



Desenvolvimento de Aplicações Web em Plataforma de Low-Code

André Alexandre Ribeiro Matos

Relatório Final de estágio do Mestrado em Informática apresentado à Escola Superior
de Tecnologia e de Gestão do Instituto Politécnico de Bragança.

Trabalho realizado sob a orientação de
Rui Pedro Sanches de Castro Lopes
Tiago Padrão

Mestrado em Informática
2021-2022

Desenvolvimento de Aplicações Web em Plataforma de Low-Code

**Relatório da UC de Projeto
Mestrado em Informática
Escola Superior de Tecnologia e Gestão**

André Alexandre Ribeiro Matos

2021-2022

A Escola Superior de Tecnologia e de Gestão não se responsabiliza pelas opiniões expressas neste relatório.

Declaro que o trabalho descrito neste relatório é da minha autoria e é da minha vontade que o mesmo seja submetido a avaliação.

André Alexandre Ribeiro Matos

Dedicatória

Dedico este trabalho aos meus pais e família, pois sempre me apoiaram nas alturas mais difíceis e que me passaram a mensagem de nunca desistir dos nossos objetivos, por muito árduo que seja o caminho. Dedico também a todos os meus professores que me ajudaram sempre a crescer neste percurso académico e por fim, não menos importante, a todos os amigos que fiz na cidade dos “Amigos para sempre”.

Agradecimentos

Agradeço aos meus pais, à minha família e amigos por todo o apoio e confiança transmitida a longo do meu percurso académico. Um agradecimento em especial à minha equipa da *Nexllence*, incluindo Ana Abreu, Camila Teixeira, Pedro Inácio e Cristina Bessa pela passagem de conhecimento e pelo ambiente fantástico que proporcionam em cada dia de trabalho. Por fim, mas não menos importante, um agradecimento ao Tiago Padrão e ao professor Rui Pedro, pois tiveram um papel importante durante este período de estágio.

Resumo

O presente relatório surge no âmbito da unidade curricular “Dissertação/Trabalho de Projeto/estágio” inserida no mestrado em Informática do IPB. Este resume os trabalhos, experiências e conhecimentos adquiridos durante a realização do estágio na Glintt, que decorreu sob a orientação do Engenheiro Tiago Padrão da entidade de acolhimento e do Professor Rui Pedro Lopes. Os principais objetivos deste estágio enquadram-se em integrar equipas de trabalho assim como integração nas tarefas diárias, consolidação do funcionamento da empresa e dos processos de comunicação efetuados. O estágio focou-se no desenvolvimento de soluções utilizando a plataforma Low-Code *OutSystems*. Apoiar a área no desenvolvimento/programação de aplicações (Web e/ou Mobile), através da tecnologia *OutSystems*, de acordo com as boas práticas estabelecidas, com qualidade e dentro dos prazos definidos. Por fim, abordar de uma maneira geral como foi o funcionamento de todo o estágio, nomeadamente, quais as capacidades desenvolvidas, quais os conhecimentos adquiridos e projetos desenvolvidos.

Palavras-chave: *Low-Code, OutSystems,*

Abstract

This report is part of the course unit “Dissertation/Project Work/Internship” in the master’s degree program in Informatics of IPB. This report summarizes the work, experiences and knowledge acquired during the internship at Glintt, which took place under the guidance of Engineer Tiago Padrão from the host entity and Professor Rui Pedro Lopes. The main target of this internship is to integrate in work teams as well as in the daily tasks, consolidation of the operation of the company and the communication processes carried out. The internship focused on the development of solutions using the Low-Code *OutSystems* platform. Support the area in the development/programming of applications (Web and/or Mobile), using the *OutSystems* technology, according to established best practices, with quality and within the defined deadlines. Finally, to address in a general way how the whole internship, namely, what skills were developed, what knowledge was acquired and the projects developed.

Keywords: *Low-Code, OutSystems*

Conteúdo

1	Introdução	xv
1.1	Enquadramento	xvi
1.2	Objetivos	xvi
1.3	Estrutura do Documento	xvi
2	Caraterização da Entidade Acolhedora	xix
2.1	Identificação da Entidade Acolhedora	xix
2.2	Descrição da Empresa.....	xix
3	Tecnologias e Ferramentas	xxi
3.1	OutSystems.....	xxi
3.2	Jira.....	xxi
3.3	Teams	xxii
3.4	CSS.....	xxii
3.5	SQL	xxii
4	Plataforma Low-Code <i>OutSystems</i>	xxiii
4.1	Vantagens da plataforma Low-Code.....	xxiv
4.2	Diferenças entre as plataformas <i>Low-Code</i> e <i>No-Code</i>	xxv
4.3	Plataforma de desenvolvimento.....	xxvi
4.4	Service Studio.....	xxvii
4.5	Service Center.....	xxix

4.6	Lifetime	xxix
4.7	Forge.....	xxx
4.8	Arquitetura	xxxi
4.9	Boas Práticas em <i>OutSystems</i>	xxxii
4.10	Abas Principais no <i>Service Studio</i>	xxxiii
5	Atividades desenvolvidas	xxxv
5.1	Variáveis	xxxv
5.2	Ações	xxxvi
5.3	Agregados.....	xxxix
5.3.1	<i>Debug</i>	xlii
5.3.2	Validação de formulários	xlii
5.3.3	Modelação de dados.....	xliii
5.4	Formações.....	xliii
5.4.1	Formação OutSystems Reactive Developer	xliii
5.4.2	Formação de OutSystems Mobile specialist.....	xliv
5.4.3	Formação de Base de Dados	xlvi
5.4.4	Formação de Soft Skills.....	xlvi
5.5	Demonstração do trabalho efetuado	xlvi
5.5.1	Projeto da Avaliação da Liga do Low Code.....	xlvi
5.5.2	Projeto Administração dos Portos	xliv
5.5.3	Processamento de honorários médicos	xliv
6	Estágio	1
6.1	Objetivos do Estágio.....	1
6.2	Diário do Estágio	1
7	Conclusões	liv
7.1	Contributos do Estágio Profissional.....	liv

Lista de Tabelas

6.1	Tarefas Projeto da liga Low-Code.....	li
6.2	Tarefas Projeto da liga Low-Code.....	lii
6.3	Tarefas do Projeto Administração dos Portos.....	lii
6.4	Tarefas Projeto Honorários clínicos	liii

Lista de Figuras

2.1	Logótipo da Glintt e Nexllence	xx
4.1	Paulo Rosado, fundador da <i>OutSystems</i>	xxiv
4.2	Plataforma <i>OutSystems</i> Ágil.....	xxvi
4.3	Service Studio [1].....	xxviii
4.4	Área de trabalho do Service Studio [2].....	xxviii
4.5	Arquitetura [3].....	xxxi
4.6	Exemplo de uma ação conforme as boas práticas de <i>OutSystems</i>	xxxii
5.1	Exemplo de uma variável de entrada	xxxvi
5.2	Exemplo de um preparation [4].....	xxxvii
5.3	Exemplo de uma ação de Servidor [4]	xxxvii
5.4	Exemplo de uma ação de ecrã [4].....	xxxviii
5.5	Exemplo de um agregado [5]	xxxix
5.6	Junção only With [5].....	xl
5.7	Junção With or without [5].....	xli
5.8	Junção With [5]	xli
5.9	Validação de formulários [6].....	xlii
5.10	Comparação entre entidade e uma tabela de base de dados.....	xliii
5.11	Associate Reactive Developer.....	xliv
5.12	Mobile Developer Specialization.....	xliv
5.13	Ecrãs da Aplicação	xlvii
5.14	Base de dados da Aplicação.....	xlvi

Capítulo 1

Introdução

As plataformas *Low-Code* vieram acrescentar bastante valor ao mercado de trabalho. Estas plataformas baseiam-se numa abordagem de desenvolvimento de software que requer pouco ou nenhum código para construir aplicações e processos. Uma plataforma de desenvolvimento de *low-code* utiliza interfaces visuais com funcionalidades simples de lógica de arrastar e largar em vez de extensas linguagens de programação. A plataforma *Low-Code* mais utilizada mundialmente é a tecnologia Outsystems, que atua no mercado do desenvolvimento rápido de aplicações de software, programando as aplicações de uma forma mais visual, o que se torna mais rápido e ágil. Esta tecnologia permite-nos desenvolver aplicações em 3 tipos de paradigmas : Web Traditional, Mobile e Reactive Web. Web traditional é um paradigma mais antigo e que se destina a quem pretende desenvolver aplicações web. O *Mobile* como o próprio nome indica, encontra-se na área de desenvolvimento de aplicações móveis. O *Reactive Web* é o mais recente paradigma da tecnologia *Outsystems* que surge para tornar todo o processo mais rápido para os utilizadores e para quem faz o uso do paradigma, além disso, pode aplicar-se tanto a web como a mobile.

O estágio realizado decorreu durante o período de 18 de Outubro de 2021 a 4 de Maio de 2022, com a duração de 40 horas semanais, perfazendo 29 semanas um total de 1134 horas. O presente relatório está estruturado em 7 capítulos.

