PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DESENVOLVIMENTO LOW-CODE: ESTUDO DE CASO COM PLATAFORMA OUTSYSTEMS - APLICAÇÃO SEVEN SHOP

MÁRCIO ANTUSA DA COSTA

GOIÂNIA
2023

MÁRCIO ANTUSA DA COSTA

DESENVOLVIMENTO LOW-CODE: ESTUDO DE CASO COM PLATAFORMA OUTSYSTEMS - APLICAÇÃO SEVEN SHOP

Trabalho de conclusão de curso, apresentado à Escola Politécnica e de Artes da Pontifícia Universidade Católica de Goiás.

Orientador: Prof. Eugênio Júlio Messala Cândido Carvalho.

GOIÂNIA

2023

MÁRCIO ANTUSA DA COSTA

DESENVOLVIMENTO LOW-CODE: ESTUDO DE CASO COM PLATAFORMA OUTSYSTEMS - APLICAÇÃO SEVEN SHOP

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharel em Ciência de Computação, em _____/_____/_____.

Profa. Ma. Ludmilla Reis Pinheiro dos Santos
Coordenadora de Trabalho de Conclusão de Curso

Banca examinadora:

Orientador(a): Me. Eugênio Júlio Messala Cândido Carvalho

Prof. Dr. Leonardo Guerra de Rezende Guedes

Prof. Me. Olegário Correa da Silva Neto

GOIÂNIA
2023

AGRADECIMENTOS

Meus sinceros agradecimentos a todos que me apoiaram durante a realização deste trabalho acadêmico, à minha família, especialmente à minha mãe, que sempre me apoiou e acreditou em mim, ao Matheus Barbosa de Sousa pelo apoio e amizade, aos meus colegas e amigos: Guilherme Henrique Mendonça Nascente e Watson Guimarães Soares, que me acompanham nessa jornada desde o início do curso.

Um agradecimento também a todos os professores que contribuíram para a minha formação, especialmente o Prof. Eugênio Júlio Messala Cândido Carvalho, pela orientação e auxílio, o Prof. Alexandre Ribeiro por me incentivar a sempre dar o meu melhor e à professora Carmen Cecilia Centeno, que sempre esteve disposta a me auxiliar no que fosse necessário. E, sobretudo, agradeço a Deus por tornar tudo isso possível.

A todos vocês, que fizeram parte desta etapa da minha vida, obrigado!

RESUMO

Este trabalho apresenta um estudo sobre a abordagem de desenvolvimento *low-code*, com ênfase em uma das principais plataformas da atualidade: a *OutSystems*. O estudo inclui uma revisão das principais abordagens de desenvolvimento, explorando suas vantagens e características. Além disso, fornece uma análise detalhada da plataforma *OutSystems*, abordando seus aspectos, funcionalidades e vantagens para o desenvolvimento rápido de aplicações empresariais. São avaliadas as capacidades de desenvolvimento visual, integração de dados e sistemas, além da escalabilidade e suporte oferecidos pela plataforma. Por fim, como parte prática do estudo de caso, foi desenvolvida uma aplicação utilizando a plataforma *OutSystems* para demonstrar o funcionamento do *low-code* em um cenário realista, ilustrando os benefícios dessa abordagem, como a facilidade de criação de interfaces de usuário, a agilidade no desenvolvimento e a capacidade de integração com outros sistemas.

Palavras-chaves: *Low-code, OutSystems*.

ABSTRACT

This work presents a study on the low-code development approach, with emphasis on one of the main platforms today: OutSystems. The study includes a review of the main development approaches, exploring their advantages and characteristics. In addition, it provides a detailed analysis of the OutSystems platform, addressing its aspects, functionalities and advantages for the rapid development of business applications. Visual development capabilities, data and systems integration, as well as scalability and support offered by the platform are evaluated. Finally, as a practical part of the case study, an application was developed using the OutSystems platform to demonstrate how low-code works in a realistic scenario, illustrating the benefits of this approach, such as the ease of creating user interfaces, agility in development and the ability to integrate with other systems.

Keywords: Low-code, OutSystems.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O uso crescente da internet e das tecnologias digitais na indústria global impulsionou a transformação digital, conhecida como a quarta revolução industrial. Essa transformação refere-se à rápida adoção da tecnologia por uma sociedade cada vez mais digitalizada. Ela ocorre de diversas maneiras, desde a interação entre as pessoas até a aquisição de produtos, o entretenimento e o meio comercial. Diante desse cenário, torna-se essencial que as empresas e organizações comerciais se adaptem a essa nova realidade, a fim de evitar tornarem-se irrelevantes e obsoletas perante seus concorrentes (Silva, 2018, p.4).

Com o aumento da competitividade, surge a necessidade de as empresas e organizações serem capazes de se adaptarem às mudanças e de fornecerem respostas rápidas e flexíveis. Isso é fundamental para garantir o futuro e o sucesso de uma empresa. É nesse contexto que este trabalho foi concebido, com o objetivo de apresentar uma abordagem alternativa de desenvolvimento, chamada *low-code*, que permite alcançar tais objetivos de forma eficiente.

1.1 Objetivos Gerais

- O principal objetivo deste trabalho é fornecer uma visão geral da abordagem de desenvolvimento *low-code*, que visa capacitar pessoas com pouco ou nenhuma experiência em programação a criarem suas próprias aplicações empresariais.

1.2 Objetivos Específicos

- Criar e desenvolver uma aplicação utilizando a plataforma *OutSystems* para exemplificar como funciona o desenvolvimento em uma plataforma low-code.

1.3 Justificativa

Como mencionado anteriormente, a sociedade está cada vez mais conectada à internet de todas as formas possíveis, o que é de fundamental importância para o setor comercial. Caso as empresas não se adaptem a essa transformação digital que está ocorrendo cada vez mais, correm o risco de ficarem ultrapassadas e obsoletas, o que resultaria em prejuízos. Com isso em mente, é importante mostrar e esclarecer para as pequenas e médias empresas que elas não precisam ter profissionais altamente qualificados para integrar seus negócios à internet, como

criar um site ou aplicação na web. Esse trabalho pode ser realizado por qualquer pessoa, utilizando uma plataforma que facilite o desenvolvimento, como a plataforma *low-code OutSystems*. Portanto, o objetivo deste trabalho é demonstrar e exemplificar que qualquer pessoa, mesmo sem conhecimento ou experiência em desenvolvimento, pode criar aplicativos com qualidade, utilizando *low-code*.

1.4 Metodologia

Este trabalho tem como objetivo utilizar a coleta de informações para explicar o conceito e o processo de desenvolvimento *low-code*, utilizando a plataforma *OutSystems* como exemplo, demonstrando e exemplificando o processo de criação de uma aplicação com essa tecnologia.

A metodologia empregada foi a seguinte:

- Pesquisa bibliográfica sobre as diferentes formas de desenvolvimento de *software* através de documentação de ferramentas e artigos na internet;
- Pesquisa bibliográfica sobre a plataforma *OutSystems* com base na documentação;
- Desenvolvimento de uma aplicação.

1.5 Estrutura do Trabalho

O trabalho está dividido em 6 capítulos, estruturados de forma a criar uma linha de raciocínio que auxilie na compreensão do conceito de *low-code*. A distribuição dos capítulos está da seguinte forma:

- **Capítulo 1:** Este capítulo consiste em uma introdução, na qual serão apresentados o tema abordado, a justificativa para a escolha do assunto e os objetivos gerais e específicos do trabalho.
- **Capítulo 2:** Traz os conceitos gerais das abordagens de desenvolvimento *high-code*, *no-code* e *low-code*. Explorando as características distintas de cada abordagem e como elas se relacionam com o processo de desenvolvimento de software.

- **Capítulo 3:** Este capítulo mergulha de forma aprofundada na metodologia *low-code*, fornecendo uma compreensão detalhada de seus princípios e técnicas de desenvolvimento.
- **Capítulo 4:** Apresenta uma fundamentação teórica da plataforma *OutSystems*, destacando seus principais aspectos e características, bem como seus principais recursos.
- **Capítulo 5:** Neste capítulo, é apresentada uma visão geral da aplicação desenvolvida para o estudo de caso, oferecendo uma compreensão abrangente do resultado final alcançado.
- **Capítulo 6:** Neste capítulo, serão abordadas as considerações finais, proporcionando uma conclusão ao trabalho desenvolvido.

2 ABORDAGENS DE DESENVOLVIMENTO

2.1 High-Code

A abordagem tradicional de desenvolvimento, chamada de *High-Code*, consiste em criar software personalizado através da escrita manual de código por desenvolvedores altamente capacitados e qualificados. Com essa abordagem, é possível criar uma ampla variedade de softwares altamente personalizados. Para isso, os desenvolvedores precisam ter conhecimento em diversas tecnologias, como linguagens de programação como Java, JavaScript e C#, bibliotecas e frameworks para o front-end, como React, Angular, além de estruturas específicas para aplicativos móveis e nativos, bem como banco de dados, tais como PostgreSQL e MySQL. Essa abordagem é ideal para uma experiência "orientada por código", permitindo que os desenvolvedores trabalhem de perto com os ciclos de implantação de código e possibilitem o desenvolvimento de qualquer tipo de software, desde aplicativos de negócios até jogos. (ABREU, 2022; BABYCH, 2022)

Embora a abordagem *High-Code* permita a implementação de algoritmos avançados e conectores personalizados entre sistemas, é importante ressaltar que essa abordagem geralmente tem um alto custo associado. Isso ocorre porque o desenvolvimento personalizado de software requer desenvolvedores altamente especializados e qualificados, que podem ser caros e precisam ser envolvidos em todas as fases do projeto. Além disso, como tudo é desenvolvido manualmente, pode levar tempo que poderia ser utilizado para atividades mais valiosas, como a criação de novos recursos e inovações. (ABREU, 2022)

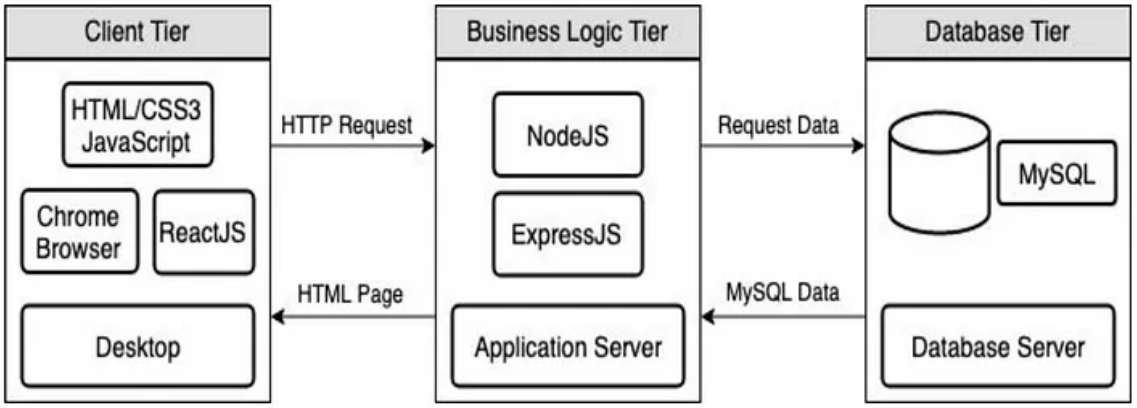
Embora a abordagem *High-Code* tenha suas vantagens para o desenvolvimento de software personalizado, é importante considerar cuidadosamente os custos envolvidos em termos de tempo, esforço e recursos necessários para sua implementação. É necessário avaliar se os benefícios oferecidos por essa abordagem compensam o investimento de recursos e se ela é adequada para as necessidades do projeto em questão. É importante equilibrar as necessidades do projeto com as restrições de recursos para escolher a abordagem de desenvolvimento de software mais adequada. (ABREU, 2022)

Exemplo: Envio de email, utilizando tecnologias *High-Code*.

Geralmente uma aplicação na abordagem *high-code* passa pelas seguintes etapas de desenvolvimento:

1. **Criar uma interface para o usuário:** Geralmente para obter os dados fornecidos pelo usuário, são criados formulários implementados utilizando diversas tecnologias, tais como HTML, CSS, frameworks e bibliotecas, como o ReactJS. Essas tecnologias são utilizadas para desenvolver a interface do usuário, também conhecida como a parte *front-end* da aplicação.
2. **Criar uma função de envio de *e-mail*:** Normalmente, essa parte é implementada no *back-end* da aplicação, onde uma *API* é criada para ser consumida no *front-end*. Essa *API* recebe os dados fornecidos pelo usuário, como o endereço de *e-mail* do destinatário e o conteúdo da mensagem, e os tratará, realizando validações antes de processá-los e enviar o *e-mail*. Para essa etapa, é comum o uso de *frameworks* e linguagens de programação, como *Java, JavaScript* e *Python*.
3. **Criar um banco de dados:** É necessário utilizar um banco de dados, como o *PostgreSQL, MySQL* e *MongoDB*, para armazenar os dados da aplicação, incluindo endereços de *e-mail* e informações pessoais dos usuários. Essa etapa é fundamental, pois os dados armazenados serão utilizados na validação da parte *back-end*, permitindo, por exemplo, a verificação de cadastro do usuário no servidor de *e-mail*.

Figura 1 - Camadas de uma aplicação high-code.



