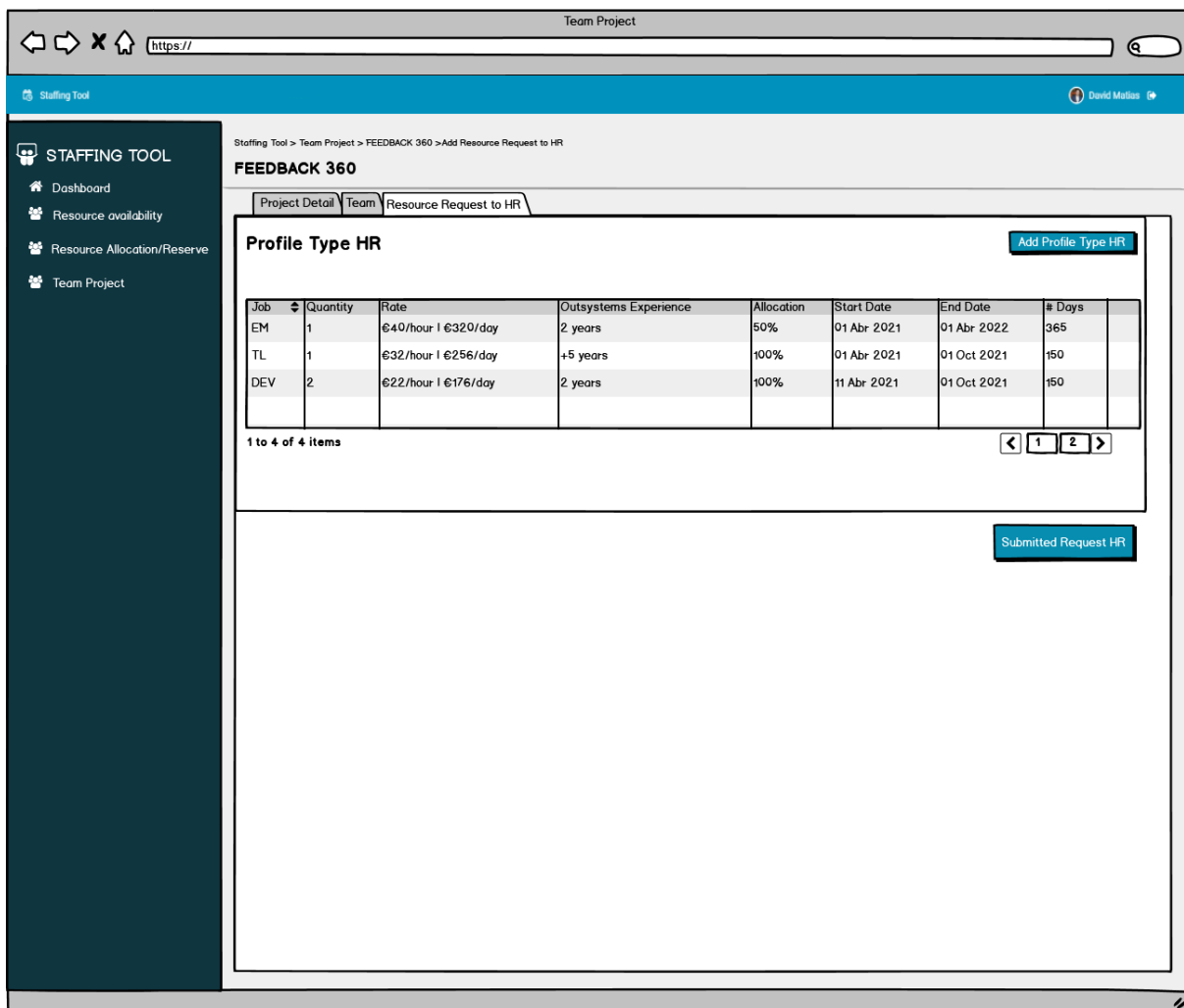


Figura 41 Wireframe da listagem dos pedidos RH

Team Project

Staffing Tool

Staffing Tool > Team Project > FEEDBACK 360 > Add Resource Request to HR

FEEDBACK 360

Profile Type HR

Job	Quantity	Rate
EM	1	
TL	1	
DEV	2	

1 to 4 of 4 items

Add Profile Type HR

Job: Quantity: Start Date:

End Date: Outsystems Experience:

Skills:

Languages: Target Rate:

Maximum Rate:

Comment:

Figura 42 Wireframe do formulário de pedido aos RH

Resource Availability

Staffing Tool

STAFFING TOOL

- Dashboard
- Resource availability
- Resource Allocation/Reserve
- Team Project

Staffing Tool > Resource Availability

Resource Availability

Search by: Name Employee or Employee Id

Job: Developer Job Grade: Routine Outsystems Experience: 2 years 3 years 4 years 5 years Status: Waiting for Approval Approved Not Approved Company: DIL Delivery Unit: CICA AMERICA APKC Technology and Knowledge

Office: PT - Porto Office Reserved Date: From 01/04/2021 To 30/04/2021 Allocated Date: From 01/04/2021 To 30/04/2021

State: Outsystems English Mobile Job Level: L4 L2 L1

Reset Search

Name	Delivery Unit	Allocated	Week 1	Week 2	Week 3	Week 4	Week 5	Job	Company	Allocation	Reserved	Status	Previous DIL Projects
Bruno Costa	EH-EA	Allocated	50%	50%	50%	50%	50%	Dev	DIL	50%			✓
Carla Ioliva	EH-EA	Reserved	100%	100%	100%	100%	100%	Dev	DIL	40%	03 May 2021 to 01 Jul 2021	Waiting for Approval	
Hugo Dias	EH-EA							Dev	DIL	80%			✓
Fabrice Rosa	Technology	100%	100%	100%	100%	100%	100%	SA	DIL	100%			✓
Carolina Martins	EH-EA	Vacations	100%	100%	100%	100%	100%	Dev	DIL	20%			✓

1 to 26 of 26 item

Figura 44 Wireframe da pesquisa de recursos para o projeto

Resource Allocation/Reserve

Staffing Tool

STAFFING TOOL

Dashboard

Resource availability

Resource Allocation/Reserve

Team Project

Staffing Tool > Resource Allocation / Reserve

Resource Allocation / Reserve

Search by Name or Employee Id

Job

Developer

Job Grade

Rookie
Intermediate
Experienced
Role Model

Company

DIL

Office

PT- Porto Office

Business Unit

EMEA
AMERICA
APAC
Technology and Knowledge

Allocated Week of

05-Jul-2021

12-Jul-2021
19-Jul-2021
26-Jul-2021

Reset

Search

Name	Job	Company	Business Unit	Allocation	05-Jul-2021	12-Jul-2021	19-Jul-2021	26-Jul-2021	02-Aug-2021	Reserved
Bruno Costa	Tech Lead - I	DIL	EMEA	40%	Sync 40%	Sync 40%	Colidis	Colidis	Colidis	
João Quadros	Developer - R	DIL	EMEA	60%	BenSaude	BenSaude	BenSaude	Vacations	100%	
Carlos Ioliva	Developer - I	DIL	AMERICA	40%	Staffing Tool	Staffing Tool	100%	100%	100%	
Hugo Dias	Developer - E	DIL	EMEA	80%	Via Verde	Via Verde	Via Verde	Via Verde	100%	