1.1 Enquadramento

O presente estágio enquadra-se na unidade curricular de Dissertação; Trabalho de Projeto; Estágio, relativo ao plano de estudos do Mestrado de Informática ministrado na Escola Superior de Tecnologia e de Gestão (ESTiG) do Instituto Politécnico de Bragança (IPB). O presente estágio foi desenvolvido na empresa Glintt. Sendo este conjunto de funções exercidas aliado aos prazos e expectativas definidas pelo cliente.

1.2 Objetivos

Os objetivos deste estágio são adquirir o máximo possível de conhecimento nos diversos paradigmas de OutSystems como Web, Mobile e Reactive e integrar projetos de onde possa contribuir ativamente e absorver o melhor possível todos os conhecimentos transmitidos pelos colegas. Passa também por uma análise técnica dos requisitos pedidos pelo cliente e aplicar os desenvolvimentos conforme o que foi pedido nesses requisitos. Desenvolver conhecimentos sobre componentes lógicas (backend), users interface (frontend) e integrações. Participar em reuniões regulares da equipa para esclarecer, alinhar tarefas ou questões. Os desenvolvimentos devem seguir as boas práticas descritas pela *OutSystems*.

1.3 Estrutura do Documento

Este relatório final de estágio encontra-se dividido em 7 capítulos, estando estes ainda divididos em várias secções, de modo a estruturar da melhor forma o relatório de estágio.

- **Capítulo 1 - Introdução**

Neste capítulo encontra-se uma introdução ao relatório, seguida dos objetivos e enquadramento.

- **Capítulo 2 - Caracterização da Entidade Acolhedora**

Neste capítulo encontram-se informações sobre a empresa que acolheu o estágio.

- **Capítulo 3 - Tecnologias e ferramentas**

É no 3.º capítulo que são demonstradas as tecnologias e ferramentas utilizadas ao longo do período de estágio.

- **Capítulo 4 - Plataforma Low-Code OutSystems**

Este capítulo aborda a tecnologia utilizada durante o período do estágio. Além disso, é especificado o funcionamento da tecnologia, vantagens e comparações com outras tecnologias.

- **Capítulo 5 - Atividades Desenvolvidas**

Neste capítulo são apresentadas informações relativas ao desenvolvimento das atividades.

- **Capítulo 6 - Estágio**

Neste capítulo são apresentadas informações relativas ao estágio: objetivos, demonstração do trabalho realizado e diário de trabalhos.

- **Capítulo 7 - Conclusões**

O último capítulo apresenta uma breve análise ao período de estágio e reflexão de todo o trabalho desenvolvido ao longo do estágio.

Capítulo 2

Caraterização da Entidade Acolhedora

2.1 Identificação da Entidade Acolhedora

Denominação da Empresa: Glintt - Healthcare Solutions

Ano de Fundação: 1995

Actividades: Mercado de Saúde, nomeadamente soluções para Hospitais, Clínicas e Farmácias. Outro pilar de atuação é a Consultoria Tecnológica, através da marca Nexllence, promovem a adoção e o desenvolvimento de soluções baseadas em tecnologias e infraestruturas emergentes para a Administração Pública e organizações de vários setores de atividade

2.2 Descrição da Empresa

A Glintt é uma empresa de referência na Península Ibéria em Consultoria e Serviços Tecnológicos, com mais de 20 anos de experiência. Através da inovação, ajudam a transformar as organizações e melhorar a vida das pessoas. São líderes Ibéricos no mercado de Saúde com cerca de 430 Hospitais, 500 Clínicas e 14.000 Farmácias em Portugal, Espanha e Angola aos quais desenvolvem/implementam soluções para os seus desenvolvimentos e

melhorias. A Glintt está presente em vários mercados como Saúde, farmácia, serviços financeiros, telecomunicações e sector público. Recentemente a Glintt lançou uma marca denominada “Nexllence”, visando desenvolver soluções tecnológicas de excelência para organizações que vão liderar o futuro digital. A nova marca aposta em tecnologias na área *OutSystems*, desenvolvimento de aplicações, automação de processos e alguns projetos em curso com a Administração Pública em identidade digital com operadores de telecomunicações. Na figura 2.1 podemos verifica os logótipos da empresa Glintt e respetiva marca Nexllence.



Figura 2.1: Logótipo da Glintt e Nexllence

Capítulo 3

Tecnologias e Ferramentas

3.1 OutSystems

Esta plataforma é baseada em desenvolvimento “Low-Code”, tecnologias em que a utilização de código no desenvolvimento de produtos e serviços, sendo que é possível adicionar componentes desenvolvidos por programadores. Esta plataforma utiliza poucos recursos, sendo possível criar uma página de uma aplicação, de forma rápida e visualmente interessante e utilizando lógica é possível desenvolver uma aplicação bem estruturada em apenas alguns dias. Esta tecnologia encontra-se atualmente na versão 11, sendo utilizada durante todo o estágio para o desenvolvimento dos vários projetos.

3.2 Jira

É uma ferramenta que permite a monitorização de tarefas e acompanhamento de projetos, garantindo a gestão de todas as suas atividades em único lugar. Este também consegue traduzir em gráficos o trabalho da equipa, acompanhando assim cada um dos *sprints* do projeto. Esta ferramenta foi bastante importante no presente estágio, pois permitiu-me organizar melhor as tarefas a ser realizadas. Cada tarefa está descrita em formato de *User storie*, que se baseia numa explicação informal e geral sobre um recurso de *software* escrita a partir da perspetiva do utilizador final.

3.3 Teams

Microsoft Teams é a plataforma unificada de comunicação e colaboração da Microsoft, que reúne envio de mensagens, vídeo chamadas, armazenamento de arquivos e integração de aplicações no local de trabalho. Esta aplicação foi-me bastante útil, pois era através da mesma que eram feitas as reuniões diárias de equipa, assim como discussões de trabalho anteriormente feito e eventuais dúvidas que surgissem.

3.4 CSS

CSS é chamado de linguagem Cascading Style Sheet e é utilizado para formatar elementos escritos numa linguagem de marcação como HTML. O CSS separa o conteúdo da representação visual do site. Com este é possível decorar uma aplicação da forma pretendida, desde alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos.

3.5 SQL

(SQL) significa “Linguagem de Consulta Estruturada”. Em suma, é utilizada para interagir com os sistemas de gestão de base de dados e executar várias tarefas como inserir e alterar registos, criar objetos na base de dados, gerir utilizadores, consultar informações e controlar transações.

Capítulo 4

Plataforma Low-Code *OutSystems*

OutSystems foi criada em 2001 pelo Engenheiro Paulo Rosado. Esta plataforma encontra-se sediada em mais de 54 países, conhecida por ser um dos “unicórnios” de Portugal (empresas avaliadas em mais de 1000000000 de euros). A OutSystems é uma empresa Portuguesa, fundada no ano 2000, reconhecida internacionalmente pela sua enorme capacidade de desenvolvimento de aplicações com qualidade, rapidez e bastante intuitivas. Esta tecnologia prima pela sua agilidade em arrastar os componentes e desenvolver um ecrã em meros segundos. Atualmente encontra-se na versão 11 que veio trazer um novo *design*, maior rapidez na criação da aplicação, novos “templates” e agilidade na arquitetura, tornando-a mais apelativa de utilizar. Na figura 4.1 podemos observar o fundador da *OutSystems* e o respetivo logótipo.

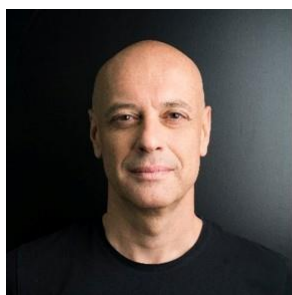


Figura 4.1: Paulo Rosado, fundador da *OutSystems*

4.1 Vantagens da plataforma Low-Code

As principais vantagens do *Low-Code* enquadram-se neste 4 parâmetros: rapidez, implementação com um clique, mais recursos, baixo risco/alto ROI. Em relação à rapidez é possível criar aplicações para várias plataformas simultaneamente e apresentar às partes interessadas exemplos de trabalho em dias ou até mesmo horas. Com o *Low-Code*, basta um único clique para enviar a aplicação para a produção. Se houver uma equipa a trabalhar num grande projeto, não existe a necessidade de esperar que os programadores terminem outro projeto demorado. Com isso, tudo pode ser feito de maneira mais rápida e com menor custo. Estas plataformas asseguram processos de segurança robustos, integração de dados e suporte multiplataforma, podendo ser facilmente personalizados. Isso traz menos riscos e mais tempo para se focarem no desenvolvimento do negócio.