Fonte: NGUYEN (2019)

2.2 No-Code

A abordagem de desenvolvimento de software *No-Code* é uma forma de criar aplicativos, applets e outros elementos de software sem a necessidade de habilidades de programação. Ela é voltada para usuários de linha de negócios e oferece uma interface visual de arrastar e soltar para montar os elementos necessários para produzir o aplicativo desejado. (OUTSYSTEMS, [ca. 2022j])

As plataformas *No-Code* são projetadas para fornecer um ambiente de desenvolvimento visual simplificado e criar aplicativos simples. Essas ferramentas são fáceis de usar e oferecem simplicidade para que qualquer pessoa possa criar aplicativos rapidamente, independentemente de sua experiência em desenvolvimento. O objetivo dessas plataformas é permitir que usuários corporativos ou desenvolvedores cidadãos criem seus aplicativos, geralmente para casos de uso específicos, sem precisar esperar meses pela equipe de TI. Isso evita que a TI seja desviada de projetos mais críticos. (OUTSYSTEMS, [ca. 2022j])

Embora o termo "*No-Code*" seja um pouco impróprio, já que todo software requer código, as ferramentas sem código abstraem o desenvolvimento de software de uma linguagem de programação, incluindo lógica e sintaxe. Diferentes objetos, elementos e componentes são representados visualmente, e um usuário os organiza para construir um aplicativo. (OUTSYSTEMS, [ca. 2022j])

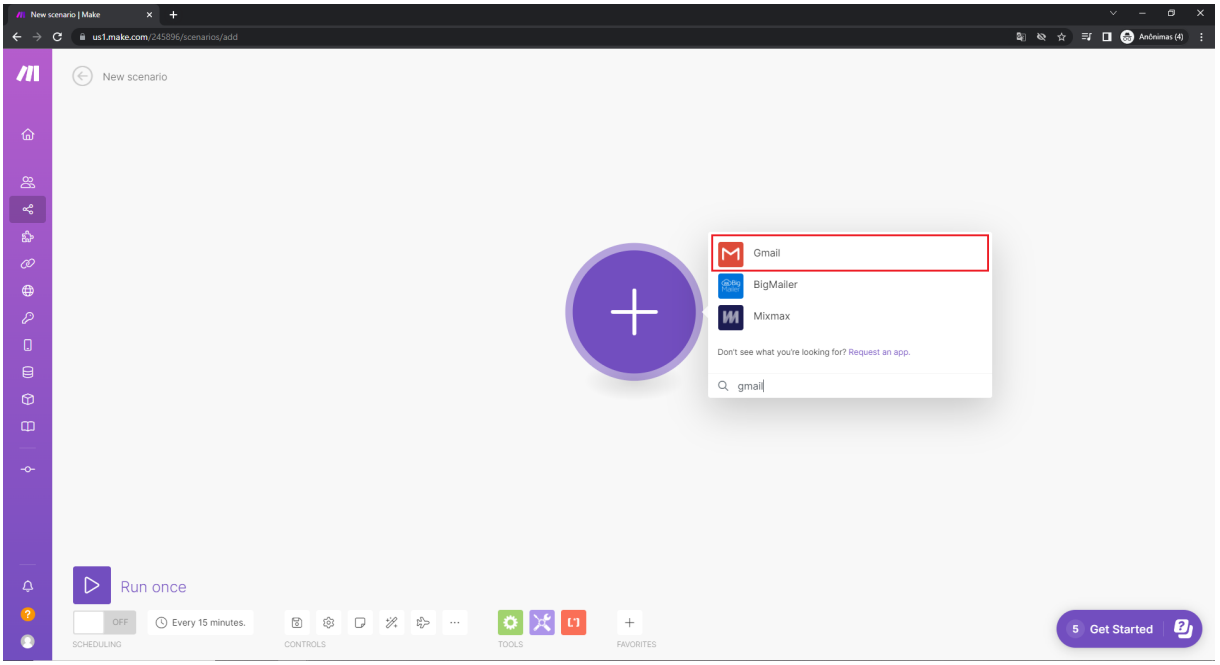
Em uma ferramenta *No-Code*, a abstração de dados oculta as instruções e detalhes subjacentes, exibindo apenas a funcionalidade necessária. Um aplicativo é representado por objetos visuais e recursos de mapeamento que mostram como vários elementos se interconectam. No final, *No-Code* é simplesmente uma camada de abstração sobre o código, permitindo que criadores criem aplicativos modernos de forma visual, sem precisar saber as linguagens de programação e tecnologias subjacentes, como HTML5, CSS e JavaScript. O Webflow é uma dessas plataformas que oferece todas essas funcionalidades de desenvolvimento sem código. (WEBFLOW, [ca. 2023])

Exemplo: Envio de *e-mail* na abordagem *No-Code*.

Neste exemplo foi utilizado a plataforma *no-code "Integromat"* para exemplificação, no entanto as etapas também são similares em outras plataformas:

1. Criar uma conta no site da plataforma.
2. Na página inicial clique em *"Create a new scenario"* para a criação do cenário da aplicação.
3. Na página que se abrirá, clique na opção com sinal de "+" para adicionar um módulo à aplicação, no caso deste exemplo será o módulo de *gmail*.

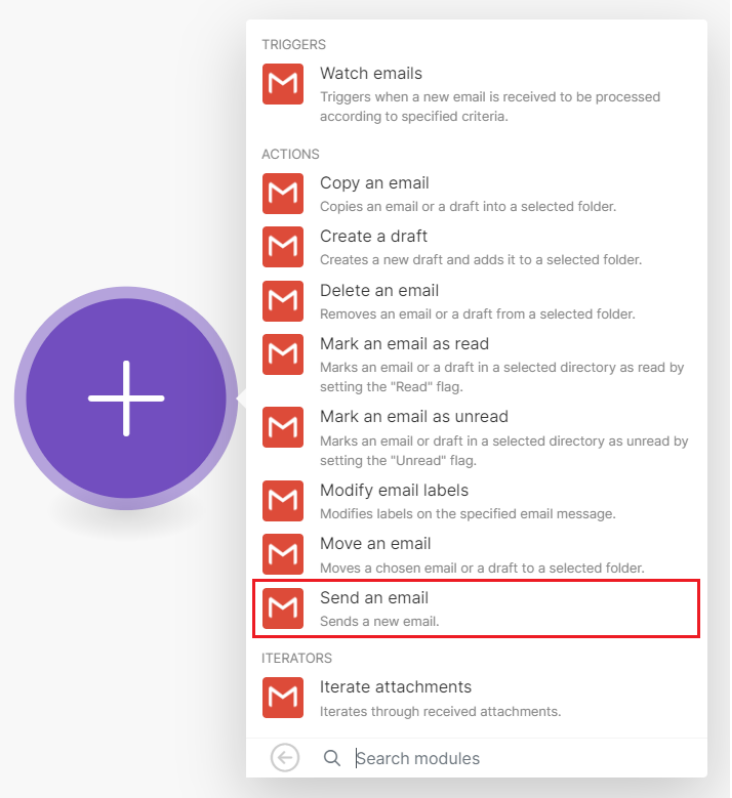
Figura 2 - Adição de módulo *"Gmail"*.



Fonte: Autoria Própria.

4. Após a adição do módulo, selecione *"Send an email"* da lista de opções.

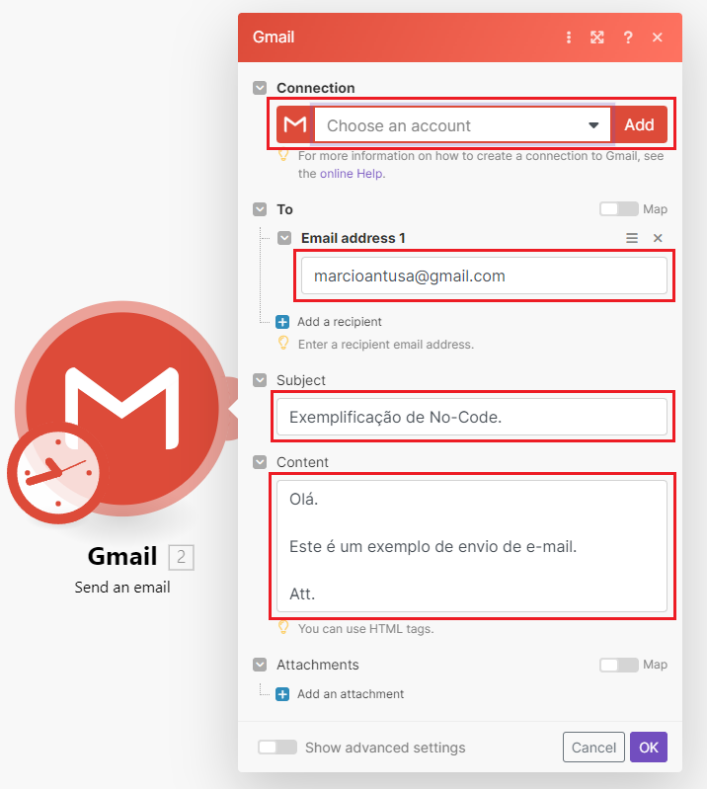
Figura 3 - Selecionando opção de envio de *"Email"*.



Fonte: Autoria Própria.

5. Clique no módulo criado e adicione o *email* do remetente, endereço de *email* do destinatário e conteúdo do *email*. Após isto clique em ok.

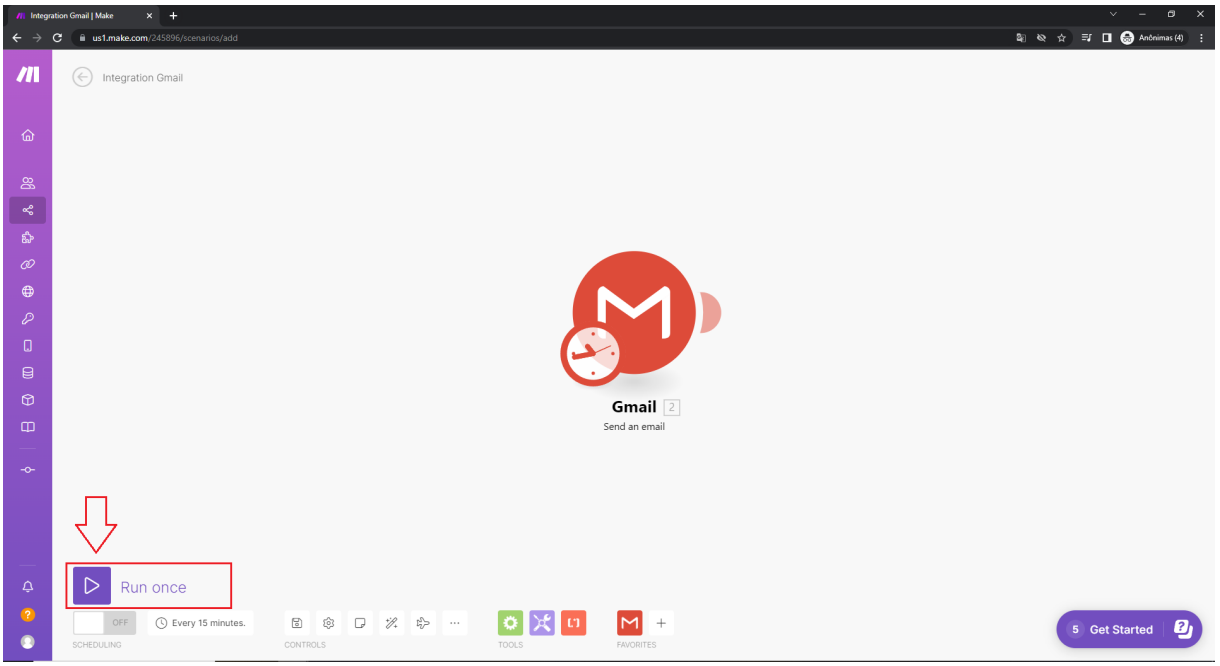
Figura 4 - Adicionando conteúdo do *"Email"*.



Fonte: Autoria Própria.

6. Por fim clique no botão "*Run once*" para publicar e executar a aplicação.

Figura 5 - Publicando e executando aplicação.



Fonte: Autoria Própria.

2.3 Low-Code

O termo "*Low-Code*" surgiu em 2014 em um relatório da Forrester Research que explorava novas plataformas de desenvolvimento de aplicativos orientadas para usuários. Desde então, o desenvolvimento *Low-Code* tem se expandido cada vez mais, especialmente no ambiente corporativo, graças à automação e simplificação progressiva de tecnologias que aumentam a produtividade no desenvolvimento de software e reduzem os custos, já que não há necessidade de programadores altamente qualificados. (IBERDROLA, [2022?])

As plataformas de *Low-Code* permitem que os usuários criem e mantenham aplicativos em diversos dispositivos e mídias digitais de maneira automática e intuitiva. A maioria dessas tecnologias é baseada em plataformas online, eliminando a necessidade de instalar ferramentas no computador do desenvolvedor, bastando acessar a plataforma pelo navegador. Além disso, o *Low-Code* permite que o aplicativo seja atualizado constantemente para atender às mudanças nas exigências dos usuários. (IBERDROLA, [2022?])

As plataformas de *Low-Code* são projetadas para serem altamente visuais e intuitivas, com uma interface modular que permite que os usuários arrastem e soltem elementos, usando uma ampla variedade de modelos de componentes. Essa abordagem simplifica a construção de fluxos de informação, apresentação de dados e automatização de processos, tornando tudo altamente versátil e atraente. (IBERDROLA, [2022?])

Embora o termo *Low-Code* seja relativamente novo, o conceito não é. Desenvolvedores avançados e cidadãos comuns usam ferramentas de baixo código há anos para criar aplicativos que atendem às necessidades específicas de negócios ou resolvem problemas específicos. Essas ferramentas tornam possível para qualquer pessoa, independentemente de sua experiência em programação, criar aplicativos de alta qualidade sem a necessidade de conhecimentos profundos de infraestrutura ou implementação de padrões que possam atrasá-los. (IBERDROLA, [2022?])

A utilização de plataformas de low-code é uma ferramenta valiosa para profissionais de TI e gestores empresariais, pois permite a entrega ágil e evolução constante do software. Com o uso dessas plataformas, os desenvolvedores são capazes de fornecer respostas rápidas e soluções personalizadas aos clientes, o que acelera significativamente o processo de desenvolvimento de software. Isso ocorre porque, em vez de escrever milhares de linhas de código e sintaxes complexas, os desenvolvedores podem simplificar o processo ao arrastar e soltar de componentes visuais para criar interfaces de usuário modernas, integrações, dados e lógica. Fazendo-se necessário escrever linhas de código, somente quando a plataforma

Low-Code não contenha alguma funcionalidade específica ou o desenvolvedor deseje criar algo que seja mais personalizado. (IBERDROLA, [2022?])

Ao permitir que os desenvolvedores se concentrem em tarefas de alto valor agregado, as plataformas de *Low-Code* podem aumentar a produtividade e reduzir os custos de desenvolvimento de software. Além disso, essas plataformas podem oferecer uma vantagem competitiva às empresas, permitindo que elas desenvolvam e implementem soluções mais rapidamente do que seus concorrentes. (IBERDROLA, [2022?])

Exemplo: Envio de *email* utilizando a abordagem *Low-Code*.

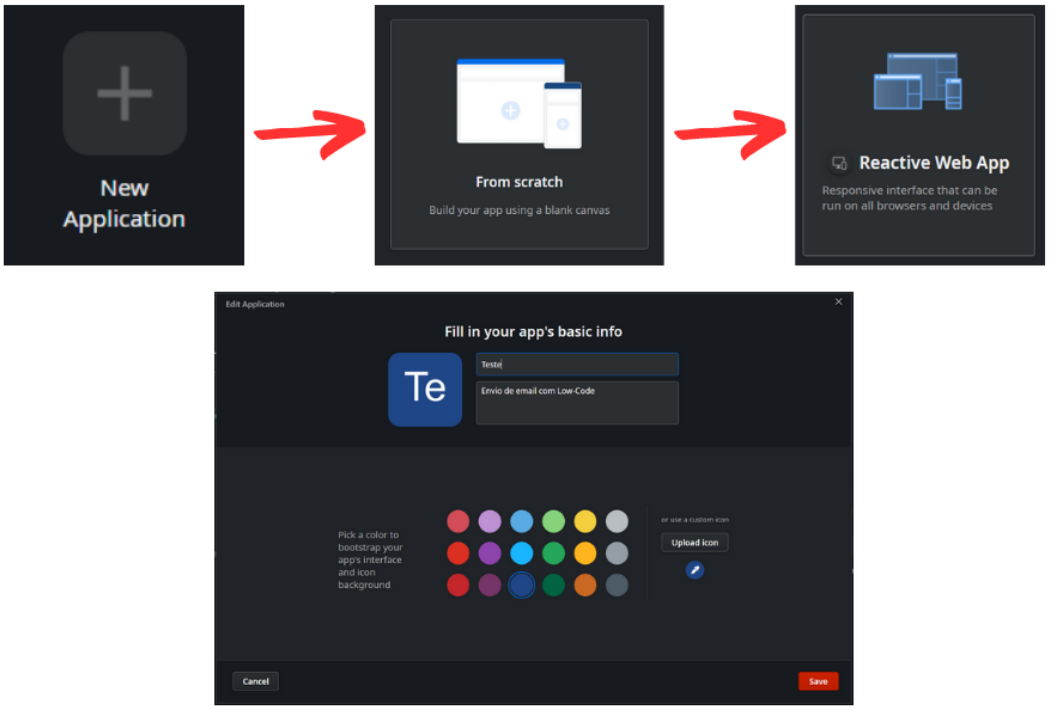
O processo de desenvolvimento em uma plataforma *low-code*, como a *OutSystems*, pode ser exemplificado pelos seguintes passos:

1. Criar uma conta no site da plataforma.
2. Fazer download do ambiente de desenvolvimento *"Service Studio"*
3. Abrir o ambiente de desenvolvimento, fazer login e criar uma nova aplicação. Para este exemplo será um *"Reactive Web App"*.