1 to 26 of 26 item

1 2

Figura 47 Wireframe da pesquisa de recursos

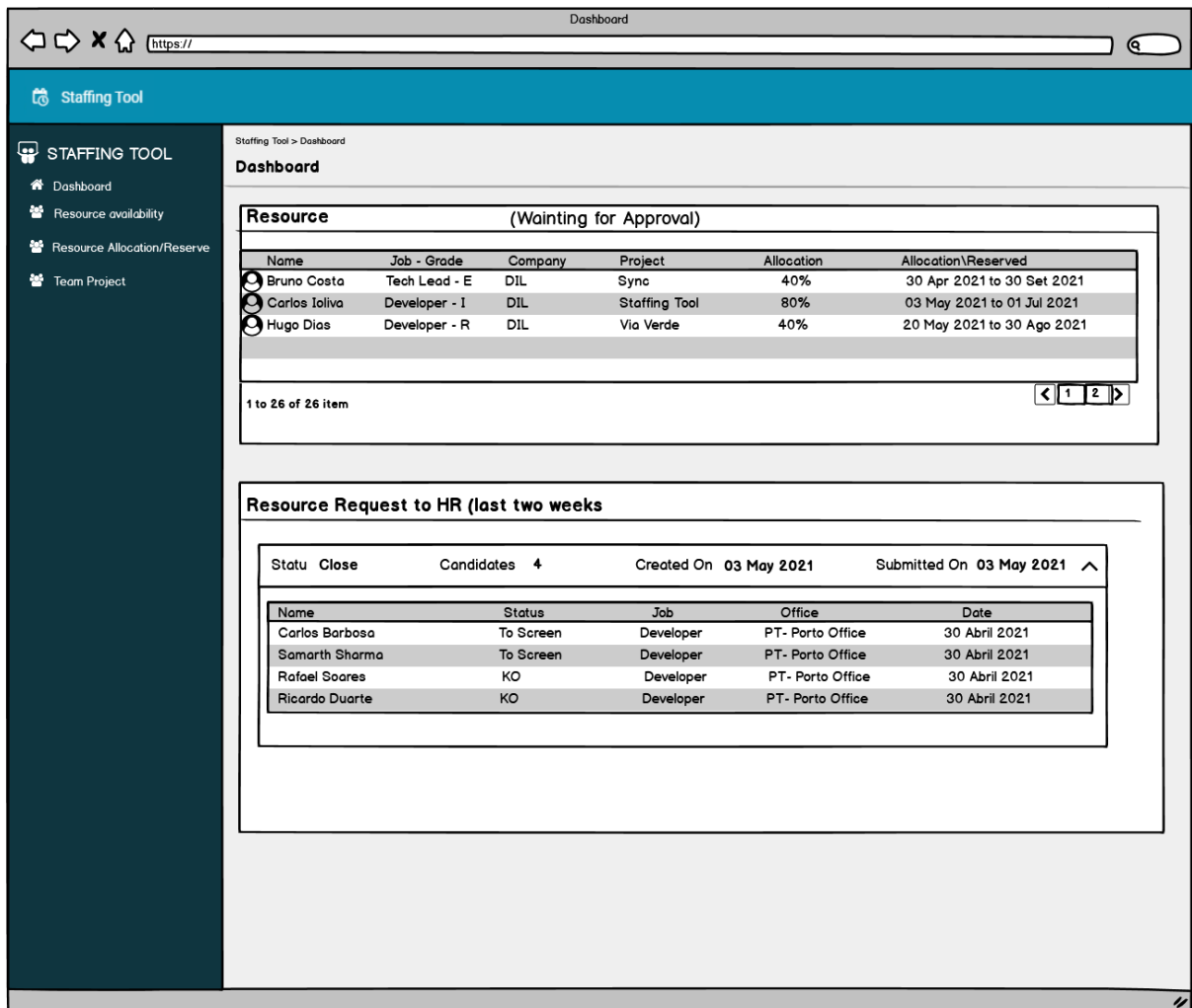


Figura 48 Wireframe do dashboard

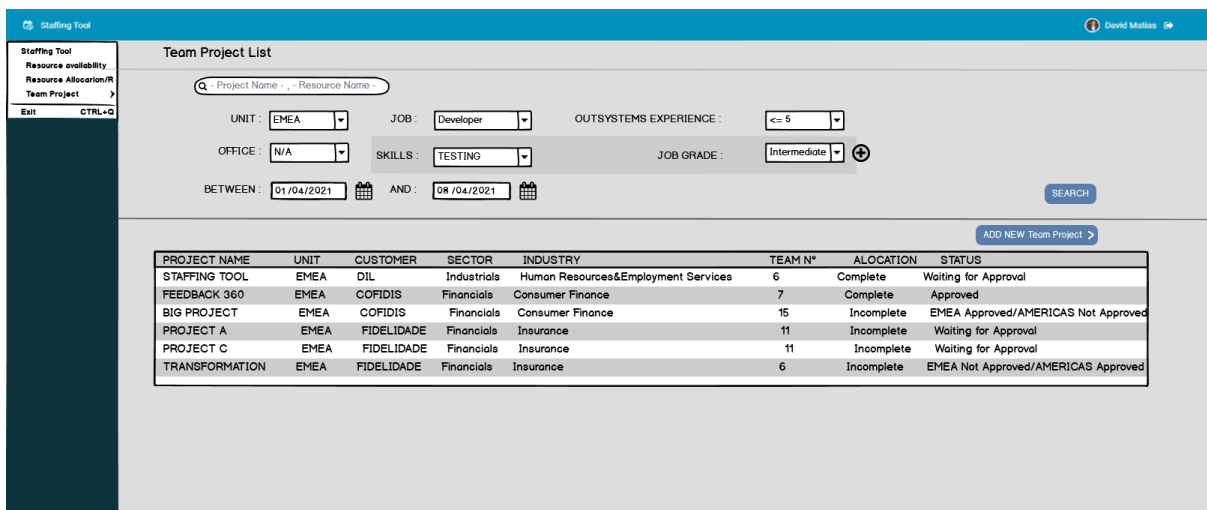


Figura 49 Wireframe da listagem dos projetos

Staffing Tool

Resource availability

UNIT: EMEA JOB: Developer OUTSYSTEMS EXPERIENCE: <= 5

OFFICE: N/A SKILLS: TESTING JOB GRADE: Intermediate
SKILLS: MOBILE JOB GRADE: Intermediate

BETWEEN: 01/04/2021 AND: 08/04/2021

RESERVED BY: Lora Wahnou ALLOCATED BY: Carlos Ialiva RESOURCE ALLOCATION: 60% AVAILABLE AT: 27/04/2021

PREVIOUS PROJECTS: ACDC Project PROJECT: Staffing Tool

TEAM PROJECT: Clark Kent, Bruce Wayne, Selina Kyle, Diana Prince JOB: Developer ALLOCATION REQUEST: 100%

OBSERVATIONS:

CREATED: 01/04/2021 START DATE: 03/05/2021 END DATE: 01/07/2021

DUE DATE: 26/04/2021 STATUS: WAITING FOR APPROVAL

CANCEL SAVE

Figura 50 Wireframe detalhe de um recurso humano e sua alocação

Staffing Tool

Project

Project Description

CUSTOMER: DIL PROJECT NAME: STAFFING TOOL UNIT: EMEA

08/06/2021

SECTOR: INDUSTRY: Human Resources&Em

OBSERVATIONS:

Perfil	Quantidade	Percentagem de alocação	Notas	Start Date	End Date
EM	1	50%	-	01/04/2021	08/06/2021
TL	1	100%	-	01/04/2021	08/06/2021
DEV	2	100%	1 - Dev senior	11/04/2021	04/06/2021
SA	1	20%	-	11/04/2021	04/06/2021