4.2 Diferenças entre as plataformas *Low-Code* e *No-Code*

À primeira vista, é fácil confundir o *Low-Code* com o *No-Code*. Por vezes é possível encontrar diferenças entre as duas plataformas, mas desconsiderando a confusão do mercado, podemos distinguir estas duas plataformas muito facilmente. Existem, literalmente, vários detalhes e recursos que diferenciam as plataformas *Low-Code* das soluções *No-Code*. A maioria deles não fica aparente no nível da *interface* do utilizador, sendo de onde vem grande parte da confusão entre as duas soluções. *Low-Code* é útil para o desenvolvimento de aplicações e portais independentes para dispositivos móveis e para a web, que provavelmente exigirão integração com outros sistemas e várias fontes de dados. Na verdade, pode ser usado para praticamente tudo. As ferramentas *No-Code*, por outro lado, só devem ser usadas para casos de uso de front-end. Portanto, a menos que se desenvolva somente aplicações bem simples e exija pouco em termos de personalização, é provável que o *Low-Code* seja a melhor opção. Outra grande diferença é que *Low-Code* permite a implementação de algumas linhas de código para ajudar no desenvolvimento da aplicação, contrariamente ao *No-Code*.

4.3 Plataforma de desenvolvimento

O poder da plataforma ágil começa com um conjunto de ferramentas e serviços que automatizam totalmente o processo de entrega de aplicações de negócio. Estas agregam-se em sistemas existentes e chegam aos utilizadores através da *Web*, *email* e dispositivos móveis. A figura 4.2 representa o modelo ágil da plataforma *Outsystems*.

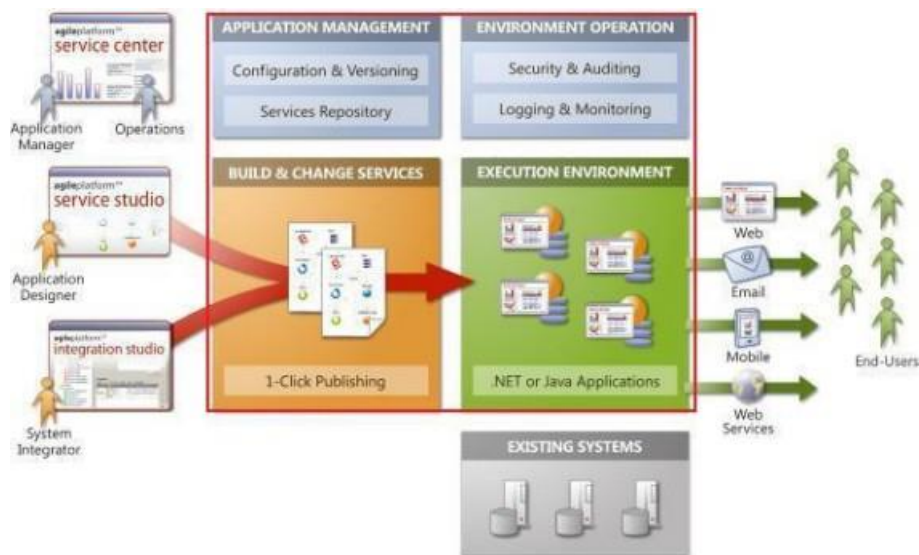


Figura 4.2: Plataforma *OutSystems* Ágil

4.4 Service Studio

O *Service Studio* é o ambiente de desenvolvimento integrado da *OutSystems*. Permite o desenvolvimento da lógica necessária, a criação de bases de dados e a montagem visual de ecrãs que são apresentados ao utilizador da aplicação. Nesses módulos é possível definir o modelo de dados da aplicação assim como permite visualmente a criação da *user interface* tanto para dispositivos móveis e aplicações web. A utilização do CSS também é efetuada dentro destes módulos para melhorar o aspeto visual dessas mesmas interfaces, assim como o desenvolvimento da lógica de negócios e dos processos para suportar essas aplicações.

Após efetuar o login no *Service Studio*, aparecerá um ecrã semelhante à da figura em que mostra todas as aplicações do servidor, onde também há a possibilidade de criar aplicações, instalar componentes de uma loja própria da *OutSystems* e também abrir as aplicações. Na figura 4.4 podem identificar 4 janelas em que se encontram a parte dos processos, a interface na qual são desenvolvidos os ecrãs da aplicação, a parte da lógica onde são criados os métodos e a parte do “Back-end” e a última janela são os dados onde respetivamente se encontra o modelo de dados da aplicação.

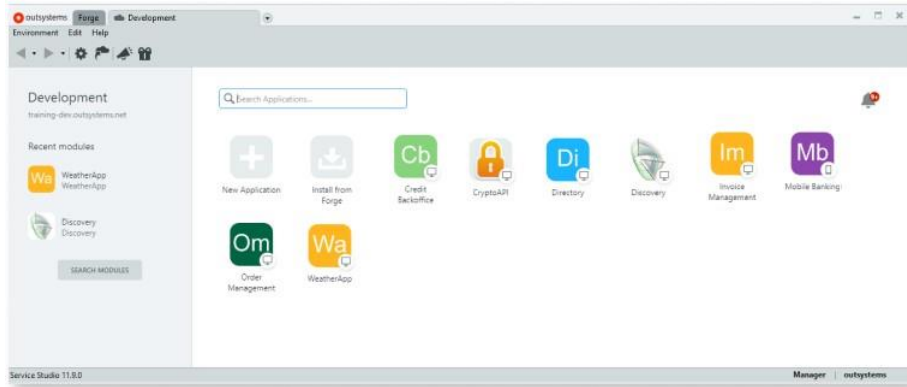


Figura 4.3: Service Studio [1]

Ao abrir uma aplicação o ecrã mostrará algo semelhante à figura 4.3.

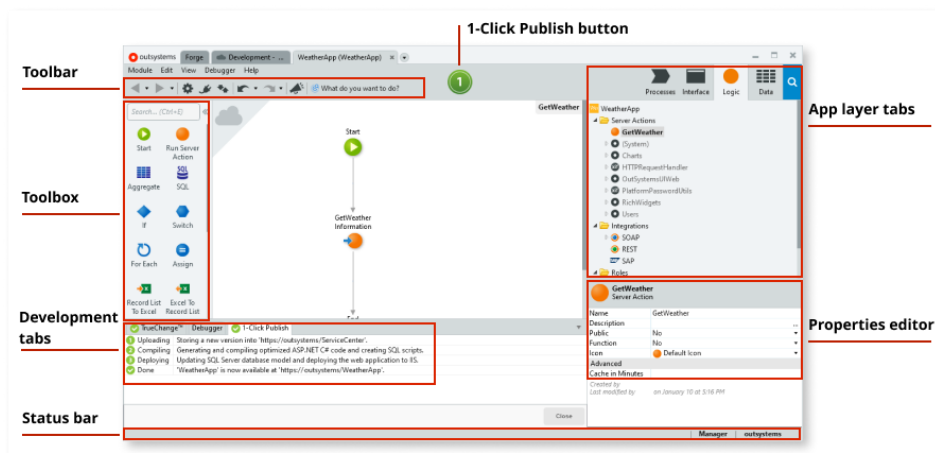


Figura 4.4: Área de trabalho do Service Studio [2]

Este é o aspeto da área de desenvolvimento de uma aplicação em *OutSystems*. Na parte da “toolbar” é possível aceder às definições da aplicação, conseguindo ter uma noção mais específica de algum erro que possa estar a bloquear a aplicação, assim como retroceder/avançar nas alterações. Também há a possibilidade de adicionar alguns componentes que sejam necessários para o desenvolvimento da aplicação. Na parte da “toolbox” encontram-se os elementos necessários para implementar os fluxos. Na parte da “toolbox” também se encontram agregados, que podem ser compostos por uma ou mais tabelas de

base de dados. A parte dos ciclos pode ser implementada através de um *for each, switch* ou utilizando o nó “if”. Também é possível utilizar outros componentes ou mesmo o código desenvolvido através da *forge*, sendo um repositório onde vários utilizadores podem publicar o seu código e utilizar os códigos lá expostos.

Cada fluxo começa com um nó de “start” sendo finalizado por um ou mais nós “end”, havendo alguns casos em que o fluxo possa ser finalizado noutra tipo de nós. Na parte de “development tabs”, pode-se observar os erros que ocorrem em tempo real no desenvolvimento, assim como algumas más práticas ou alguns avisos.

4.5 Service Center

O *Service Center* é uma aplicação web que permite gerir e inspecionar a aplicação desenvolvida, mostrando informação detalhada sobre os erros ocorridos, assim como a localização deles.

4.6 Lifetime

LifeTime é a consola centralizada para gerir os ambientes *OutSystems*, sendo eles, aplicações, utilizadores e segurança, cobrindo todo o ciclo de vida da aplicação, desde o desenvolvimento até à implementação.

Uma infraestrutura típica da *OutSystems* compreende os seguintes ambientes:

- Desenvolvimento: O ambiente onde os aplicativos são inicialmente desenvolvidos e testados;
- Qualidade: o ambiente em que os analistas e utilizadores de negócios experimentam a aplicação para garantir a qualidade;
- Produção: O ambiente que hospeda a versão da aplicação com os quais os utilizadores finais interagem.