   Na página inicial faça os seguintes passos:

   1. Clique em "*New Application*"
   2. "*From Scratch*"
   3. "*Reactive Web App*"
   4. Defina um nome e descrição para a aplicação
   5. Finalizar clicando em "*Create App*"
   6. Adicione um módulo para começar o desenvolvimento.

Figura 6 - Criando aplicação *"Reactive Web"*.



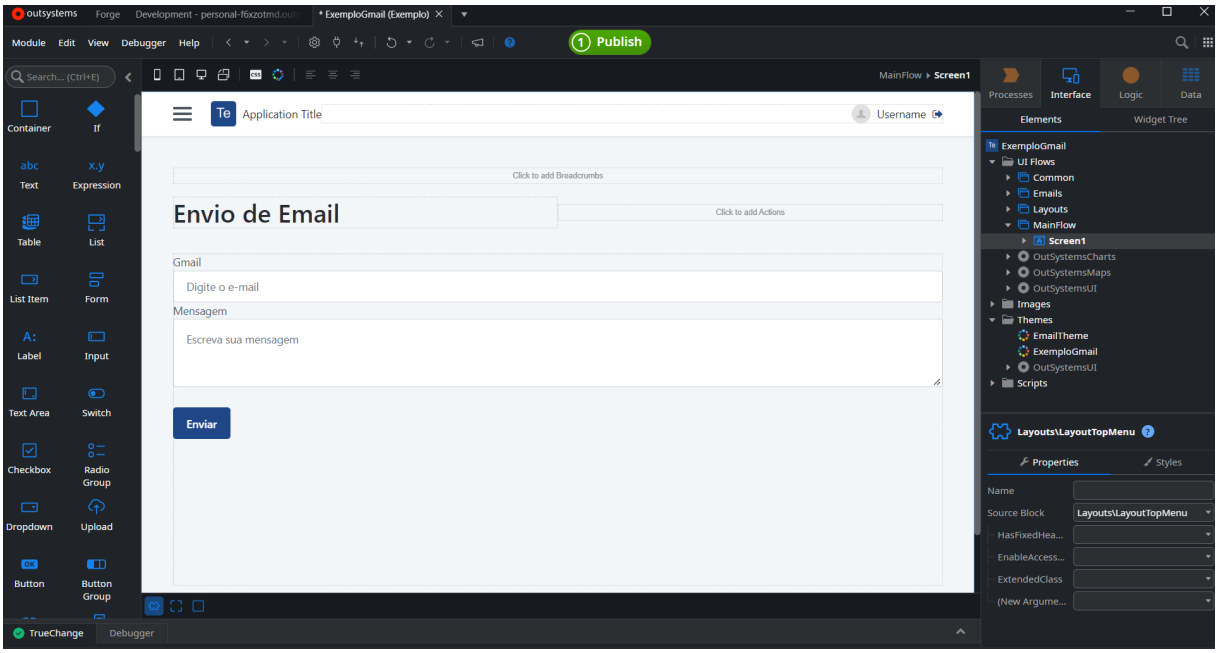
Fonte: Autoria Própria.

4. Criar um fluxo de IU de e-mail

   Isso pode ser feito com os seguintes passos:

   1. Cria uma página em branco arrastando um componente *"Screen"* da coluna lateral esquerda para a tela.
   2. Defina um título e crie os campos de entradas para os dados do usuário. Criando um *input* para o endereço e conteúdo do *e-mail*.
   3. Após isso crie um botão "Enviar" para acionar a ação de enviar *e-mail*.

Figura 7 - Criando interface de usuário.



Fonte: Autoria Própria.

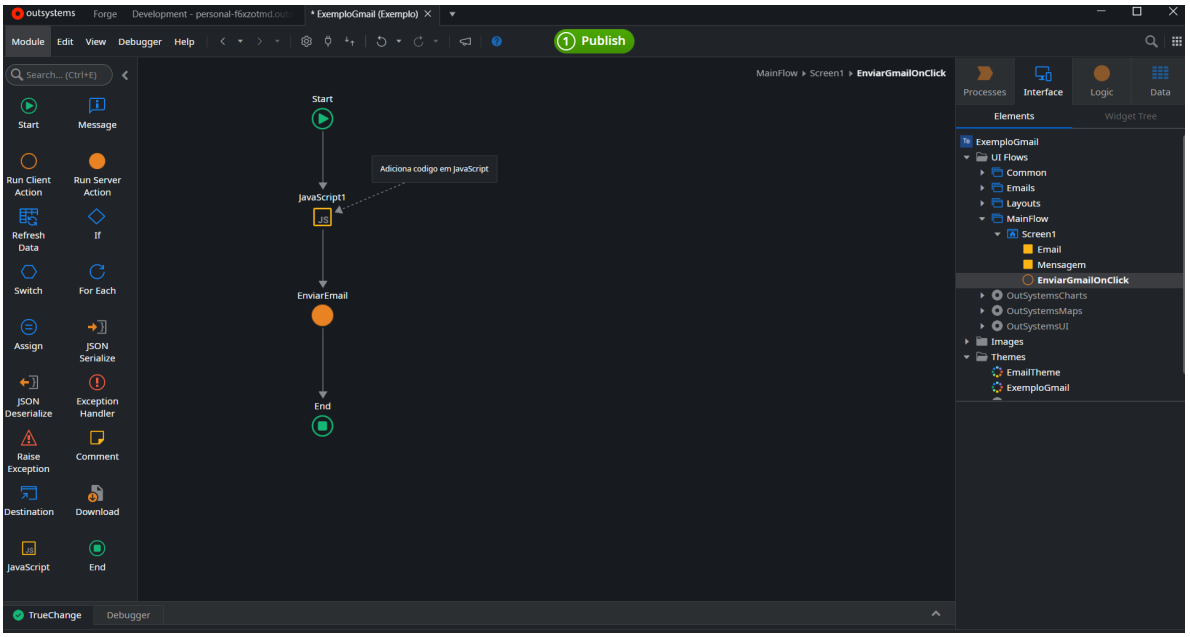
5. Criar lógica da aplicação

Para o envio do email é necessário criar uma *"server action"* que contenha a lógica necessária para o envio do *e-mail* após clicar no botão "Enviar".

Isso pode ser feito da seguinte forma:

1. Dando um duplo clique no botão para criar evento *"OnClick"* e definir o fluxo lógico da ação do botão, arrastando componentes da coluna do lado esquerdo para a tela. Para este exemplo foi criado um fluxo que se inicia, passa pelo componente *"JavaScript"* para adicionar código de forma manual e pela *"Server Action"* que é criada na parte lógica da aplicação para envio do *e-mail* e termina.

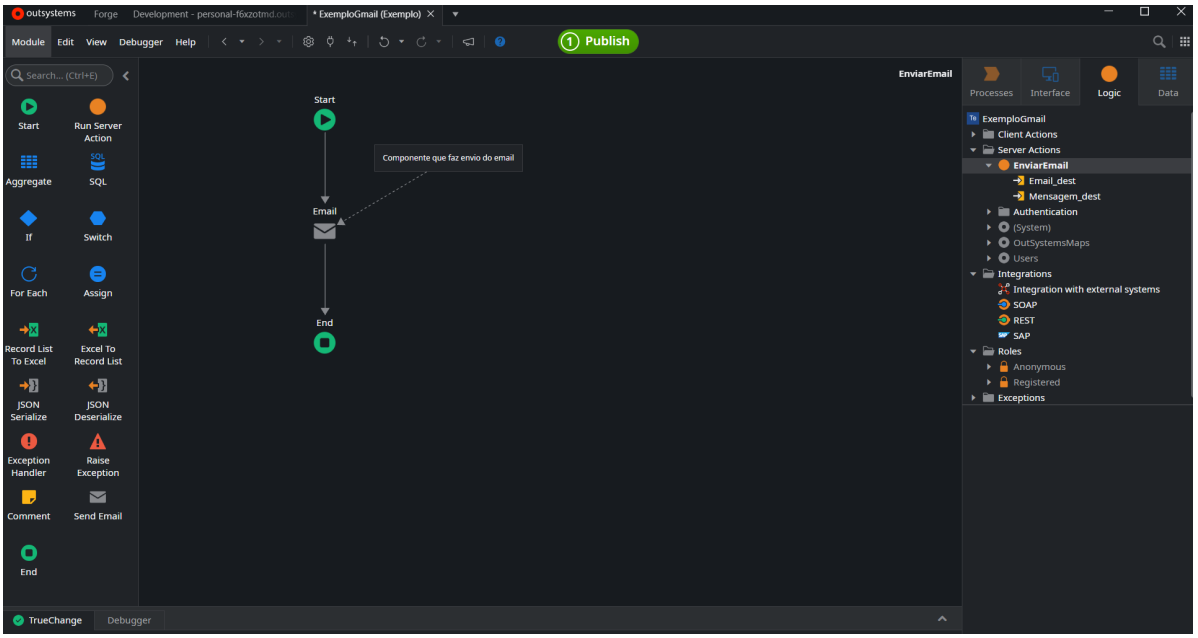
Figura 8 - Criando fluxo lógico do evento *"OnClick"*.



Fonte: Autoria Própria.

2. Criando uma *"Server Action"* vá na aba *"Logic"* na parte superior direita, pasta *"Server Actions"* e clique nela com o botão direito e crie uma nova *"Server Action"*. Após isso arraste um componente *"Send Email"* da coluna lateral esquerda, defina o destinatário e conteúdo do email e finalize o fluxo.

Figura 9 - Criando *Server Action*.

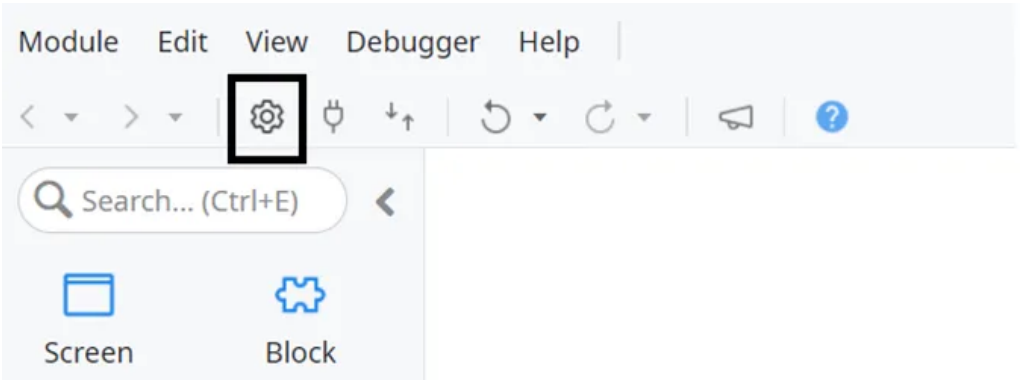


Fonte: Autoria Própria.

6. Configuração de SMTP de *e-mail* no *"Service Center"*

Abra o *"Service Center"* clicando na opção *"Module Management in Service Center"* conforme a imagem a seguir:

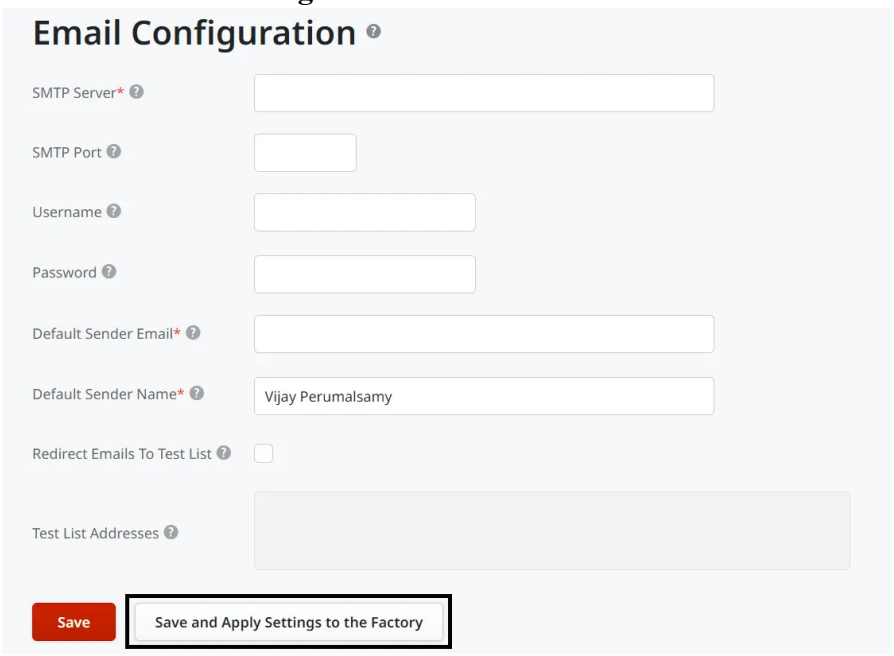
Figura 10 - Abrindo *Service Center*.



Fonte: PERUMALSAMY (2023).

No *"Service Center"*, vá para *"Administration"* > *"Email tab"* e defina as configurações de SMTP na guia "*Email"* e clique em *"Save and Apply Settings to the Factory"* PERUMALSAMY (2023):

Figura 11 - Abrindo *Service Center*.



Fonte: PERUMALSAMY (2023).

7. Pronto, agora basta clicar em *"Publish"* no *"Service Studio"* para publicar e executar a aplicação.

2.4 High-Code Vs No-Code/Low-Code

Tabela 1 - Comparação entre o High-Code e No-Code e Low-Code.

CARACTERÍSTICA	DESENVOLVIMENTO HIGH-CODE	DESENVOLVIMENTO NO-CODE/LOW-CODE
Recursos de desenvolvimento	Desenvolvedores com expertise em linguagens de programação para: ● Desenvolvimento de aplicativos web ● Desenvolvimento de aplicativos móveis	Uma interface intuitiva de arrastar e soltar para desenvolvimento de aplicativos web/móveis por: ● Desenvolvedores cidadãos ● Desenvolvedores profissionais
Tempo de desenvolvimento	Dura meses e inclui: ● Extensa codificação ● Várias revisões de protótipo ● Testes e correções de bugs	É acelerado em 10x graças a: ● Sem necessidade de codificação ● Modelos de design prontos ● Interface de arrastar e soltar
Custo de desenvolvimento	Depende da taxa horária do desenvolvedor: ● EUA e Canadá - $ 100- $ 175 ● Europa Central - $ 40- $ 85	Depende do plano de assinatura e do número de usuários: ● Plataforma OutSystems - a partir de $ 4.000 por mês

	• Ucrânia - $ 25- $ 40	• Plataforma VisionX - $ 799 por desenvolvedor por mês
Multiplataforma	• O aplicativo deve ser desenvolvido separadamente para cada plataforma	O aplicativo pode funcionar: • Mobile • Web • Nuvem
Manutenção	• Requer desenvolvimento adicional • Precisa de suporte extensivo	• Fácil de atualizar e estender • Ótimo para prototipagem
Implantação	• Requer muitos passos • Requer muitos recursos de desenvolvimento • Lento e complexo	• Implantação para vários ambientes com apenas um click, o que reduz consideravelmente o tempo
Segurança	• A segurança é responsabilidade dos desenvolvedores. • Os desenvolvedores têm controle total sobre os dados do banco de dados.	• As melhores práticas de segurança já implementadas nativamente. • Em certas plataformas, os desenvolvedores têm menor controle sobre os dados armazenados.