CANCEL SAVE

Figura 51 Wireframe do resumo do projeto

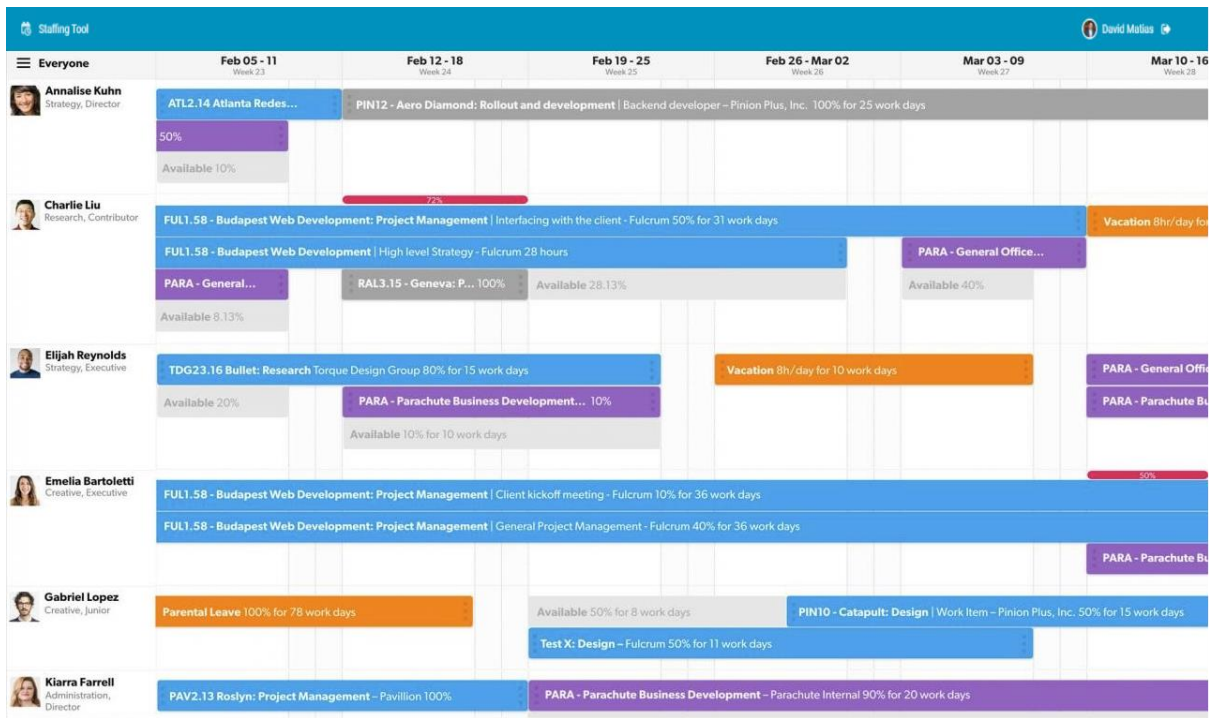


Figura 52 Wireframe do calendário no alocação dos recursos em projeto

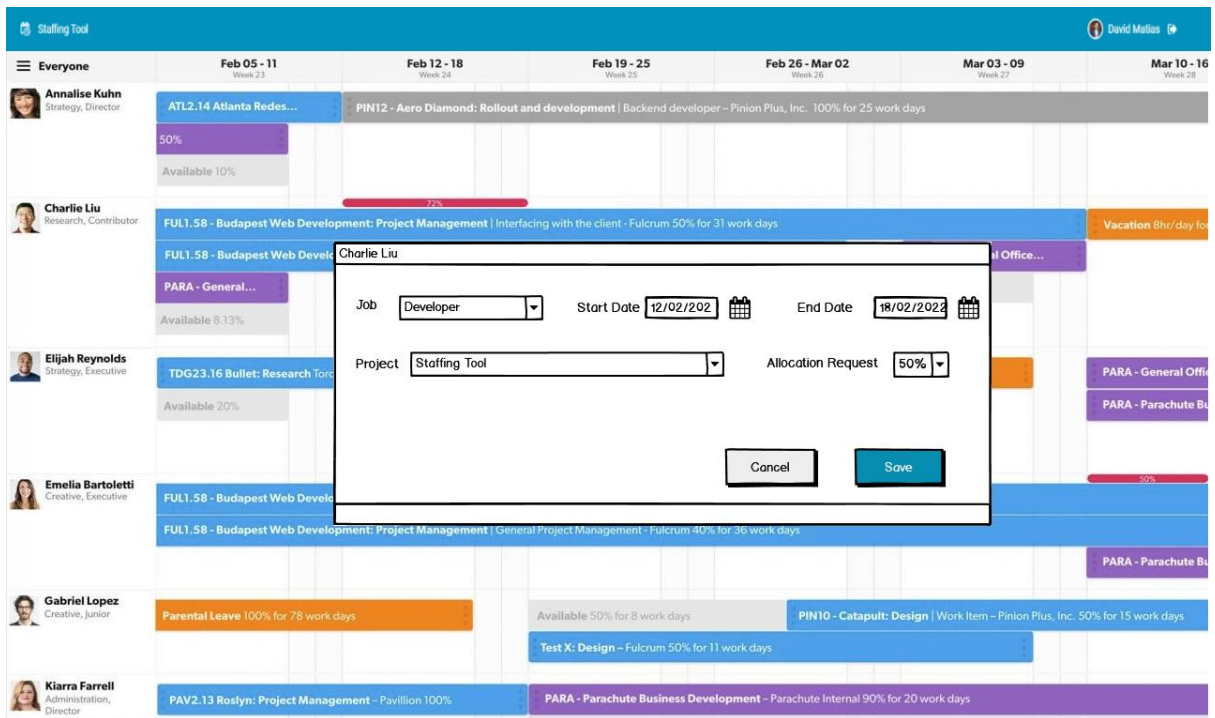


Figura 53 Wireframe da alocação de recursos em projeto através do calendário