4.7 Forge

A *Forge* é um repositório de módulos de código aberto reutilizáveis, conectores, componentes de *interface* do utilizador e soluções de negócios para ajudar a acelerar o tempo de entrega das aplicações.

4.8 Arquitetura

Um ambiente *OutSystems* fornece toda a infraestrutura necessária para permitir o desenvolvimento e entrega de aplicações. As edições disponíveis possuem pelo menos três ambientes adequados para desenvolvimento, teste e produção. Um ambiente agrega um conjunto de componentes que permite ao utilizador desenvolver código visual, integrações, compilar e acrescentar o seu próprio código e acessar e usar as aplicações desenvolvidas no *browser* ou um simulador para dispositivo móvel. O desenvolvimento do ambiente usa vários componentes de infraestrutura subjacentes, como um servidor da aplicação, base de dados para armazenar metadados e dados da aplicação.

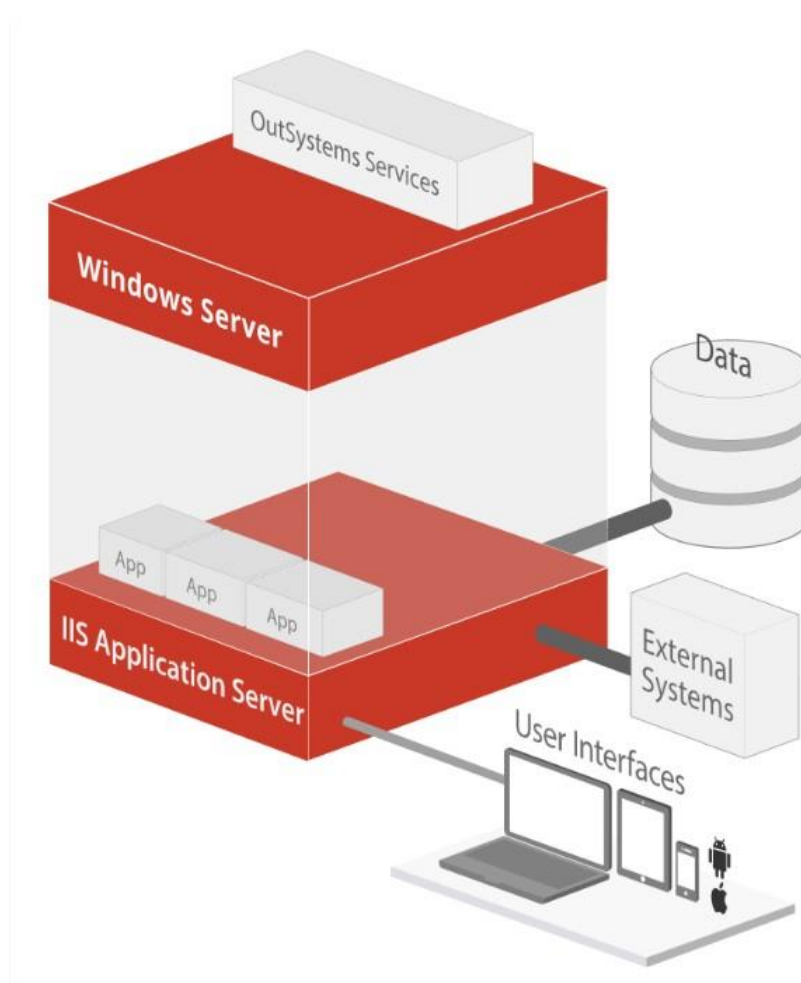


Figura 4.5: Arquitetura [3]

4.9 Boas Práticas em *OutSystems*

Existem inúmeras práticas que devem ser seguidas quando se desenvolve em *OutSystems*. São práticas que no início costumam a ser interiorizadas, mas que após praticar bastante são muito úteis tanto na leitura do código por parte de outras pessoas, quer ao nível de organização. No início deste estágio foram-me transmitidas boas práticas como, evitar *labels* sem descrições e comentar todo o tipo de lógica que seja pouco clara para facilitar a sua leitura.

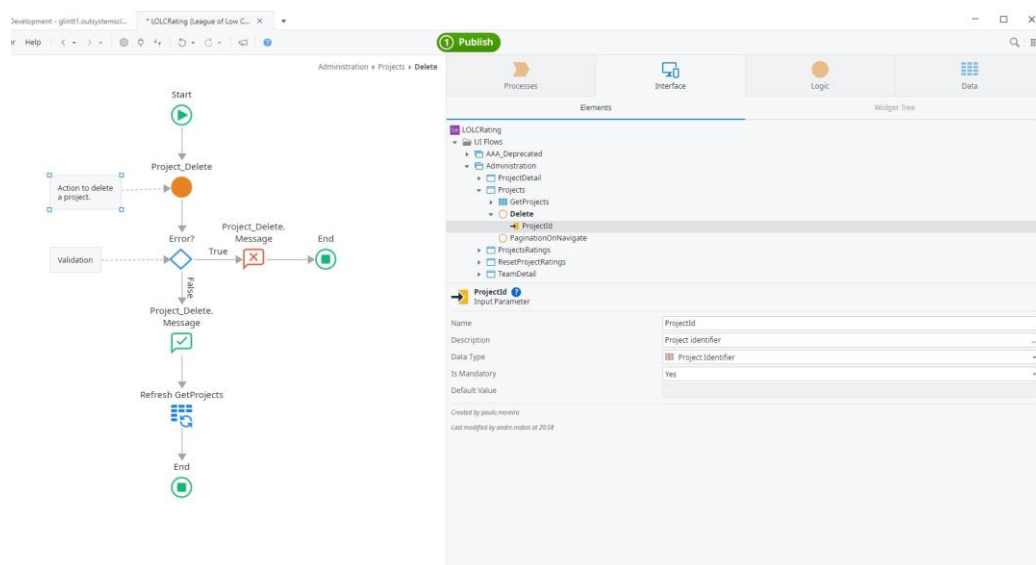


Figura 4.6: Exemplo de uma ação conforme as boas práticas de *OutSystems*

Como podemos ver na figura 4.6 a lógica está comentada e indentada, facilitando a sua leitura e perceção do código. Outra boa prática são também as descrições tanto na ação como nas suas variáveis.

Ao nível de base de dados, para além de se preencher a descrição de cada um dos atributos da entidade, esta não deve ultrapassar os 50 atributos. As chaves estrangeiras também devem ser sempre validadas, assim como a forma de apagar os dados das tabelas. Isolar atributos de texto complexo e dados binários em tabelas separadas e encriptar passwords na base de dados. Sempre que possível deve-se evitar a utilização das *queries* avançadas e trocar pelos agregados, por questão de performance da aplicação e boa prática.

4.10 Abas Principais no *Service Studio*

Processes

Nesta secção o utilizador pode efetuar operações assíncronas através da criação de processos e/ou ações de temporização (*Timers*). Estas operações permitem à aplicação executar certas ações de acordo com condições pré-definidas ou num intervalo temporal, por exemplo, uma vez por dia.

Interface

A *Interface* é onde se visualiza e se constrói o esboço visual da aplicação. Aqui é possível criar ecrãs e blocos, sendo que, um bloco pode ser posto dentro de um ecrã ou dentro de outro bloco, mas um ecrã não pode ser colocado dentro de um bloco. A ideia de construção de blocos é tornar uma parte visual de um ecrã reutilizável, se existir a necessidade de criar dois ecrãs em que uma dada parte é igual, em vez de definirmos a mesma parte duas vezes fazemos um bloco que inserimos nos em cada um dos ecrãs poupando tempo de desenvolvimento.

Logic

Esta aba contém algumas ações que são relativas à parte lógica, podendo também ter algumas ações como criar, eliminar e atualizar entidades (CRUD). Para definir lógica nesta plataforma faz-se uso de ações. Visualmente uma ação pode ser vista como um fluxograma, tornando a manipulação de dados mais apelativa e de fácil leitura.

Data

A aba Data tal como o nome indica é onde fica definida a base de dados que suporta a aplicação. Em *OutSystems*, existem dois tipos de tabelas que podem ser utilizados para desenvolver a base de dados, entidades e entidades estáticas. As entidades permitem a criação, atualização e eliminação de um registo durante a utilização da aplicação. Uma entidade é constituída por um identificador que funciona como chave, mas pode ter mais atributos de vários tipos. Para além do tipo identificador, um atributo pode ser definido em diferentes tipos de dados como inteiro, inteiro longo, texto, número de telefone, *email*, decimal, moeda, booleano, data, data e hora, tempo e binário.

Capítulo 5

Atividades desenvolvidas

No início do estágio comecei por estudar o novo paradigma de *OutSystems Reactive Web Developer*. Para tal visualizei todos os vídeos e realizei os respectivos exercícios do curso de *Reactive Web* que estão presentes no site da *OutSystems*. Com estes vídeos adquiri algum conhecimento como modelação de dados em que aprendi como criar entidades de Base de Dados e entidades estáticas, usadas maioritariamente para definição dos dados. Demonstrações sobre como criar essas entidades e como inicializar dados em entidades de planilhas do Excel. Na parte da lógica foi abordado como definir as ações e fluxos lógicos para implementar a lógica de negócios, incluindo os diferentes tipos de ações e como elas podem interagir entre si, assim como demonstrações sobre como criar caminhos condicionais e ciclos.