Fonte: PETLOVANA, [ca. 2023].

2.5 No-Code Vs Low-Code

As plataformas de baixo código apresentam uma vantagem significativa sobre as abordagens tradicionais de desenvolvimento de software, e se mostram melhor alinhadas com os requisitos dinâmicos do mundo dos negócios atual. Ao considerar a abordagem de baixo código para o desenvolvimento de software, é comum surgir o debate entre *Low-Code* e *No-Code.* Mas, afinal, qual é a diferença entre essas duas abordagens e qual é mais adequada para o contexto atual e futuro do desenvolvimento de aplicativos?. (NERO, 2021)

Tabela 2 - Low-Code vs No-Code.

CARACTERÍSTICAS	**LOW-CODE**	**NO-CODE**
Público Alvo	Desenvolvedores	Usuários de negócios
Objetivo Primário	Velocidade de desenvolvimento	Facilidade de uso
Necessidade de Código	Pouco, mas presente	Nenhuma codificação necessária
Customização	Personalização total disponível	Modelos pré-construídos podem ser personalizados
Desenvolvimento de ponta a ponta	Todas as plataformas fornecem desenvolvimento de ponta a ponta	Algumas plataformas fornecem apenas recursos limitados
Complexidade do aplicativo	Pode criar aplicativos complexos	Pode criar aplicativos simples
Propósito	Ferramentas de desenvolvimento rápido de aplicativos de última geração para desenvolvedores profissionais	Aplicativo de auto atendimento para usuários corporativos

Fonte: NERO (2021).

3 ABORDAGEM LOW-CODE

Nos últimos anos, testemunhamos o surgimento de uma nova geração de ambientes de desenvolvimento de software, conhecida como *"low-code"* ou código baixo. Esses ambientes não apenas oferecem uma perspectiva promissora de aumentar significativamente a produtividade no desenvolvimento de software, mas também abrem novas possibilidades para promover a aliança entre negócios e tecnologia da informação, bem como capacitar os usuários. (TALEB, [ca. 2023])

O termo *"low-code"* abrange várias plataformas que são conhecidas de diferentes formas, como plataforma de código baixo (LCP), plataforma de aplicativo de código baixo (LCAP) e plataforma de desenvolvimento de código baixo (LCDP). Essas plataformas oferecem ferramentas e recursos que simplificam o processo de criação de aplicativos, permitindo que desenvolvedores com diferentes níveis de experiência em programação aproveitem os benefícios. (TALEB, [ca. 2023])

Com o *low-code*, mesmo desenvolvedores com algum nível de experiência em programação podem desfrutar de um significativo aumento de produtividade no desenvolvimento de aplicações em comparação com abordagens mais tradicionais. Essa abordagem reduz a necessidade de escrever código extensivo manualmente, oferecendo uma interface visual intuitiva e ferramentas de arrastar e soltar que simplificam o processo de criação de aplicativos. Isso permite que os desenvolvedores se concentrem mais na lógica do negócio e nos requisitos do usuário, em vez de lidar com detalhes técnicos complexos. (TALEB, [ca. 2023])

Além disso, o low-code promove a colaboração entre equipes de negócios e tecnologia da informação. Com a facilidade de uso das plataformas de código baixo, as pessoas de diferentes áreas podem participar do processo de desenvolvimento de software, fornecendo *insights* valiosos e contribuindo para a criação de soluções mais eficientes. Essa abordagem não apenas agiliza o desenvolvimento, mas também fortalece a parceria entre as áreas, permitindo que os negócios acompanhem as demandas do mercado de forma mais rápida e eficaz. (TALEB, [ca. 2023])

Em suma, o low-code representa uma revolução no desenvolvimento de software, capacitando os desenvolvedores a criar aplicações com maior rapidez e eficiência. Essa abordagem inovadora não apenas aumenta a produtividade, mas também promove a colaboração entre equipes e impulsiona a transformação digital dos negócios. Com o

surgimento contínuo de novas plataformas de *low-code*, podemos esperar um futuro promissor para o desenvolvimento de software, onde a tecnologia se torna mais acessível e os limites da inovação são constantemente desafiados. (TALEB, [ca. 2023])

3.1 Características das Plataformas Low-Code

A característica distintiva do *low-code* é que ele é um método de desenvolvimento de *software* que se destaca por sua abordagem visual em vez de processual. Os desenvolvedores têm a capacidade de criar aplicativos com facilidade, selecionando itens de uma ampla variedade de módulos e modelos pré-codificados, e arrastando-os e soltando-os em uma interface gráfica do usuário (GUI) de forma lógica. (TALEB, [ca. 2023])

Uma das principais vantagens do low-code é que até 90% da lógica de um aplicativo já está pronta nos módulos pré-escritos, deixando apenas cerca de 10% para ser codificado manualmente, quando necessário para atender a requisitos específicos ou interfaces especiais. Esse processo de design é até 10 vezes mais rápido do que os métodos tradicionais, o que resulta em custos mais baixos e maior produtividade. (TALEB, [ca. 2023])

Com o low-code, os desenvolvedores podem criar aplicativos de forma eficiente, utilizando componentes pré-desenvolvidos, sem a necessidade de escrever todo o código do zero. Isso reduz consideravelmente o tempo de desenvolvimento, permitindo que se concentrem em personalizações e funcionalidades únicas. Além disso, a abordagem visual facilita a compreensão e colaboração entre os membros da equipe de desenvolvimento, acelerando ainda mais o processo. (TALEB, [ca. 2023])

Dessa forma, o low-code se torna uma solução poderosa para empresas e desenvolvedores que buscam agilidade, eficiência e redução de custos em seus projetos de desenvolvimento de software. Ao aproveitar os benefícios do low-code, é possível obter resultados mais rápidos e satisfatórios, impulsionando a inovação e a transformação digital de forma eficaz. (TALEB, [ca. 2023])

3.2 Benefícios do Low-Code para Desenvolvedores de Software

O uso do *low-code* traz uma série de benefícios para os desenvolvedores de *software* profissionais. Essa abordagem de modelagem visual permite que usuários com experiência em

negócios, mas sem habilidades de codificação, produzam aplicativos de forma rápida e fácil, automatizando seus fluxos de trabalho. No entanto, o *low-code* também é extremamente vantajoso para os próprios desenvolvedores experientes. (TALEB, [ca. 2023])

Uma das principais vantagens é a velocidade. O *low-code* é significativamente mais rápido do que a codificação tradicional. Ele permite que os desenvolvedores criem aplicativos de negócios complexos e sofisticados, adicionando apenas pequenas quantidades de código manual aos seus projetos de baixo código. Isso agiliza o processo de desenvolvimento, permitindo que os desenvolvedores entreguem soluções de software mais rapidamente. (TALEB, [ca. 2023])

Além disso, o *low-code* oferece flexibilidade durante todo o ciclo de desenvolvimento. Se os requisitos do projeto mudarem, os designers podem fazer ajustes rapidamente em qualquer ponto do processo. Isso garante que os aplicativos possam se adaptar às futuras mudanças nos ambientes tecnológicos ou de negócios, sem que seja necessário descartar meses ou anos de código anteriormente produzido. (TALEB, [ca. 2023])

Outro benefício importante do *low-code* é a colaboração aprimorada entre desenvolvedores de *software* e usuários finais não técnicos. A natureza visual do desenvolvimento de baixo código permite que ambos tenham um entendimento comum do design. Isso possibilita a produção de protótipos visuais que estão alinhados de forma precisa com os requisitos operacionais reais. Essa colaboração eficaz acelera o processo de desenvolvimento e reduz erros dispendiosos, tornando-se uma vantagem da abordagem de *low-code*. (TALEB, [ca. 2023])

Além dos benefícios mencionados, o low-code também proporciona uma maior disponibilidade de pessoal, pois permite que programadores menos qualificados ou mesmo não programadores participem do processo de desenvolvimento. Isso amplia a quantidade de recursos disponíveis e agiliza futuras atualizações e iterações do projeto. O low-code também permite a criação de ferramentas personalizadas para departamentos individuais, promovendo a eficiência e a inovação de baixo custo. (J.BIGELO, 2023)

Outra vantagem é o alojamento de projetos de nicho de forma rápida e econômica. O low-code pode acomodar aplicações com pequenas bases de usuários, tornando-os uma opção viável mesmo para projetos com orçamentos limitados. Além disso, as plataformas de low-code facilitam a gestão do desempenho, governança e conformidade, permitindo o controle do desenvolvimento e fornecendo recursos de instrumentação, análise e relatórios para melhorar a eficiência e solucionar problemas. (J.BIGELO, 2023)

Em suma, o uso do low-code traz benefícios significativos para os desenvolvedores e usuários finais. A velocidade de desenvolvimento acelerada, a flexibilidade, a colaboração aprimorada, a maior disponibilidade de pessoal, a eficiência melhorada, a inovação de baixo custo, o alojamento de projetos de nicho e a gestão aprimorada são todos pontos fortes dessa abordagem, proporcionando uma maneira eficaz de criar aplicativos. (J.BIGELO, 2023)

3.3 Princípios de Desenvolvimento de Baixo Código

A abordagem de low-code é altamente automatizada, porém, as plataformas e os processos de desenvolvimento não são completamente automáticos. Além disso, o sucesso dos projetos low-code é fortemente influenciado por princípios sólidos de negócios e tecnologia, por isso é importante se atentar nos seguintes pontos:

Entenda a plataforma low-code.

As plataformas de baixo código não executam todo o trabalho automaticamente. É essencial o envolvimento de todas as partes interessadas, desde desenvolvedores, analistas de negócios e proprietários de projetos, para participarem ativamente durante o desenvolvimento em plataformas de baixo código. O tempo investido no aprendizado da plataforma abrirá portas que poderão ser usadas em projetos futuros, potencialmente agregando ainda mais valor à iniciativa. (J.BIGELO, 2023)

Evite customizações.

O poder do low-code reside na utilização de componentes pré-definidos que podem ser facilmente arrastados e soltos no fluxo de trabalho. Esses componentes são projetados para serem genéricos e se adaptarem a uma ampla gama de casos de uso, embora nem sempre atendam às necessidades específicas de um determinado projeto. Entretanto é possível modificar os componentes já existentes e criar novos, como elementos de interface do usuário e design visual, porém é importante ressaltar que essa personalização pode aumentar o tempo de desenvolvimento, esforço e custos envolvidos, além de potencialmente comprometer a velocidade e simplicidade que as tecnologias de baixo código oferecem. (J.BIGELO, 2023)

Generalizar personalizações.

Ao personalizar um aplicativo, é importante levar em consideração os recursos fornecidos pelo contexto dos objetivos de uso comuns ou de alto nível. Dessa forma, ao projetar um componente personalizado, é possível pensar na sua possível reutilização em outros projetos, maximizando assim a eficiência e a economia de recursos. (J.BIGELO, 2023)

Não negligencie a equipe.

Para utilizar efetivamente uma plataforma de baixo código, a equipe responsável pela seleção e adoção dessa tecnologia deve ter um entendimento completo dos requisitos e objetivos de negócio do projeto em questão. Embora esses requisitos possam ser mais simples em comparação com os projetos de software tradicionais, é essencial que os proprietários do produto e outras partes interessadas importantes estejam disponíveis para responder a perguntas e revisar as iterações em um ambiente de desenvolvimento ágil fornecido pela plataforma de baixo código. (J.BIGELO, 2023)

A compreensão profunda dos requisitos de negócio é crucial para garantir que a plataforma de baixo código seja configurada e personalizada de acordo com as necessidades específicas do projeto. (J.BIGELO, 2023)

4 PLATAFORMA OUTSYSTEMS

A plataforma *OutSystems* é uma criação da empresa de mesmo nome fundada em 2001 por Paulo Rosado, que buscava uma maneira mais eficiente de desenvolver e implementar aplicativos empresariais. Desde então, a plataforma evoluiu para se tornar uma solução de desenvolvimento de aplicativos de baixo código *(low-code)* líder no mercado, permitindo que empresas criem aplicativos personalizados de forma mais rápida e fácil do que nunca. (*OUTSYSTEMS,* [ca. 2022k])

Uma das principais vantagens da plataforma *OutSystems* é sua facilidade de uso. Com a interface intuitiva e de arrastar e soltar, os desenvolvedores podem criar aplicativos sem precisar escrever códigos complexos, tornando o processo de desenvolvimento mais eficiente e acessível a uma gama mais ampla de profissionais. Além disso, a plataforma *OutSystems* suporta várias linguagens de programação e integrações com outros sistemas, permitindo que os desenvolvedores criem aplicativos altamente personalizados e adaptáveis. (*OUTSYSTEMS,* [ca. 2022e])

A estrutura da plataforma *OutSystems* é baseada em quatro componentes principais: o *Service Studio*, o *Integration Studio*, o *LifeTime* e o *OutSystems Platform Server*. O *Service Studio* é a interface de desenvolvimento onde os desenvolvedores criam aplicativos usando um editor visual e uma linguagem de modelagem de fluxo de trabalho. O *Integration Studio* é usado para integrar aplicativos criados na plataforma *OutSystems* com outros sistemas empresariais. O *LifeTime* é uma plataforma de gerenciamento que permite aos desenvolvedores implantar e gerenciar aplicativos, bem como controlar o acesso dos usuários. Finalmente, o *OutSystems Platform Server* é o servidor que hospeda os aplicativos criados na plataforma. (*OUTSYSTEMS,* [ca. 2022e])

Além de sua facilidade de uso e flexibilidade, a plataforma *OutSystems* também é conhecida por sua capacidade de criar aplicativos de alta qualidade e seguros. A plataforma possui recursos de segurança avançados, incluindo autenticação de usuário, gerenciamento de permissões e criptografia de dados. Além disso, a plataforma *OutSystems* segue as melhores práticas de desenvolvimento e testes de software, garantindo que os aplicativos criados sejam confiáveis e de alta qualidade. (*OUTSYSTEMS,* [ca. 2022e])

Em resumo, a plataforma *OutSystems* é uma solução de desenvolvimento de aplicativos de baixo código líder no mercado, criada para ajudar empresas a criar aplicativos personalizados de forma mais rápida, fácil e eficiente. Com sua interface intuitiva, recursos de

segurança avançados e suporte para várias linguagens de programação, a plataforma *OutSystems* é uma escolha popular para empresas que buscam uma maneira mais eficiente de criar aplicativos de qualidade. (*OUTSYSTEMS,* [ca. 2022e])

4.1 Quem Pode Desenvolver Com *Outsystems*?

O *OutSystems* é uma ferramenta de desenvolvimento que atende a diversos perfis profissionais, desde analistas de negócios com pouca ou nenhuma experiência em programação até desenvolvedores experientes em busca de novas ferramentas tecnológicas. A facilidade de aprendizado é uma das vantagens do *OutSystems* em relação a outras tecnologias, como *.NET, Java* e *Swift*, o que torna seu uso acessível a qualquer pessoa com conhecimentos básicos de desenvolvimento. (*OUTSYSTEMS,* [ca. 2022b])

4.2 Como Isso é Possível?

A facilidade de leitura e organização das aplicações desenvolvidas no *OutSystems* é resultado de sua natureza visual. Essa característica se estende tanto para o desenvolvimento *front-end* quanto para o *back-end*, abrangendo desde as interfaces de usuário até a criação e consulta de bancos de dados, passando pela lógica de negócios e integração. (*OUTSYSTEMS,* [ca. 2022b])

4.3 Quanto Tempo Demora Para Aprender *Outsystems*?

Geralmente, desenvolvedores juniores e experientes têm uma curva de aprendizado mais rápida ao utilizar a plataforma *OutSystems*, pois esses profissionais já possuem conhecimentos prévios em desenvolvimento de *software* e familiaridade com as tecnologias envolvidas. Além disso, tendem a ter habilidades para estruturar e arquitetar soluções de forma eficiente, identificar componentes reutilizáveis e manipular dados, entre outras competências. O período de aprendizado para se tornar proficiente em *OutSystems* varia de 1 a 3 semanas, dependendo do histórico do desenvolvedor e das opções de treinamento disponíveis. (*OUTSYSTEMS,* [ca. 2022c])

Mesmo pessoas com outras experiências, como analistas de negócios, podem facilmente aprender a utilizar a plataforma *OutSystems*. Os desenvolvedores podem se tornar produtivos em apenas uma semana, graças ao uso de construções de desenvolvimento de

software que já são familiares a desenvolvedores que possuem conhecimento em outras tecnologias. Para dominar a plataforma, são necessárias apenas noções básicas de programação e uma compreensão dos conceitos de banco de dados relacionais, que são habilidades fundamentais no desenvolvimento de software. (*OUTSYSTEMS,* [ca. 2022c])

Tabela 3 -Tempo médio de aprendizado.

	Menos de uma semana	**De uma a menos de duas semanas**	**Duas ou mais semanas**
Arquitetos de soluções		x	
Desenvolvedores	x		
Analistas de negócios			x
DBAs	x		

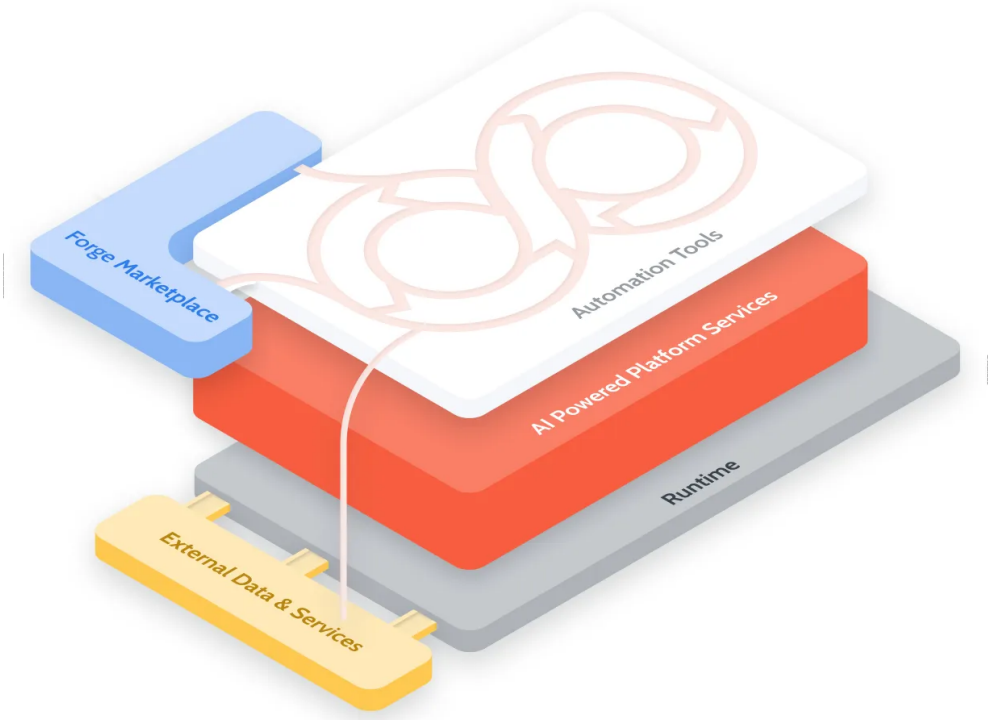
Fonte: *OUTSYSTEMS,* [ca. 2022c]

A plataforma *OutSystems* simplifica a complexidade das diversas tecnologias envolvidas no desenvolvimento de aplicações, incluindo *HTML5, CSS3, JavaScript, SQL, C#* e tecnologias móveis nativas integradas via *Cordova*. Essa abstração permite que profissionais que não possuem experiência nessas tecnologias se tornem produtivos em um curto período de tempo, geralmente menos de um mês. Ao mesmo tempo, desenvolvedores experientes podem usufruir de ganhos significativos em produtividade, já que podem criar extensões de plataforma e componentes visualmente reutilizáveis que encapsulam suas habilidades nas tecnologias subjacentes. (*OUTSYSTEMS,* [ca. 2022c])

4.4 Arquitetura

A arquitetura da plataforma *OutSystems* é um ecossistema em camadas que oferece aos desenvolvedores a capacidade de criar aplicativos de forma rápida, precisa e preparados para o futuro. *(OUTSYSTEMS,* [ca. 2022a])

Figura 12 - Arquitetura OutSystems.



Fonte: *OUTSYSTEMS,* [ca. 2022a]

A plataforma *OutSystems* conta com uma série de ferramentas, construtores, processos e componentes, todos incorporados com tecnologia de Inteligência Artificial (IA). Esses elementos têm como objetivo solucionar os desafios relacionados à integração de dados, desenvolvimento e design de aplicativos, bem como gerenciamento do ciclo de vida do desenvolvimento de *software (SDLC)*. Para dar suporte a essa camada superior, há uma camada de serviços que utiliza tecnologia de *IA* para automatizar processos complexos de gerenciamento de mudanças e revisão de arquitetura, eliminando tarefas repetitivas e suposições no desenvolvimento de aplicativos. Essa camada também oferece mais de 250 pontos de segurança, além de geração de código e registro. É importante destacar que a camada de tempo de execução é independente do processo de desenvolvimento, permitindo que os aplicativos sejam implantados em diferentes ambientes, incluindo a nuvem e localmente. (*OUTSYSTEMS,* [ca. 2022a])

4.4.1 Como a Outsystems Cria Aplicativos Rapidamente

A arquitetura da *OutSystems* proporciona aos desenvolvedores a oportunidade de criar aplicativos completos e atrativos de forma ágil e eficiente, por intermédio de um ambiente de desenvolvimento visual e ferramentas que automatizam e otimizam as fases cruciais do ciclo de vida do desenvolvimento de software (SDLC). (*OUTSYSTEMS,* [ca. 2022a])

- Ferramentas especializadas aceleraram o processo de desenvolvimento de aplicações, tornando mais simples a criação de tipos específicos de aplicativos e a conexão com dados corporativos, reduzindo com isso a complexidade do processo.
- A integração com mais de 275 sistemas corporativos, banco de dados e aplicativos de terceiros possibilitou a criação de um ambiente flexível, escalável e aberto, concebido para permitir um desenvolvimento ágil e flexível a mudanças.
- Modelos e padrões de interface do usuário previamente desenvolvidos têm o benefício de evitar a duplicação de esforços, o que poupa tempo e permite que experiências de alta performance e visualmente atrativas sejam criadas de forma mais rápida em diferentes dispositivos.
- Um repositório de código reutilizável criado por uma comunidade de mais de 300.000 desenvolvedores e parceiros, que disponibilizam componentes e conectores. Isso acarreta em menos necessidade de programação e retrabalho para os usuários.
- A utilização de inteligência artificial durante o processo de desenvolvimento, ajuda os desenvolvedores na criação de aplicativos, indicando as melhores ações a serem tomadas e as fontes de auxílio mais adequadas, reduzindo problemas e acelerando a entrega do projeto.

4.4.2 Como a Outsystems Cria Aplicativos Corretamente

Segundo a *OutSystems* [ca. 2022a], sua arquitetura é constituída por uma série de recursos integrados e infraestrutura que garantem a segurança, resiliência e escalabilidade dos aplicativos empresariais modernos:

- Os serviços de aplicativos fornecidos pela *OutSystems* garantem a escalabilidade, governança e conformidade necessárias para que haja um bom funcionamento das aplicações.
- Durante o processo de desenvolvimento e execução do aplicativo móvel, mais de 200 verificações de segurança são feitas para garantir a proteção dos usuários. Além disso, para garantir ainda mais a segurança, há um recurso especial que adiciona automaticamente camadas de segurança durante a implantação do aplicativo.
- Os aplicativos são constantemente monitorados a fim de assegurar que as boas práticas de desenvolvimento sejam seguidas, isso é feito utilizando mecanismos de inteligência artificial que detecta e soluciona possíveis problemas previamente, o que evita falhas de concepção e redundância de esforços.
- Com dados de desempenho de aplicativos em tempo real, é possível detectar falhas e pontos de melhoria, contribuindo para a redução de custos e aumento da satisfação dos usuários.

4.4.3 Como a *Outsystems* Cria Aplicativos Para o Futuro

A *OutSystems* concebeu uma arquitetura que visa facilitar o gerenciamento de mudanças pelos desenvolvedores. Através de serviços de plataforma, inteligência artificial e ferramentas visuais, torna-se possível introduzir de forma contínua novos recursos e capacidades, permitindo que os desenvolvedores criem aplicativos que se adaptam rapidamente às alterações no mercado e nas tecnologias disponíveis. *(OUTSYSTEMS,* [ca. 2022a]*)*

- Uma verificação constante de mudanças é realizada por um serviço que tem por objetivo identificar bugs e falhas na arquitetura, avaliar os impactos das alterações nas dependências e oferecer controle de equipe e de arquitetura.
- A plataforma permite que usuários de negócios modifiquem fluxos de trabalho e processos fazendo uso de automação inteligente, juntamente com aceleradores pré-construídos.

- A partir do catálogo de componentes da *OutSystems*, é possível obter tecnologias inovadoras e em constante evolução, como *IA, IoT e RPA*, e integrá-las de forma rápida e simples aos seus aplicativos.
- Com um pipeline de *CI/CD* completamente automatizado, é possível implantar, modificar e reverter um aplicativo com apenas um clique, enquanto a aprovação baseada em função é utilizada para governar todos os estágios e camadas do *software*.

4.4.4 Ferramentas de Desenvolvimento e Gestão Outsystems

Na arquitetura *OutSystems*, encontra-se o conjunto de ferramentas fundamentais para o desenvolvimento e gestão de diversas aplicações, que abrangem desde as destinadas ao uso móvel e pessoal até aquelas consideradas críticas para o funcionamento das empresas. (*OUTSYSTEMS,* [ca. 2022e])

4.4.4.1 Camada de Desenvolvimento

A plataforma *OutSystems* disponibiliza aos seus usuários um ambiente de desenvolvimento visual, além de outras ferramentas que contam com a inteligência artificial para acelerar o processo de criação de aplicações destinadas a atender as necessidades futuras. (*OUTSYSTEMS,* [ca. 2022e])

As ferramentas nesta camada incluem:

- *Service Studio*
- *Experience Builder*
- *Workflow Builder*
- *Integration Studio*
- *Forge*

4.4.4.1.1 Service Studio

O *Service Studio* é um ambiente completo que permite a criação de todas as partes de uma aplicação, desde o modelo de dados até a interface do usuário, passando pela lógica de negócios, fluxos de processos e integrações, além das políticas de segurança. Nesse ambiente, os designers têm a possibilidade de arrastar e soltar elementos visuais para criar as UIs, processos de negócios, lógica e modelos de dados para seus aplicativos. Os aplicativos criados nesse ambiente podem consumir e expor serviços da *Web SOAP* e *REST*. O desenvolvimento é auxiliado por inteligência artificial, que orienta os desenvolvedores durante o processo de desenvolvimento, guiando-os nas próximas etapas, preenchendo campos automaticamente e oferecendo informações referentes à *IDE*. (*OUTSYSTEMS,* [ca. 2022e])

O aplicativo tem pontos de gancho em todas as suas camadas, o que permite que os desenvolvedores estendam as camadas com seu próprio código, caso precisem trabalhar fora do ambiente de desenvolvimento. Além disso, o ambiente é equipado com um mecanismo completo de verificação de referência e autocorreção, impulsionado por *IA*, que funciona em segundo plano para garantir que os aplicativos não contenham erros e que as alterações não afetem os aplicativos existentes. O *Platform Server* é o componente de servidor da plataforma *OutSystems* que é responsável pela maior parte do trabalho de pós-design. (*OUTSYSTEMS,* [ca. 2022e])

4.4.4.1.2 Experience Builder

O *Experience Builder* é uma ferramenta especializada que faz parte de uma coleção de ferramentas projetadas para agilizar o desenvolvimento de aplicativos e simplificar a complexidade de conectá-los a dados corporativos. Com essa ferramenta, desenvolvedores e designers podem criar protótipos de aplicativos móveis atraentes e altamente utilizáveis em poucas horas, e gerar código pronto para produção em minutos. (*OUTSYSTEMS,* [ca. 2022e])

O *Experience Builder* utiliza modelos de fluxo para aplicativos móveis comuns, permitindo que os desenvolvedores criem experiências que os usuários já conhecem, sem precisar começar do zero. Além disso, com dados de amostra contextual, é possível compartilhar *feedback* entre usuários e partes interessadas. (*OUTSYSTEMS,* [ca. 2022e])

O *Experience Builder* oferece muitas vantagens para desenvolvedores e designers, incluindo acesso a aplicativos líderes do setor, melhores práticas, animações nativas integradas e um *plug-in* nativo para recursos do dispositivo. Além disso, ao concluir a criação do código, é possível fazer a entrega com total confiança. (*OUTSYSTEMS,* [ca. 2022e])

4.4.4.1.3 Workflow Builder

O *Workflow Builder* é uma ferramenta poderosa que simplifica os desafios e aumenta a produtividade para especialistas em negócios, analistas e proprietários de processos internos. Com esta ferramenta, os usuários podem criar aplicativos de negócios sem a necessidade de conhecimentos avançados em desenvolvimento, pois o assistente abstrai o processo de design do fluxo de trabalho. Além dos aplicativos criados terem as mesmas arquiteturas e estruturas padrão das contrapartes desenvolvidas profissionalmente, pois são gerados em um conjunto aberto, otimizado e documentado de componentes de servidor e cliente. (*OUTSYSTEMS,* [ca. 2022e])

Com a ajuda do *Workflow Builder*, proprietários de processos de negócios podem criar aplicativos de gerenciamento de casos com complexidade que varia de simples a média em questão de minutos, abrangendo desde a aprovação de despesas até o gerenciamento de quadro de horários. À medida que esses aplicativos de linha de negócios crescem em importância e escala, equipes multifuncionais compostas por membros de diferentes níveis de habilidade podem continuar a desenvolvê-los no *Service Studio*. Lá, especialistas em negócios e desenvolvedores profissionais têm a oportunidade de colaborar e refinar processos de fluxo de trabalho, interfaces de usuário, lógica de negócios e modelos de dados, para criar soluções ainda mais eficientes e personalizadas para seus usuários. (*OUTSYSTEMS,* [ca. 2022e])

4.4.4.1.4 Integration Studio

O *Integration Studio* é um ambiente de desenvolvimento que possibilita a criação de componentes e integrações de maneira simplificada e intuitiva na plataforma *OutSystems*. Esses componentes possibilitam a conexão com diversos sistemas e microsserviços de terceiros, expandindo, assim, as funcionalidades da plataforma. (*OUTSYSTEMS,* [ca. 2022e])