5.1 Variáveis

As variáveis podem ser de 3 tipos, variáveis de entrada, saída e locais. Uma variável de entrada permite fornecer dados a um elemento para uso posterior. O parâmetro de entrada fica disponível nesse elemento. Por exemplo, ao adicionar a variável de entrada a uma ação do cliente, esta pode fornecer um valor para esse parâmetro de entrada ao chamar a ação do cliente. Use o valor no fluxo lógico da ação do cliente, por exemplo, em expressões ou como parte dos valores da variável de entrada de outras chamadas. Estas variáveis estão

disponíveis em ações do servidor, ações de cliente, ecrãs e blocos, processos e outros. A variável de entrada “Weather”. Uma variável de saída permite retornar valores calculados de uma ação, processo ou elemento de fluxo de processo. Podemos também observar a variável de saída “Result”. Uma variável local existe apenas no escopo do elemento pai, por exemplo, um ecrã ou uma ação. Uma variável local só pode ser atribuída e usada localmente dentro desse escopo. As variáveis locais são destruídas quando a execução sai do escopo do elemento pai. A figura 5.1 a ação tem uma variável local “city”, que representa uma entidade.

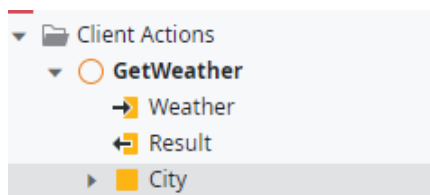


Figura 5.1: Exemplo de uma variável de entrada

5.2 Ações

As ações são uma parte muito importante da tecnologia OutSystems, com elas é possível implementar a maioria do código, desde desenvolver lógica de ecrãs, validações e *CRUD*. Estas ações vão variando dependendo do tipo de paradigma que se utiliza, por exemplo, em “Web traditional”, temos 3 tipos de ações: *Preparation*, ações de servidor e ações de ecrã. O *Preparation* é uma ação responsável por renderizar todo o tipo de conteúdo lógico do ecrã, por exemplo, carregar os dados de uma lista de contactos, neste caso é preciso carregar o respetivo agregado da base de dados, para ser apresentado no ecrã.

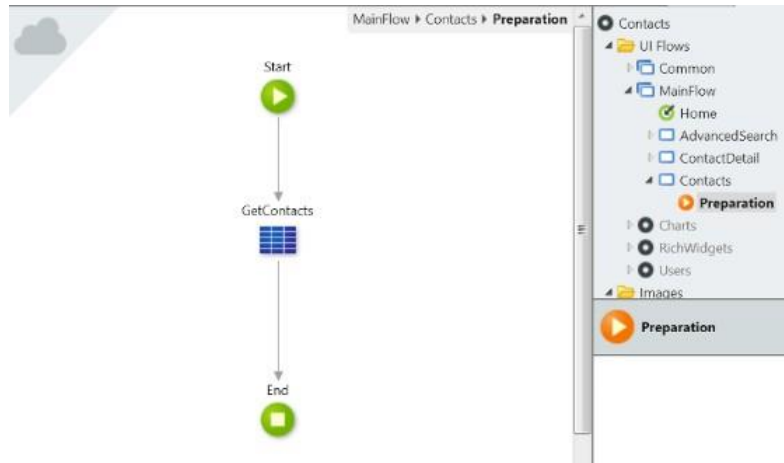


Figura 5.2: Exemplo de um preparation [4]

As ações de servidor, como o próprio nome indica, são ações executadas do lado do servidor visando fazer alterações da parte do servidor, como, por exemplo, guardar um determinado contacto na base de dados. Ação da figura 5.3 faz a validação do contacto introduzido.

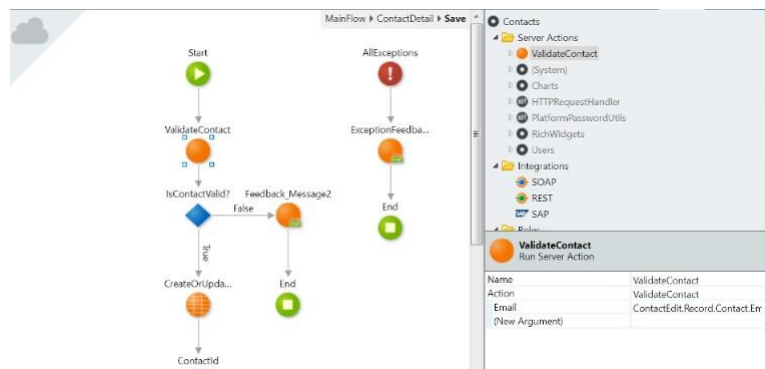


Figura 5.3: Exemplo de uma ação de Servidor [4]

As ações de ecrã permitem chamar elementos do ecrã como Botões ou Links, ou uma notificação de um bloco. As ações de ecrã têm escopo local para o ecrã com acesso a todos as variáveis de entrada do ecrã e *Widgets*.

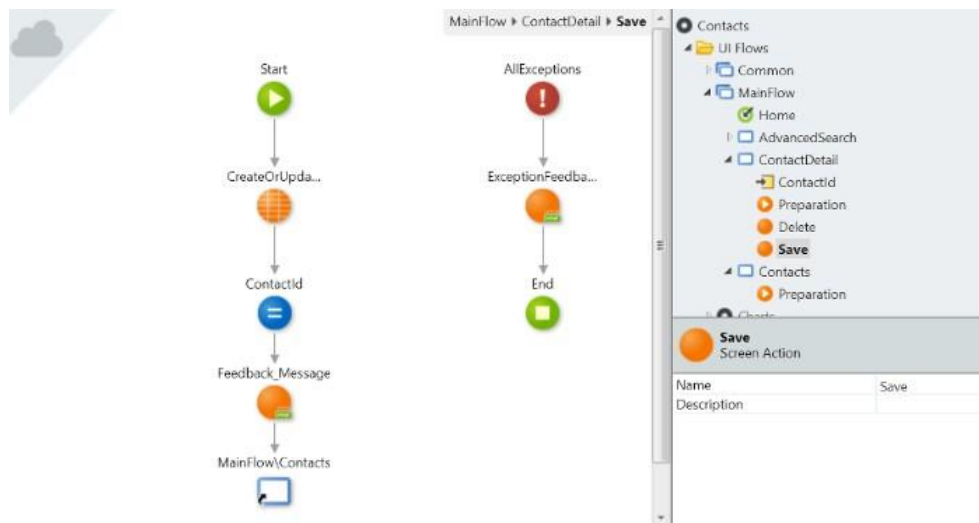


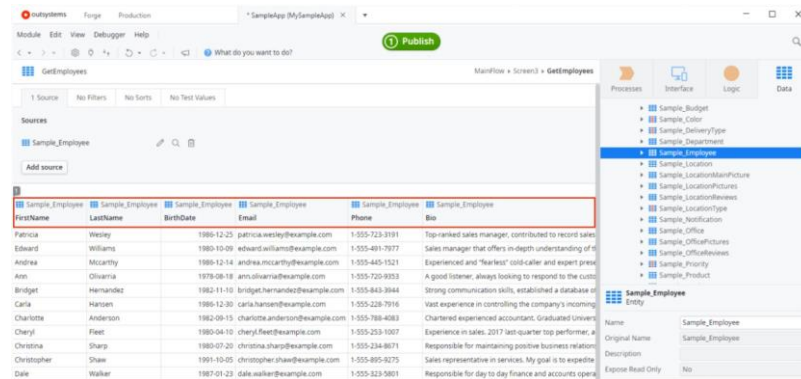
Figura 5.4: Exemplo de uma ação de ecrã [4]

A ação da figura 5.3 pretende guardar um contacto na base de dados, por um botão “Guardar”, que no que lhe concerne ao clicar nele vai executar esta ação.

No paradigma de “Reactive” e “Mobile”, as ações são ligeiramente diferentes, definindo-se em ações de cliente, ações de servidor e ações de dados. Ações de cliente são o equivalente móvel das ações do servidor. Eles são executados no dispositivo, não no servidor. As ações do cliente estão, portanto, disponíveis apenas para aplicativos móveis. As ações de dado são utilizadas para buscar dados complexos à base de dados ou sistemas externos após o carregamento do ecrã, estas ações são executadas no servidor. Ações de dados e agregações do lado do cliente começam a pesquisar dados simultaneamente após o carregamento da tela.

5.3 Agregados

Na parte dos agregados, percebi a sua definição e como usá-la para criar visualmente consultas simples à base de dados como também demonstrações sobre como criar um agregado, filtrar dados obtidos, classificar dados e testar agregados durante o desenvolvimento.



Sample_Employee	Sample_Employee	Sample_Employee	Sample_Employee	Sample_Employee	Sample_Employee
FirstName	LastName	BirthDate	Email	Phone	Bio
Patricia	Wesley	1980-12-25	patricia.wesley@example.com	1-555-723-3191	Top-ranked sales manager, contributed to record sales.
Edward	Williams	1980-10-09	edward.williams@example.com	1-555-491-7977	Sales manager that offers in-depth understanding of d
Andrea	McCarthy	1980-12-14	andrea.mccarthy@example.com	1-555-445-1521	Experienced and "fearless" cold-caller and expert pres
Ann	Olivaria	1978-08-18	ann.olivaria@example.com	1-555-720-9353	A good listener, always looking to respond to the custo
Bridget	Hernandez	1982-11-10	bridget.hernandez@example.com	1-555-843-3944	Strong communication skills, established a database of
Carla	Hansen	1980-12-30	carla.hansen@example.com	1-555-228-7916	Vast experience in controlling the company's incoming
Charlotte	Anderson	1982-09-15	charlotte.anderson@example.com	1-555-788-4083	Chartered experienced accountant, graduated Univer
Cheryl	Fleet	1980-04-10	cheryl.fleet@example.com	1-555-253-1007	Experience in sales, 2017 best-quarter top performer, a
Christina	Sharp	1980-07-20	christina.sharp@example.com	1-555-234-8071	Responsible for maintaining positive business relation
Christopher	Shaw	1991-10-05	christopher.shaw@example.com	1-555-895-9275	Sales representative in services, My goal is to expedite
Dale	Walker	1987-01-23	dale.walker@example.com	1-555-323-5801	Responsible for day to day finance and accounts opera

Figura 5.5: Exemplo de um agregado [5]

Na figura 5.1 podemos verificar o exemplo de um agregado que nada mais é que a junção de uma ou mais tabelas da base de dados para disponibilizar dados úteis a serem usados na aplicação. É também possível aplicar alguns filtros assim como ordenações nas tabelas. Normalmente os dados não são armazenados numa única Entidade. Assim, ao realizar consultas nos dados, é necessário combinar os registos armazenados nas várias entidades da base de dados. Estas junções de entidades podem ser de três tipos diferentes: “only with”, “with”, “with or without’.

A junção “only with”, visa juntar apenas os atributos que possuem valores correspondentes em ambas as tabelas. Este pode ser comparado ao “Inner join” utilizado em SQL.

Issue	Engineer	Engineer
Title	Name	PhoneNumber
Error message when trying to open a page	Jason Montgomery	555-345
Touch pad doesn't respond	John Fergusson	555-774
Cannot install Skype	John Fergusson	555-774
Error message when trying to save a document	Jonathan Fawakes	555-129
Pixel problem of the display	Luis Monteiro	555-734
Machine Slow	Luis Monteiro	555-734
No network connection	Rita Dante	555-934
Hardware request	Rita Dante	555-934
The image only show a red cross	Yami Na Chin	555-462

Figura 5.6: Junção only With [5]

A junção “with or without”, significa a ordem das entidades na condição de junção que faz diferença nas linhas retornadas. A ideia é recuperar todos os registos da primeira entidade e combinar as linhas da segunda entidade a eles. Portanto, se a ordem das entidades for trocada, os resultados diferirão, semelhante ao “Left join”.

Issue	Engineer	Engineer
Title	Name	PhoneNumber
Applications keep crashing		
How can I make a phone call out of Lync		
Drive is not accessible		
The window doesn't refresh		
Error message when trying to open a page	Jason Montgomery	555-345
Touch pad doesn't respond	John Fergusson	555-774
Cannot install Skype	John Fergusson	555-774
Error message when trying to save a document	Jonathan Fawakes	555-129
Machine Slow	Luis Monteiro	555-734
Pixel problem of the display	Luis Monteiro	555-734
Hardware request	Rita Dante	555-934
No network connection	Rita Dante	555-934
The image only show a red cross	Yami Na Chin	555-462

Issue	Engineer	Engineer
Title	Name	PhoneNumber
	Ben Stevens	555-967
	Ann Joshua	555-834
Cannot install Skype	John Fergusson	555-774
Error message when trying to open a page	Jason Montgomery	555-345
Error message when trying to save a document	Jonathan Fawakes	555-129
Hardware request	Rita Dante	555-934
Machine Slow	Luis Monteiro	555-734
No network connection	Rita Dante	555-934
Pixel problem of the display	Luis Monteiro	555-734
The image only show a red cross	Yami Na Chin	555-462
Touch pad doesn't respond	John Fergusson	555-774

Figura 5.7: Junção With or without [5]

Por fim, a junção “with”, junta todos os atributos das duas tabelas, assemelhando-se ao “Full join”.

Issue	Engineer	Engineer
Title	Name	PhoneNumber
Applications keep crashing		
Drive is not accessible		
How can I make a phone call out of Lync		
The window doesn't refresh		
	Ann Joshua	555-834
	Ben Stevens	555-967
Error message when trying to open a page	Jason Montgomery	555-345
Cannot install Skype	John Fergusson	555-774
Touch pad doesn't respond	John Fergusson	555-774
Error message when trying to save a document	Jonathan Fawakes	555-129
Machine Slow	Luis Monteiro	555-734
Pixel problem of the display	Luis Monteiro	555-734
Hardware request	Rita Dante	555-934
No network connection	Rita Dante	555-934
The image only show a red cross	Yami Na Chin	555-462

Figura 5.8: Junção With [5]

5.3.1 *Debug*

Um dos pontos muito importantes deste curso da *OutSystems* foi quando aprendi a fazer “debug” na plataforma, pois quando temos um problema, uma grande parte das vezes esta é a solução e isso ajuda no processo de desenvolvimento. O tema blocos e eventos também é importante no mundo do *OutSystems*, pois estes fazem com que o código seja reutilizável em alguns casos facilitando o desenvolvimento do projeto. Um bloco nada mais é que uma *interface* reutilizável pelo utilizador em que podem ser inseridos alguns componentes, padrões ou até mesmo outros blocos. Um bloco pode ter variáveis locais, variáveis de entrada, ações de ecrã, *placeholders* e até mesmo eventos, tornando possível a comunicação para fora do bloco.

5.3.2 **Validação de formulários**

Quando os utilizadores inserem dados num formulário e clicam no botão de "submiter formulário", verifica-se que os dados são válidos. Se os dados parecerem válidos, a aplicação poderá continuar. caso contrário, é boa prática informar os utilizadores quais campos contêm os erros e como corrigi-los.

Por exemplo, se o utilizador no campo da quantidade tentar introduzir um valor que não corresponda a um inteiro, então aparece uma mensagem a indicar o tipo de valor da variável e uma mensagem que avisa para validar os campos.

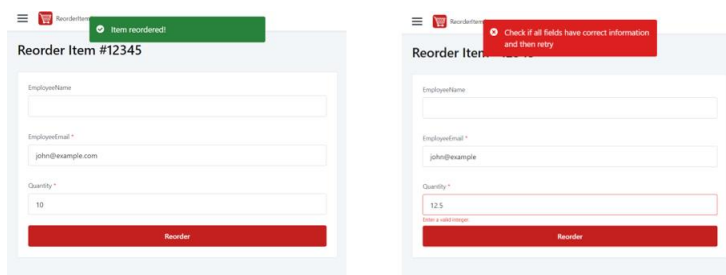


Figura 5.9: Validação de formulários [6]

5.3.3 Modelação de dados

As informações ou os dados de cada aplicação representam vários conceitos de negócios. Em *OutSystems* estes conceitos são modelados e denominados como entidades, estas entidades agregam todos os dados da aplicação. Na imagem representada abaixo, podemos verificar as principais diferenças entre as entidades e *OutSystems* e uma tabela de base de dados tradicional.

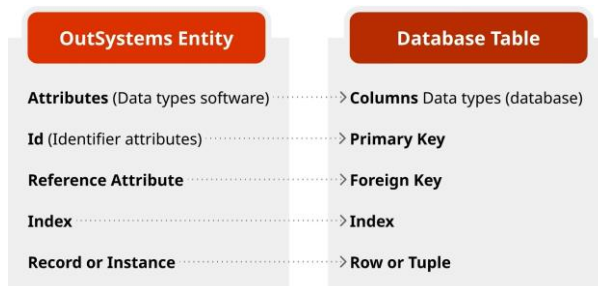


Figura 5.10: Comparação entre entidade e uma tabela de base de dados

5.4 Formações

5.4.1 Formação OutSystems Reactive Developer

A empresa proporcionou-me a integração numa formação intensiva do paradigma *Reactive* com a duração de uma semana. Nesta formação foram abordados temas como, uma visão geral do paradigma 'Reactive', modelo de dados, ações lógicas, 'SQL Queries', segurança, temas e estilos, listas e tabelas, validações etc. Aprofundei os meus conhecimentos neste novo paradigma e também realizei o desenvolvimento de aplicação para tonificar os novos conhecimentos. O objetivo principal desta formação seria depois conseguir a aprovação do Certificado 'Associative Reactive Developer'. Este certificado consiste numa prova de escolha múltipla com uma estrutura de 50 perguntas e duração de até 2 horas. Para conseguir a aprovação é necessário obter 70%, que se traduz num acerto de 35 em 50 perguntas.