Esses componentes são criados a partir do *Visual Studio* e das diversas bibliotecas *.NET* disponíveis na plataforma, e uma vez criados, esses componentes podem ser reutilizados

em todas as aplicações desenvolvidas na OutSystems, proporcionando assim maior eficiência e produtividade para os desenvolvedores. (*OUTSYSTEMS,* [ca. 2022e])

Quando um componente é publicado no ambiente de desenvolvimento, ele é compilado utilizando um compilador *.NET*, e o resultado dessa compilação gera *DLLs* que posteriormente são enviadas para o *Platform Server*. (*OUTSYSTEMS,* [ca. 2022e])

4.4.4.1.5 Forge

Este é um repositório que armazena uma variedade de componentes de interface de usuário, bibliotecas e conectores com o propósito de acelerar a entrega de resultados e aumentar a produtividade da equipe. Esses módulos pré-existentes podem ser reutilizados em diversas aplicações criadas na plataforma, permitindo assim a rápida criação de soluções. (*OUTSYSTEMS,* [ca. 2022e])

O ambiente de desenvolvimento é integrado à *Forge*, o que possibilita que desenvolvedores e outros interessados utilizem esses componentes em vários projetos de forma rápida e prática. (*OUTSYSTEMS,* [ca. 2022e])

A maioria dos recursos presentes na *Forge* da plataforma *OutSystems*, é desenvolvida e compartilhada pela sua própria comunidade, e para garantir a confiabilidade e eficiência desses componentes, eles passam por um rigoroso processo de curadoria realizado por especialistas. Especialistas esses que avaliam se os componentes entregam a funcionalidade prometida e se seguem as melhores políticas de segurança, documentação e qualidade de código. Além disso, existem os componentes com suporte, que são os únicos a receber manutenção e suporte direto da equipe da *OutSystems*, de acordo com os termos da assinatura de cada cliente. (*OUTSYSTEMS,* [ca. 2022e])

4.4.4.2 Gerenciamento

Para garantir entregas bem-sucedidas tanto no presente quanto no futuro, é crucial gerenciar de forma eficaz a implantação de aplicativos em todo o ciclo de vida do desenvolvimento de *software (SDLC).* A plataforma *OutSystems* oferece diversas soluções para facilitar esse gerenciamento, como consoles, ferramentas e um repositório centralizado que possibilita a gestão dos ambientes de forma mais eficiente, reduzindo as chamadas "dívidas técnicas" e aumentando a facilidade de reutilização dos módulos de aplicativos. (*OUTSYSTEMS,* [ca. 2022e])

Os principais componentes desta camada são:

- *Lifetime*
- *Service Center*
- *Architecture Dashboard*

4.4.4.2.1 *Lifetime*

O *Lifetime* é um console versátil que permite o gerenciamento centralizado de diversos ambientes de desenvolvimento, controle de qualidade e produção, tanto em nuvem como em servidores locais. Com recursos avançados de automação de *DevOps*, o *Lifetime* capacita indivíduos e equipes a preparar seus aplicativos desde a fase de desenvolvimento até a implantação na produção. Além disso, o Console *Lifetime* oferece uma ampla gama de recursos, como ambiente de implantação completo, para garantir uma transição suave e eficiente dos aplicativos para a produção. (*OUTSYSTEMS,* [ca. 2022e])

Esses recursos são:

- Desenvolvimento de aplicações em ambientes de teste. Durante o processo de desenvolvimento de um aplicativo, o *LifeTime* realiza uma análise de dependência entre os diferentes ambientes para assegurar que os testes não interfiram na execução dos aplicativos no ambiente final. Após a implementação, todos os aplicativos impactados pelas mudanças são automaticamente atualizados para utilizar as versões mais recentes de código e outras atualizações necessárias. (*OUTSYSTEMS,* [ca. 2022e])
- O *LifeTime* permite que todos os aplicativos da *OutSystems* tenham seu desempenho constantemente monitorado, a fim de melhorar e otimizar a experiência do usuário, otimização essa que vem antes mesmo do usuário sequer perceber algum problema ou lentidão no aplicativo. (*OUTSYSTEMS,* [ca. 2022e])
- A administração de permissões para a equipe de *TI* é uma tarefa crítica na gestão de uma organização. A plataforma *OutSystems* fornece essa funcionalidade por meio do

LifeTime, permitindo configurar o nível de acesso de cada usuário da equipe de *TI* para cada ambiente específico. (*OUTSYSTEMS,* [ca. 2022e])

4.4.4.2.2 *Service Center*

Segundo a *OutSystems,* [ca. 2022e], o *Service Center* é uma plataforma que supervisiona as diversas operações de um ambiente, além de lidar com as configurações específicas tanto do ambiente quanto do aplicativo, abrangendo:

- Colocar aplicações *offline* e *online*
- Revertendo para uma versão anterior do aplicativo
- Configurando propriedades do aplicativo, terminais de serviço da *Web* e agendamento de tarefas em lote.

4.4.4.2.3 *Architecture Dashboard*

A plataforma de monitoramento de dívida técnica da *OutSystems*, conhecida como *Architecture Dashboard*, utiliza tecnologia de inteligência artificial para examinar automaticamente tanto o código quanto o tempo de execução das aplicações, com o objetivo de fornecer recomendações que possam melhorar o desempenho, a segurança, a arquitetura e a experiência do usuário. (*OUTSYSTEMS,* [ca. 2022e])

O *Arquitetural Dashboard* oferece uma visão abrangente e unificada da dívida técnica em todas as aplicações do portfólio empresarial, além de destacar as interdependências entre os diferentes módulos. Utilizando a tecnologia de inteligência artificial da plataforma *OutSystems*, cada módulo é categorizado de acordo com sua respectiva classificação, o que facilita a identificação de áreas problemáticas. A arquitetura é descoberta automaticamente, gerando um mapa visual que mostra as áreas com problemas, que variam de leves a graves. Com estas informações, os líderes de *TI* podem priorizar ações e solucionar problemas de acordo com a necessidade de cada projeto. (*OUTSYSTEMS,* [ca. 2022e])

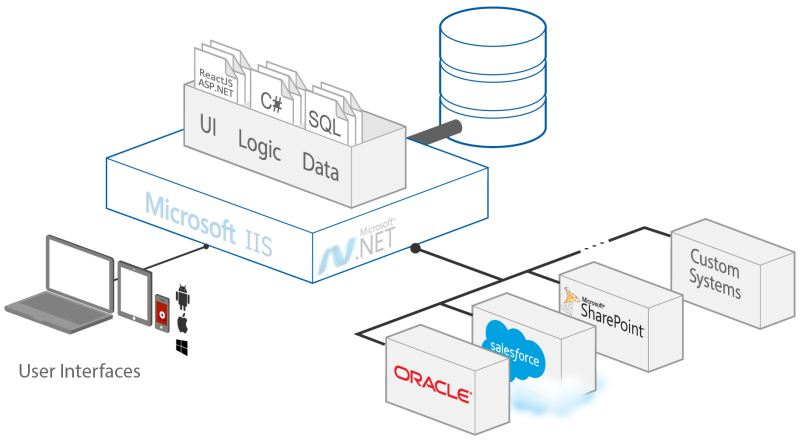
A fim de ajudar os desenvolvedores a resolver problemas de forma mais eficiente, o fornecimento de informações detalhadas sobre práticas recomendadas violadas, seu impacto e como corrigi-las é fundamental. Por meio da divisão em módulos individuais, é possível identificar as áreas problemáticas e vinculá-las aos relatórios correspondentes, possibilitando

que os desenvolvedores resolvam rapidamente problemas específicos. Além disso, é possível utilizar um recurso de refatoração guiada para identificar oportunidades de refatoração e reaproveitamento de código, oferecendo sugestões precisas para melhorar a qualidade do código. (*OUTSYSTEMS,* [ca. 2022e])

4.4.5 Arquitetura Padrão Sem Bloqueio

A *OutSystems* é uma plataforma que utiliza modelos de aplicativos para criar soluções otimizadas e padronizadas, prontas para serem executadas em uma infraestrutura *web* ou em arquiteturas em nuvem. Além disso, a plataforma faz uso de banco de dados relacional para armazenar e gerenciar os dados dos aplicativos desenvolvidos. (*OUTSYSTEMS,* [ca. 2022i])

Figura 13 - Arquitetura padrão *OutSystems*.



Fonte: *OUTSYSTEMS,* [ca. 2022i]

Após serem criados, os aplicativos desenvolvidos na plataforma *OutSystems* podem ser executados de maneira autônoma, sem depender de quaisquer componentes do sistema. Além disso, todas as informações pertinentes ao seu aplicativo permanecem sob seu controle e podem ser acessadas utilizando ferramentas padrão. (*OUTSYSTEMS,* [ca. 2022i])

Essa abordagem garante que:

- **As aplicações são otimizadas para desempenho e segurança** - Cada vez que a *OutSystems* realiza a reimplementação de seus modelos, a empresa atualiza suas aplicações com as mais recentes medidas de segurança e otimizações de desempenho. Para garantir a segurança dos sistemas desenvolvidos, são utilizados os melhores padrões de código, validados durante todo o processo de engenharia e otimização, em conformidade com as melhores práticas de segurança. (*OUTSYSTEMS,* [ca. 2022i])
- **Os aplicativos são compilados em código padrão, usando estruturas não proprietárias.** Conforme as tecnologias avançam, a *OutSystems* atualiza seu compilador para aproveitar os novos *frameworks* e recursos disponíveis, mas fazendo isso de tal forma a simplificar seu uso na plataforma. (*OUTSYSTEMS,* [ca. 2022i])
- **As aplicações são executadas em servidores e bases de dados de aplicações *web* padrão, independentemente dos componentes proprietários da *OutSystems*.** Isso oferece uma maior flexibilidade para a implantação de aplicativos em qualquer provedor de infraestrutura, seja local ou em um serviço de infraestrutura como *(IaaS)*. Além disso, o controle dos componentes de infraestrutura subjacentes é mantido pelo desenvolvedor, permitindo assim a utilização de seus próprios processos e ferramentas para tarefas como monitoramento de aplicativos e verificação de segurança. (*OUTSYSTEMS,* [ca. 2022i])
- **Você não ficará preso a OutSystems.** A *OutSystems* desenvolve suas aplicações utilizando tecnologias *standard*, como o modelo de banco de dados. Isso significa que as aplicações são protegidas, mesmo se um usuário optar por encerrar sua assinatura, os seguintes dados serão mantidos e poderão ser resgatados a qualquer momento:

    - A versão mais recente do código-fonte dos aplicativos gerados. Isso possibilita ao usuário implantá-los e executá-los em sua infraestrutura sem a necessidade de interpretadores personalizados, tempos de execução ou qualquer biblioteca proprietária específica. (*OUTSYSTEMS,* [ca. 2022i])

    - A versão mais recente do esquema de banco de dados. Isso preserva os dados já existentes para que você possa utilizá-los, migrá-los e integrá-los com outros

> aplicativos, ou simplesmente mantê-los para fins históricos. (*OUTSYSTEMS,* [ca. 2022i])

4.4.6 Integração

A plataforma *OutSystems* simplifica o processo de integração, permitindo que os desenvolvedores gerenciem as configurações sem a necessidade de escrever código personalizado. Essa abordagem reduz significativamente o tempo e o esforço necessários para integrar diferentes sistemas e serviços, eliminando erros comuns. (*OUTSYSTEMS,* [ca. 2022d])

Além disso, a integração com serviços *SOAP* e *REST* e sistemas *SAP* é incorporada na plataforma, o que significa que todos os métodos e estruturas de dados necessários são gerados automaticamente. Os desenvolvedores podem utilizá-los visualmente na lógica do aplicativo, sem precisar se preocupar com a complexidade da integração. (*OUTSYSTEMS,* [ca. 2022d])

Do ponto de vista do desenvolvedor, invocar um método *OutSystems*, uma *API REST* ou um *SAP BAPI* é praticamente indistinguível. Essa facilidade na forma de trabalhar permite que os desenvolvedores trabalhem de forma mais eficiente, sem precisar se preocupar com as nuances de cada sistema ou serviço. (*OUTSYSTEMS,* [ca. 2022d])

4.4.7 Visão Geral da Segurança *Outsystems*

A plataforma *OutSystems* disponibiliza uma ampla gama de recursos integrados para garantir a segurança em todas as etapas do ciclo de vida de um aplicativo. Todos os aplicativos desenvolvidos e entregues por meio do *OutSystems* recebem orientação de segurança provida pelo *AI Security Mentor*. Além disso, o código gerado é transparente e acessível, permitindo que cada aplicativo seja auditado e atenda aos mais rigorosos requisitos de segurança. (*OUTSYSTEMS,* [ca. 2022g])

A plataforma *OutSystems* oferece uma solução eficiente para acelerar o desenvolvimento de aplicações seguras, bem como sua implementação em um ambiente de tempo de execução seguro. Com uma ampla gama de recursos integrados, a plataforma é capaz de gerenciar o ciclo de vida completo das aplicações, promovendo uma atribuição clara de responsabilidades nos processos de *DevOps*. Isso estabelece as bases para um ciclo de vida

de desenvolvimento de software *(SDLC)* seguro e confiável, ajudando as empresas a garantir a qualidade e a segurança de seus produtos. (*OUTSYSTEMS,* [ca. 2022g])

4.4.7.1 Segurança do Aplicativo

Ao desenvolver aplicações com a plataforma *OutSystems*, é possível desfrutar de um nível extra de segurança em relação ao próprio código da aplicação. Isso ocorre porque os modelos visuais dos aplicativos são automaticamente convertidos em código *.NET*, utilizando as melhores práticas e políticas de segurança, o que resulta na redução da exposição dos aplicativos web e mobile às vulnerabilidades mais comuns. (*OUTSYSTEMS,* [ca. 2022g])

Com a plataforma *OutSystems*, os desenvolvedores têm a flexibilidade de substituir os padrões de código de segurança *standart*, por soluções personalizadas mais avançadas. Às verificações de segurança integradas, a plataforma *OutSystems* é capaz de alertar proativamente os desenvolvedores sobre possíveis vulnerabilidades de segurança ao publicarem seus aplicativos. Isso garante uma abordagem mais segura e responsável para o desenvolvimento de aplicativos, reduzindo os riscos de falhas de segurança e protegendo a privacidade dos usuários finais. (*OUTSYSTEMS,* [ca. 2022g])

A *OutSystems* possibilita a geração de aplicações *standard* durante o tempo de execução, possibilitando a utilização de ferramentas padrões de avaliação de segurança, como análise de código estático, que faz a verificação do código produzido durante a execução. (*OUTSYSTEMS,* [ca. 2022g])

A fim de assegurar altos padrões de segurança para suas aplicações, a *OutSystems* utiliza uma vasta gama de ferramentas de segurança, como parte de seu processo automatizado de garantia de qualidade. Para verificar vulnerabilidades de código, a empresa utiliza as principais ferramentas de análise de código do mercado e as configura para realizarem verificações automáticas durante uma série de testes. Esses testes são conduzidos de acordo com rigorosos critérios, exigindo que todas as vulnerabilidades do código sejam corrigidas antes do lançamento. (*OUTSYSTEMS,* [ca. 2022g])

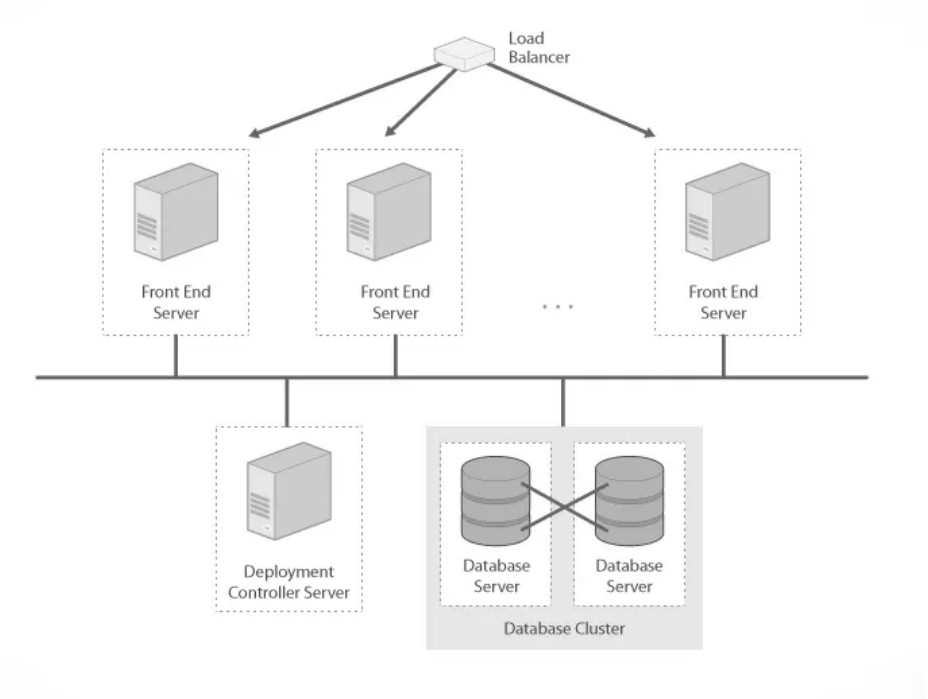
Quando são identificadas novas vulnerabilidades no código gerado, a *OutSystems* imediatamente as corrige e as inclui nas próximas atualizações da plataforma. Dessa forma, após a atualização, todas as correções são instaladas automaticamente nos aplicativos, o que reduz o custo de manutenção. (*OUTSYSTEMS,* [ca. 2022g])

4.4.8 Escalabilidade

A *OutSystems* foi projetada com foco especial na escalabilidade, possuindo uma arquitetura flexível capaz de oferecer uma ampla variedade de opções, incluindo escalabilidade vertical e horizontal, que podem ser ajustadas para atender às necessidades específicas de cada cliente. (*OUTSYSTEMS,* [ca. 2022h])

Para garantir alta escalabilidade, a arquitetura distribuída da *OutSystems* suporta o balanceamento de carga e eliminação de pontos únicos de falha no ambiente de execução, sendo que para aumentar a capacidade dos seus aplicativos, basta adicionar mais servidores *front-end* ao cluster apropriado, sendo que todos os aplicativos necessários serão instalados e configurados automaticamente. (*OUTSYSTEMS,* [ca. 2022h])

Figura 14 - Servidores *front-end.*



Fonte: *OUTSYSTEMS,* [ca. 2022h]

4.4.9 Desempenho

Durante a fase de desenvolvimento, a *OutSystems* tem como prática orientar os desenvolvedores para assegurar que cada aplicativo seja criado de maneira a evitar quaisquer limitações de desempenho, desde a etapa de validação de design. Além disso, ao gerar o código fonte, a *OutSystems* busca otimizar o desempenho em todas as camadas da aplicação, desde o número de conexões necessárias para acessar o banco de dados até o tamanho das páginas utilizadas. Com isso, a plataforma é capaz de garantir que os aplicativos desenvolvidos possuam as funcionalidades ideais para atender às necessidades dos usuários. (*OUTSYSTEMS,* [ca. 2022f])

O editor visual da plataforma *OutSystems* disponibiliza uma ferramenta de teste de validação de desempenho, que permite que os desenvolvedores identifiquem possíveis gargalos no desempenho do aplicativo antes mesmo de realizarem a implantação. Essa funcionalidade é extremamente útil para garantir que o aplicativo ofereça uma excelente experiência ao usuário final, já que os desenvolvedores podem identificar e solucionar os problemas de desempenho com antecedência. (*OUTSYSTEMS,* [ca. 2022f])

4.4.9.1 Contenção e Otimização de Código

A plataforma *OutSystems* é projetada para oferecer aplicativos de alta qualidade com desempenho excepcional. Todo o código gerado passa por otimizações que garantem o comportamento correto do aplicativo, evitando problemas de escalabilidade e confiabilidade. (*OUTSYSTEMS,* [ca. 2022f])

Além disso, a *OutSystems* usa técnicas de compactação para minimizar a largura de banda, resultando em um tamanho menor de página e reduzindo os gargalos de desempenho. O enfileiramento automático de solicitações *AJAX* garante que os usuários finais não sobrecarregue o servidor com várias solicitações simultâneas. A prevenção inteligente de solicitações *AJAX* duplicadas impede que o servidor receba solicitações idênticas, enquanto a renderização parcial da tela usando *AJAX* reduz ainda mais o tráfego e melhora a experiência do usuário. (*OUTSYSTEMS,* [ca. 2022f])

Ao reduzir a busca no banco de dados, é possível garantir que somente os dados necessários para o funcionamento do aplicativo sejam buscados, independentemente do que foi codificado pelo desenvolvedor. Além disso, os algoritmos inteligentes de gerenciamento de memória do banco de dados garantem que o uso de memória seja mínimo, carregando

apenas conjuntos de dados na memória como último recurso. Com isso, é possível obter uma maior eficiência na execução do aplicativo, reduzindo o tempo de resposta e aumentando a capacidade de processamento. (*OUTSYSTEMS,* [ca. 2022f])

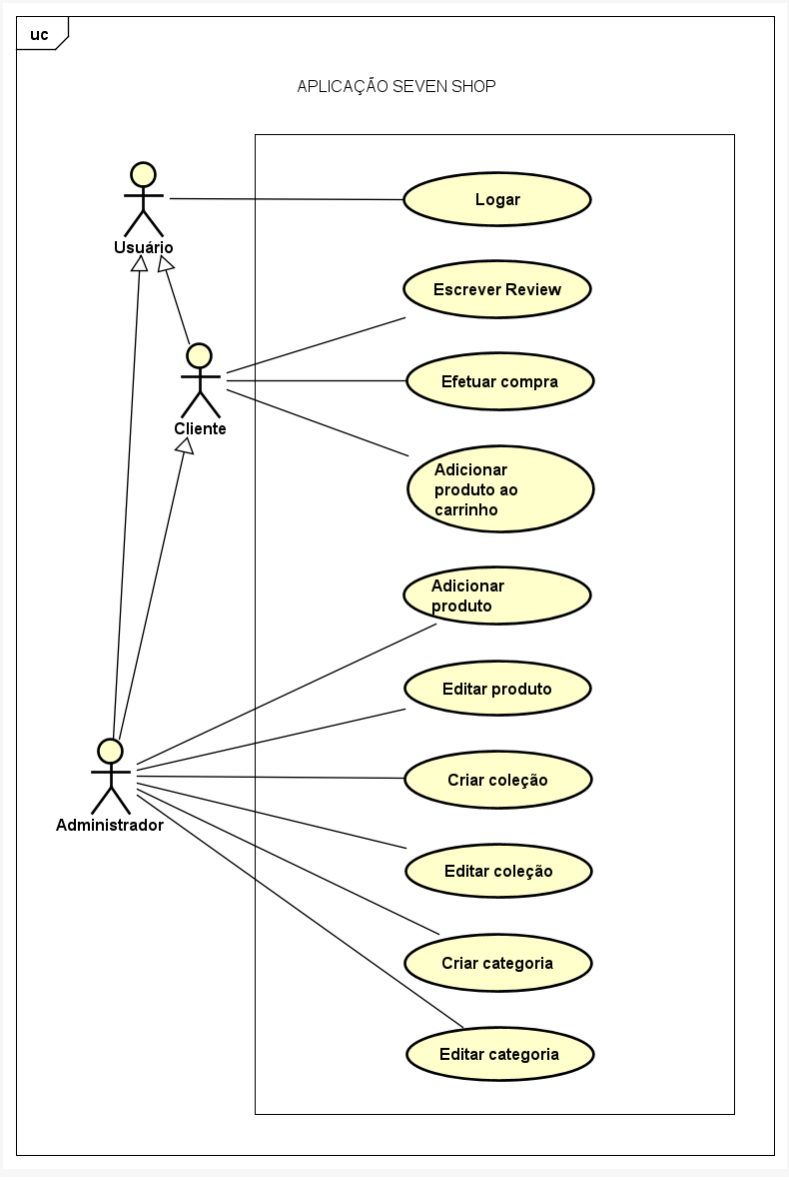
5 ESTUDO DE CASO

Este capítulo visa apresentar um estudo de caso que utiliza a plataforma *OutSystems* para desenvolver um aplicativo de comércio eletrônico para uma empresa fictícia. O objetivo principal é demonstrar por meio deste aplicativo que tipo de aplicações podem ser feitas utilizando *low-code*.

5.1 Diagrama de Caso de Uso

O diagrama a seguir ilustra os casos de uso da aplicação, com seus atores e funcionalidades.

Figura 15 - Diagrama de caso de uso da aplicação.



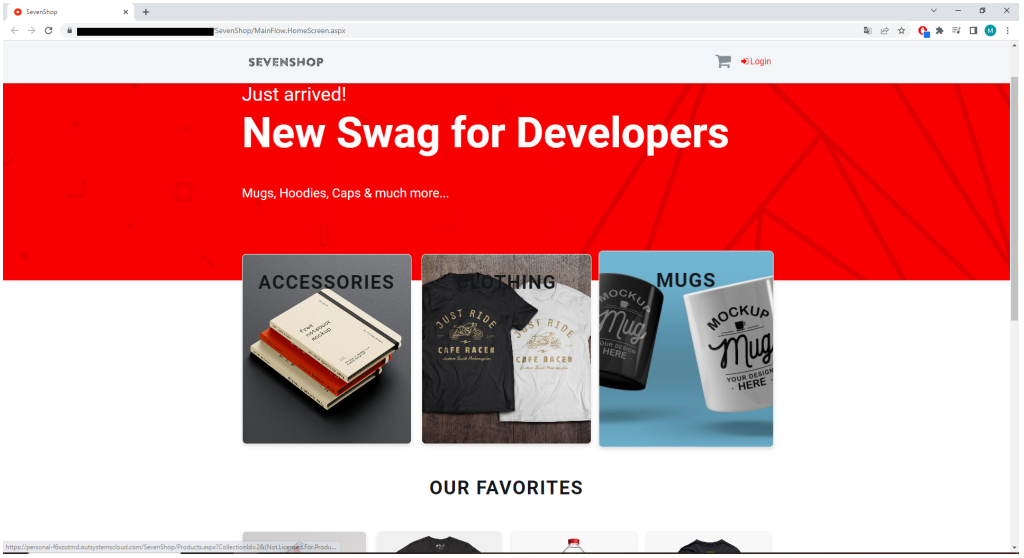
Fonte: Autoria própria.

5.2 Visão Geral do Aplicativo

A aplicação *e-commerce* de venda online criada utilizando a plataforma *OutSystems* possui as seguintes características:

- Página inicial, que apresenta alguns dos produtos da loja.

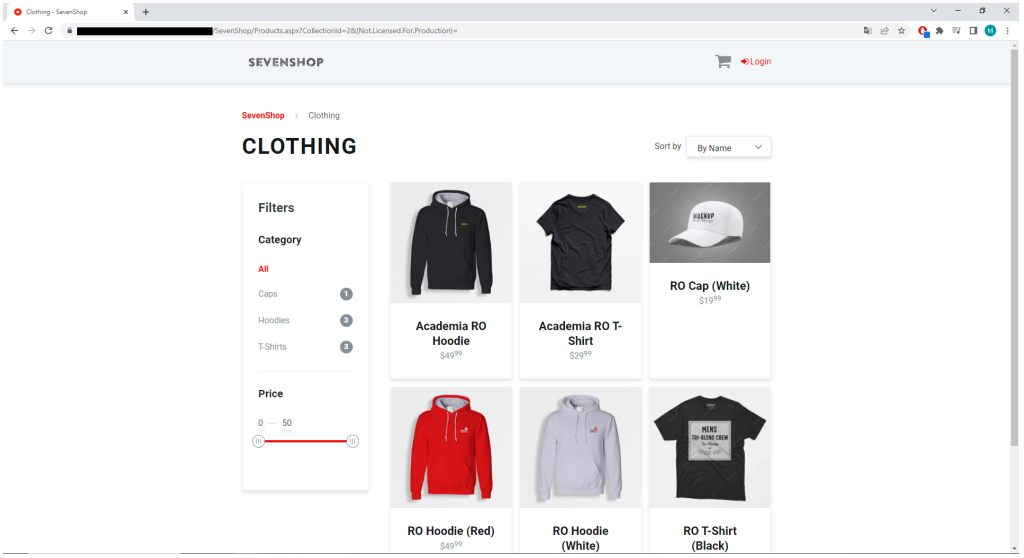
Figura 16 - Tela inicial da aplicação.



Fonte: Autoria própria.

- Página de categorias de produtos e seus respectivos itens.

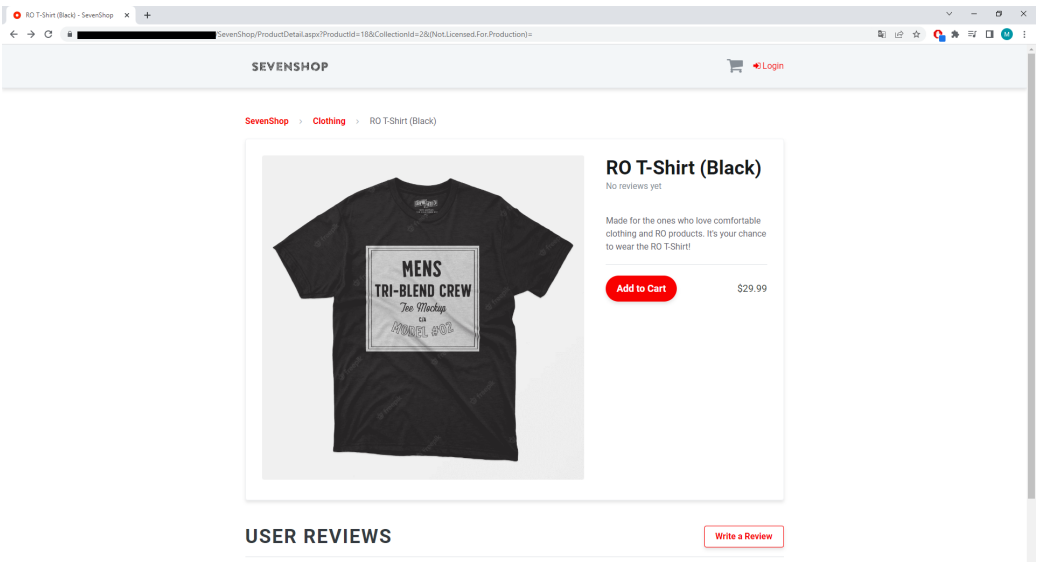
Figura 17 - Tela contendo categorias e produtos.



Fonte: Autoria própria.

- Página de detalhes do produto, que irá abrir assim que um produto for selecionado, possuindo também opções de adicioná-lo ao carrinho de compras ou escrever um review.