Após conseguir a aprovação do certificado, iria integrar numa equipa e desenvolver projetos para clientes.



André Matos

has successfully completed the certification process to be
officially recognized as



**Associate Reactive
Developer**



Figura 5.11: Associate Reactive Developer

5.4.2 Formação de OutSystems Mobile specialist

A certificação Associate Reactive Developer é um pré-requisito para fazer o exame Mobile Developer Specialization. Esta certificação baseia-se em tópicos específicos do desenvolvimento de aplicações móveis OutSystems, como armazenamento local, aplicações offline, sincronização de dados, plugins móveis e desenvolvimento de aplicações nativas. Para conseguir a aprovação ao certificado é necessário ter aprovação mínima de 70%, que se traduz num acerto de 14 em 20 perguntas com a duração do exame até 1 hora. O exame concentra-se em conhecimentos práticos e cenários que validam o conhecimento e experiência profissional de quem efetua o exame.

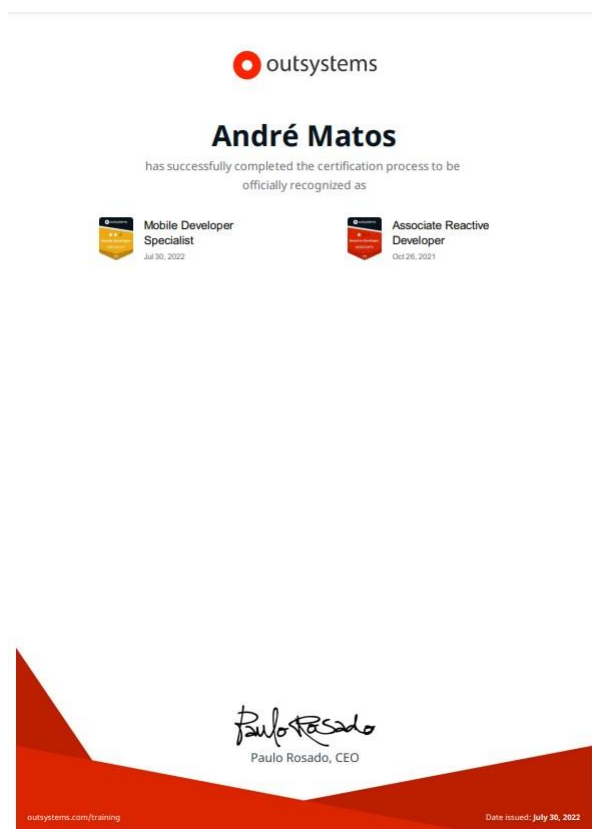


Figura 5.12: Mobile Developer Specialization

5.4.3 Formação de Base de Dados

Esta formação teve a duração de um dia e permitiu-me consolidar alguns conhecimentos em base de dados. A formação abordou temas como um sistema de gestão de base de dados relacional, as respetivas restrições, chaves primárias, chaves estrangeiras, criação de tabelas, queries avançadas, operações em SQL, recursividade, 'views', 'CTE(common table expression)', entre outros. Esta formação foi bastante pertinente, pois apesar de ter conhecimento sobre Modelação de dados e 'SQL' permitiu-me conhecer algumas áreas que desconhecia e que certamente me vão ser úteis no futuro.

5.4.4 Formação de Soft Skills

A empresa também proporcionou a escolha de quatro formações para o meu desenvolvimento pessoal e profissional. Os temas escolhidos foram: "Gestão de Stress", "Pensamento crítico e criativo", "Gestão de Tempo", "Liderança e trabalho em equipa". Na minha opinião estas formações acrescentaram valor ao meu dia a dia profissional, pois consegui conciliar as tarefas diárias mais fluida e geri-las de forma menos conturbada.

5.5 Demonstração do trabalho efetuado

Ao longo deste percurso, fiz parte de algumas equipas e por conseguinte projetos diferentes. Nesta secção, por questões legais, alguns dos projetos mencionados em baixo, devido ao acordo de confidencialidade com os clientes, estes projetos não serão especificados ao máximo.

5.5.1 Projeto da Avaliação da Liga do Low Code

A Liga do Low Code, consiste numa competição dentro da empresa que reúne algumas equipas e cada equipa vai desenvolver uma aplicação. Este projeto surgiu da necessidade de fazer a avaliação de cada um dos projetos de forma mais fácil e intuitiva.

O que foi proposto foi desenvolver uma aplicação em que seja possível os júris efetuar a avaliação de cada umas equipas baseada em determinados parâmetros de cada um dos projetos. Esta aplicação apresentaria depois uma avaliação final para cada um dos projetos e determinaria o vencedor daquela competição. O projeto foi desenvolvido na tecnologia “Reactive Web Applications” sendo desenvolvido por mim, acompanhado por uma pessoa com muita experiência na área para me guiar e indicar o melhor caminho a seguir.

Ecrãs da Aplicação

Nesta secção encontram-se alguns dos ecrãs da aplicação

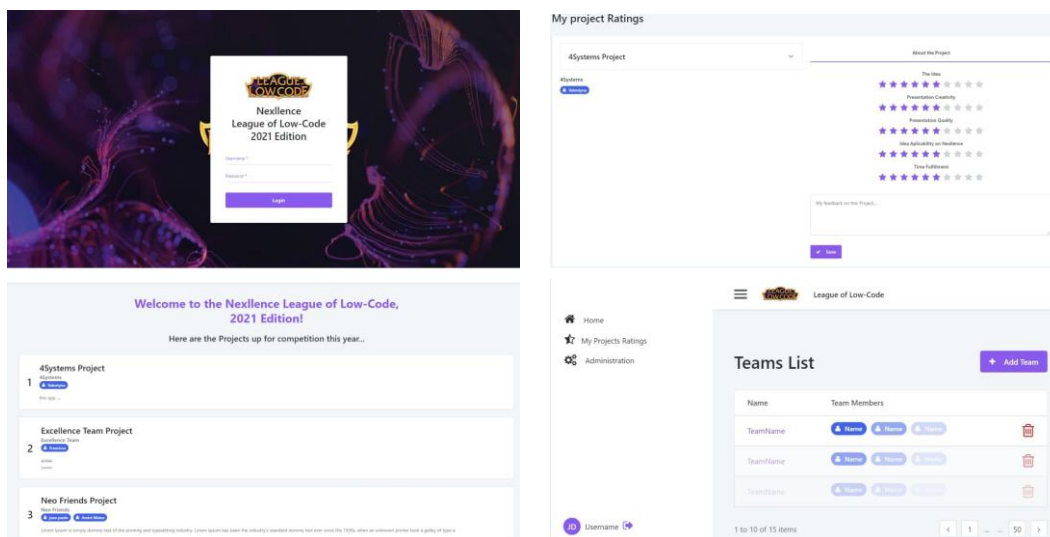


Figura 5.13: Ecrãs da Aplicação

Base de dados

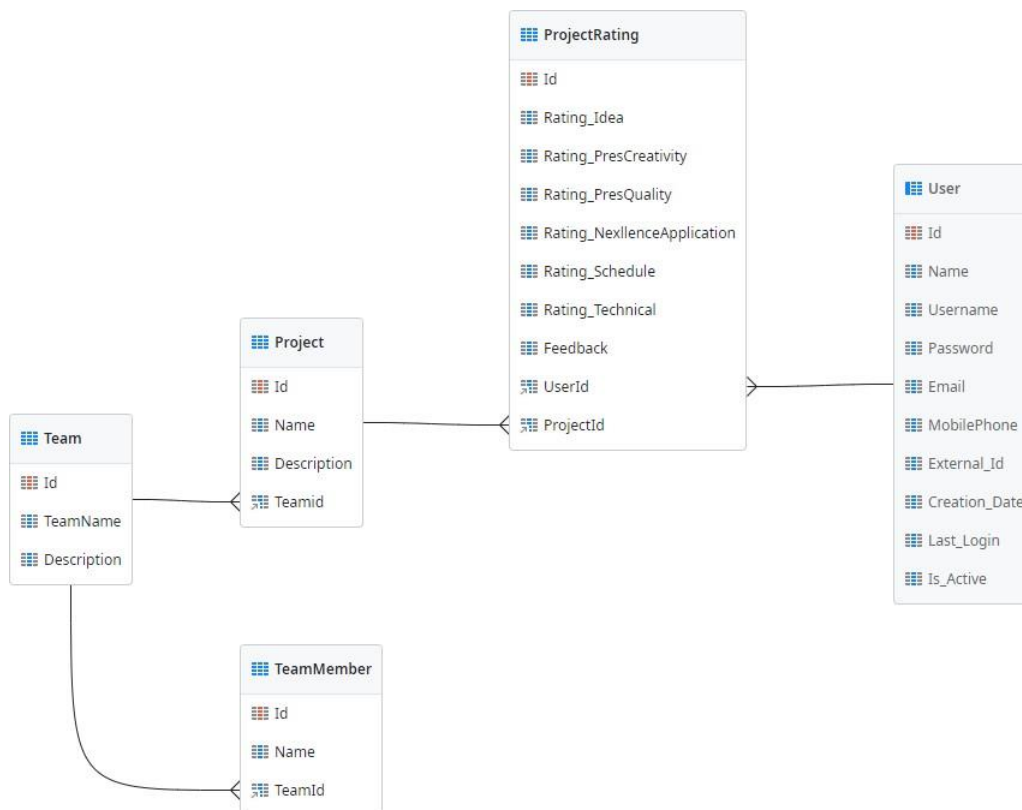


Figura 5.14: Base de dados da Aplicação

A base de dados da figura 5.14 serviu como base de dados da aplicação. Em que cada equipa era constituída por um ou mais elementos e cada equipa poderia ter um ou mais projetos. Depois os júris iam avaliar cada um dos projetos da equipa, o projeto com a melhor avaliação tendo em conta os vários parâmetros seria o vencedor da edição da liga Low-Code.

5.5.2 Projeto Administração dos Portos

Este projeto surgiu da necessidade de controlar e faturar o consumo de energia através das instalações existentes. As estratégias adotadas seriam a redução das emissões de poluentes e a facilidade ao acesso à água potável durante as viagens. O projeto tinha reuniões diárias para falar sobre o trabalho realizado, a realizar e eventuais dúvidas. Para além de ser usado o "teams" para as reuniões diárias, também se utilizou o Jira para simplificar o processo de desenvolvimento da aplicação e distribuição das tarefas da mesma. O objetivo seria desenvolver uma aplicação "Mobile" em que facilitasse ao mestre da embarcação o acesso tanto a água como a eletricidade.

A realização de um "Back-Office" também foi desenvolvida no paradigma "Reactive Web" para facilitar o manuseamento de dados. Consegui dar o meu contributo para realizar algumas das tarefas propostas. Em suma foi uma experiência bastante importante, pois absorvi bastante conhecimento da equipa em que estava integrado, como as boas práticas usadas na tecnologia, de como gerir o meu tempo para tentar entregar uma tarefa e sobretudo de como trabalhar em equipa.

5.5.3 Processamento de honorários médicos

O Portal de Honorários Clínicos consiste numa solução integrada que uniformiza toda a informação e regras dos Honorários Médicos. No Portal é possível configurar as regras de cálculo, registar a atividade dos profissionais de saúde e, por um motor de regras, efetuar o processamento e respetivo cálculo dos Honorários Clínicos. Esta aplicação foi desenvolvida no paradigma "Web Traditional", uma área em que não tinha muito conhecimento e em que este projeto foi bastante útil para o aprimorar. Em suma foi uma experiência interessante com bastantes dificuldades pelo meio, no entanto, este projeto fez com que crescesse a nível profissional e a nível pessoal, pelos valores que a equipa me foi passando ao longo do tempo.

Capítulo 6

Estágio

6.1 Objetivos do Estágio

O presente estágio profissional surgiu devido à vontade de ganhar experiência no mercado de trabalho na área de OutSystems. Depois da conclusão da Unidade Extracurricular de *Outsystems* no Instituto Politécnico de Bragança, ganhei o gosto pela área e decidi candidatar-me à academia da Glintt. Os objetivos deste estágio são adquirir o máximo possível de conhecimento nos diversos paradigmas de *Outsystems* como Web, Mobile e Reactive e integrar projetos de onde possa contribuir ativamente e absorver o melhor possível todos os conhecimentos transmitidos pelos colegas.

6.2 Diário do Estágio

Ao longo do estágio registei as tarefas feitas semanalmente para facilitar este diário. O trabalho foi realizado desde dezoito de Outubro a quatro de maio, proferindo oitocentas e quarenta e oito horas de trabalho, cerca de cento e seis dias de trabalho úteis.

Tabela 6.1: Tarefas Projeto da liga Low-Code

Tarefa	Descrição da Tarefa
Página do projeto	<ul style="list-style-type: none"> • Desenvolvimento da listagem dos projetos. • Mockups do ecrã. • Criação das ações lógicas para eliminar projetos. • Paginação e ordenação da listagem.
Página de detalhe do projeto	<ul style="list-style-type: none"> • Mockups do ecrã. • Criação do formulário para criar um projeto novo. • Criação das ações lógicas para criar/editar projetos. •
Página da Equipa	<ul style="list-style-type: none"> • Desenvolvimento da listagem das equipas. • <i>Mockups</i> dos ecrãs. • Criação das ações lógicas para eliminar projetos. • Paginação e ordenação da listagem. • Criação de web block com os membros das equipas, para reutilizar noutros ecrãs • Links para entrar no detalhe da equipa a partir da listagem das equipas.
Página de detalhe da equipa	<ul style="list-style-type: none"> • <i>Mockups</i> dos ecrãs. • Criação do formulário para criar uma equipa nova. • Criação das ações lógicas para criar/editar projetos. • Adicionar membros à equipa, a partir do detalhe da mesma

Tabela 6.2: Tarefas Projeto da liga Low-Code

Tarefa	Descrição da Tarefa
Página de detalhe da equipa	<ul style="list-style-type: none"> • <i>Mockups</i> dos ecrãs. • Criação do formulário para criar uma equipa nova. • Criação das ações lógicas para criar/editar projetos. • Adicionar membros à equipa, a partir do detalhe da mesma
Página de avaliação dos projetos	<ul style="list-style-type: none"> • <i>Mockups</i> dos ecrãs. • Criação de acordeões com cada uma das equipas da liga. • Permite exportar para Excel as avaliações. • Eliminar avaliações.

Tabela 6.3: Tarefas do Projeto Administração dos Portos

Tarefa	Descrição da Tarefa
Tarefas realizadas	<ul style="list-style-type: none"> • Criação de entidades e atributos • Desenvolvimento de <i>CRUD</i>. • Paginação e ordenação de listagens. • Criação de entidades e atributos • Desenvolvimento de <i>queries</i> avançadas • Desenvolvimento de CSS • Correção de erros • Criação de lógica avançada. • Desenvolvimento de ecrãs. • Aplicação de filtros aos agregados.

Tabela 6.4: Tarefas Projeto Honorários clínicos

Tarefa	Descrição da Tarefa
Tarefas realizadas	<ul style="list-style-type: none"> • Criação de entidades e atributos • Desenvolvimento de <i>CRUD</i>. • Paginação e ordenação de listagens. • Desenvolvimento de pop-ups. • Criação de entidades e atributos • Desenvolvimento de <i>queries</i> avançadas • Desenvolvimento de CSS • Correção de erros • Desenvolvimento de timers. • Criação de lógica avançada. • Desenvolvimento de ecrãs. • Aplicação de filtros aos agregados.

Capítulo 7

Conclusões

7.1 Contributos do Estágio Profissional

Ao longo do período do presente estágio cumprido na empresa GLINTT, foi possível adquirir conhecimento substancial na área de *OutSystems*. Este foi importante para o desenvolvimento de competências profissionais e pessoais. A evolução em termos pessoais e profissionais é notória desde o começo do estágio. O contacto com profissionais de excelência fez aprimorar os meus conhecimentos em termos técnicos e a nível pessoal e é nesse sentido que quero continuar. o estágio contribuiu para perceber o mundo empresarial a partir do seu interior, serviu também para adquirir a responsabilidade que está no cumprimento de prazos relativos tanto a formação como a projetos na sua análise e desenvolvimento. Em relação aos paradigmas de *Outsystems* “Traditional Web”, “Reactive Web”, “Mobile”, posso afirmar que o conhecimento nestes paradigmas solidificaram bastante com este estágio, no entanto, ainda há um longo caminho pela frente e certamente muito a aprender.

Bibliografia

- [1] URL: https://success.outsystems.com/Documentation/11/Getting_started/Service_Studio_Overview.
- [2] URL: https://success.outsystems.com/Documentation/11/Getting_started/Service_Studio_Overview.
- [3] URL: https://success.outsystems.com/Documentation/11/Setup_and_maintain_your_OutSystems_infrastructure/Setting_Up_OutSystems/Possible_setups_for_an_OutSystems_infrastructure/Infrastructure_architecture_and_deployment_options.
- [4] URL: https://success.outsystems.com/Documentation/11/Developing_an_Application/Implement_Application_Logic/Actions_in_Web_Applications.
- [5] URL: https://success.outsystems.com/Documentation/11/Reference/OutSystems_Language/Data/Handling_Data/Queries/Supported_Join_Types.
- [6] URL: https://success.outsystems.com/Documentation/11/Developing_an_Application/Design_UI/Forms/Validate_the_fields_of_a_form.