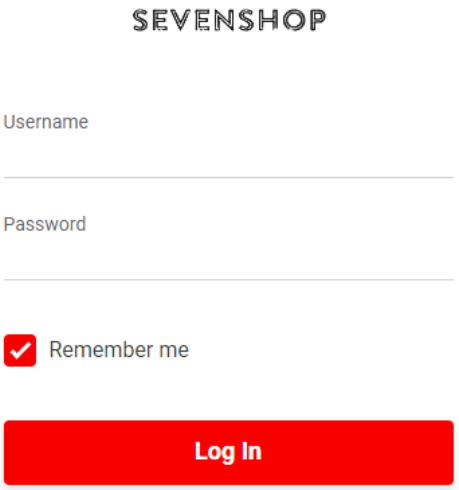
Figura 18 - Tela de detalhes do produto.



Fonte: Autoria própria.

- Tela de login, onde o usuário precisará logar para adicionar produto ao carrinho de compras ou escrever um comentário.

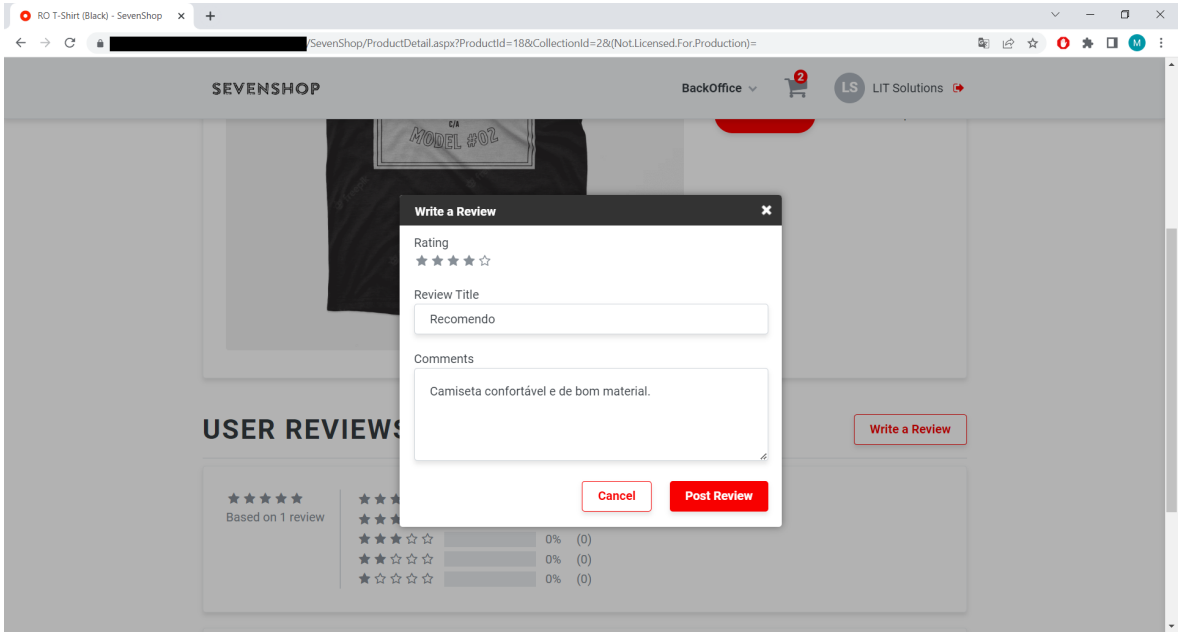
Figura 19 - Tela de login.



Fonte: Autoria própria.

- Popup para escrever *reviews* na página de detalhes do produto.

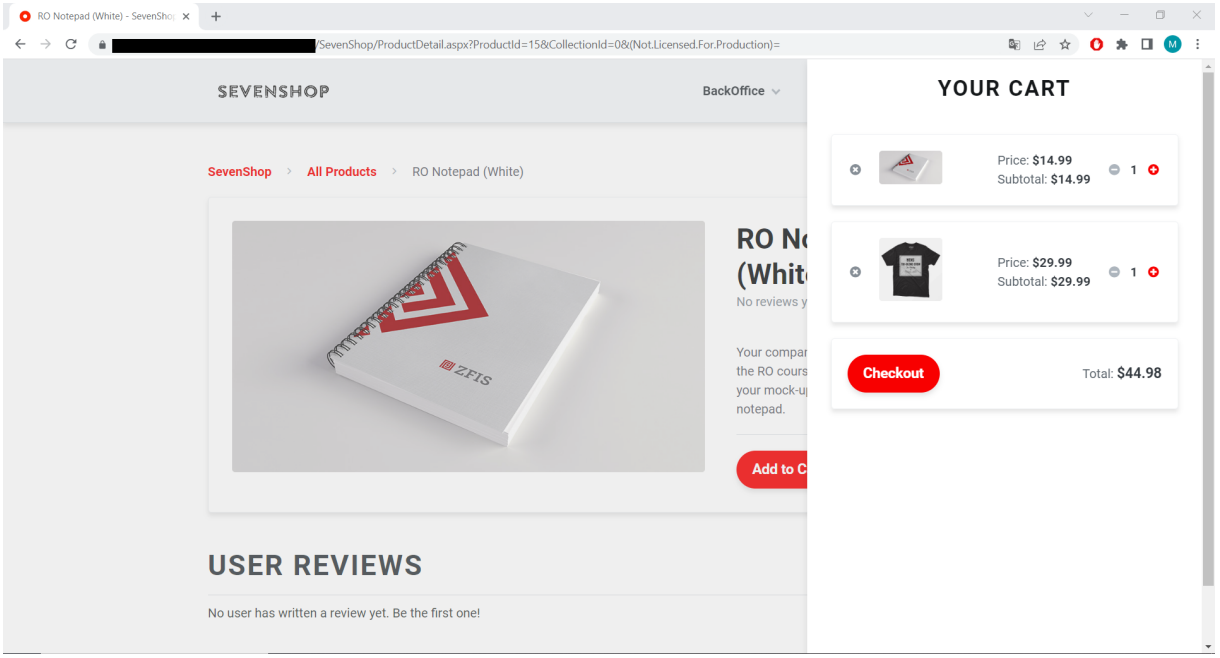
Figura 20 - Postando review de produtos.



Fonte: Autoria própria.

- Carrinho de compras, para o usuário adicionar produtos a lista de compras.

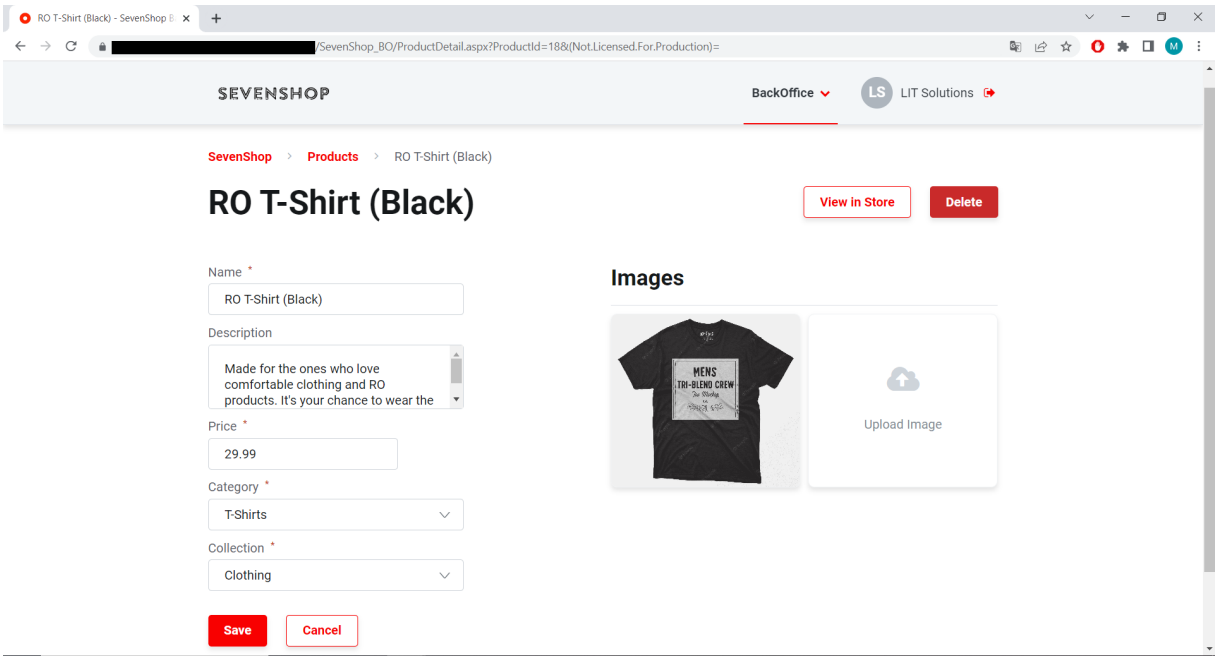
Figura 21 - Carrinho de compras.



Fonte: Autoria própria.

- Página para criação, edição e exclusão de produtos, categorias e coleções pelo usuário administrador. A figura 22 mostra uma dessas opções, no caso edição de um produto.

Figura 22 - Edição de produto.



Fonte: Autoria própria.

5.3 Visão Geral da Arquitetura

Para garantir uma estrutura sólida, manutenção facilitada e capacidade de escalabilidade, a aplicação foi organizada em três camadas distintas. A figura 23 ilustra os módulos desenvolvidos e suas respectivas camadas.

Figura 23 - Arquitetura em camadas dos módulos da aplicação.

Fonte: Autoria própria.

5.3.1 End User

Esta camada é da interface com o usuário final da aplicação. Ela é composta pelos módulos visuais que apresentam as informações e permitem interações do usuário com o sistema. Aqui é onde a experiência do usuário é criada e personalizada. Os desenvolvedores geralmente trabalham nessa camada para criar telas, formulários, relatórios, gráficos e outros elementos visuais.

A aplicação possui os seguintes módulos nesta camada:

1. **SevenShop:**
   Módulo end-user principal da aplicação.
2. **SevenShop_BO:**
   Módulo backoffice da aplicação.

5.3.2 Core

A camada de *"core"* é a camada de negócios da aplicação. Nesta camada, são definidas as regras de negócio, fluxos de trabalho, integrações com sistemas externos e outras funcionalidades de negócio. Os desenvolvedores trabalham nessa camada para implementar a lógica de negócio da aplicação.

A aplicação possui os seguintes módulos nesta camada:

1. **SevenShop_BL:**
   Módulo de regras de negócios que orquestra vários conceitos do aplicativo.
2. **SevenShop_Carts_CS:**
   Mantém os carrinhos de compras do usuário.
3. **SevenShop_Products_CS:**
   Contém Ações e Entidades relacionadas a produtos.
4. **SevenShop_Reviews:**
   Contém ações e entidades relacionadas a avaliações de clientes.
5. **SevenShop_Collections_CS:**
   Contém Ações e Entidades relacionadas a coleções de produtos.
6. **SevenShop_Categories_CS:**
   Contém Ações e Entidades relacionadas a categorias de produtos.
7. **SevenShop_Bootstrap:**
8. Módulo contendo ações de bootstrap que geram dados iniciais do aplicativo.

5.3.3 Foundation

Esta camada é a base da aplicação e contém as funcionalidades compartilhadas, tais como a autenticação de usuários, gerenciamento de sessões, gerenciamento de usuários e permissões. A camada *Foundation* é comum a todas as aplicações desenvolvidas na plataforma OutSystems e é mantida pela própria OutSystems. Os desenvolvedores podem estender a camada de fundação para adicionar funcionalidades adicionais, mas geralmente não precisam trabalhar nessa camada.

A aplicação possui os seguintes módulos nesta camada:

1. **SevenShop_Assets:**
   Repositório contendo imagens e outros recursos binários.
2. **SevenShop_Th:**

Módulo de temas da aplicação.

3. **SevenShop_Roles_Lib:**

   Módulo de controle de permissões de usuário.

6 CONCLUSÃO

Neste capítulo serão abordadas as considerações finais referentes ao tema proposto, juntamente com os desafios enfrentados durante essa jornada em direção à conclusão deste trabalho.

6.1 Considerações Finais

Este trabalho teve o intuito de apresentar uma forma de desenvolvimento de aplicativos alternativa e diferente do tradicional, mas que possibilita que muitas pessoas construam suas próprias aplicações para suas empresas de forma independente e livres de qualquer pré-requisito. E que mesmo sem o usuário possuir experiência ou conhecimentos aprofundados, consiga por meio de uma plataforma *low-code* realizar seus objetivos pessoais e empresariais.

6.2 Trabalhos Futuros

1. Avaliação do desempenho e escalabilidade de aplicações desenvolvidas com *low-code* em comparação com abordagens tradicionais, com ênfase nos resultados obtidos e implicações para a eficiência.

6 REFERÊNCIAS

ABREU, Paul. **High-Code, Low-Code, No-Code, or COTS: How to Choose the Right Approach for Your Business?**. OutSystems, 08 de Jun. de 2022. Disponível em: <https://www.outsystems.com/blog/posts/high-code-vs-low-code-vs-no-code-vs-cots/>. Acesso em: 11 de Abr. 2023.

BABYCH, Maksym. **High-Code, Low-Code, or No-Code: Which Is Right for Your Startup?** What are the differences, and how do you choose the most profitable startup option?. INC, 03 de Feb. de 2022. Disponível em: <https://www.inc.com/young-entrepreneur-council/high-code-low-code-or-no-code-which-is-right-for-your-startup.html#:~:text=High%2Dcode%20or%20traditional%20programming,more%20expensive%20than%20different%20approaches>. Acesso em: 11 de Abr. 2023.

IBERDROLA. **O que é low code ou programação sem código?.** Iberdrola, [2022?]. Disponível em: <https://www.iberdrola.com/innovacao/low-code#:~:text=O%20conceito%20de%20low%20code,aplicativos%20orientadas%20para%20o%20usu%C3%A1rio>. Acesso em: 11 de Abr. 2023.

J.BIGELO. Stephen. **What is low-code? A guide to enterprise low-code app development.** TechTarget. 10 de Jan. 2023. Disponível em: <https://www.techtarget.com/searchsoftwarequality/What-is-low-code-A-guide-to-enterprise-low-code-app-development>. Acesso em: 26 de Mai. 2023.

NERO, Terence. ***Low Code Platform: The Future of Software Development.*** Cuelogic, 04 de mai. de 2021. Disponível em: <https://www.cuelogic.com/blog/low-code-platform>. Acesso em: 11 de Abr. 2023.

NGUYEN, Calvin. **FullStack Development with M.E.R.N Stack: Part 1**. 24 de Out. 2019. Imagem JPEG. Disponível em: <https://levelup.gitconnected.com/a-complete-guide-build-a-scalable-3-tier-architecture-with-mern-stack-es6-ca129d7df805>. Acesso em: 21 de Abr. 2023.

OUTSYSTEMS. ***Architecture.*** *OutSystems,* [ca. 2022a]. Disponível em: <https://www.outsystems.com/evaluation-guide/architecture/>. Acesso em: 07 de Mai. 2023>.

OUTSYSTEMS. ***Getting Started With OutSystems.*** *OutSystems*, [ca. 2022b]. Disponível em: <https://www.outsystems.com/evaluation-guide/who-can-develop-with-outsystems/?sc_lang=en>. Acesso em: 07 de Mai. 2023.

OUTSYSTEMS. ***How long does it take to learn OutSystems?.*** *OutSystems*, [ca. 2022c]. Disponível em: <https://www.outsystems.com/evaluation-guide/how-long-does-it-take-to-learn-outsystems/>. Acesso em: 07 de Mai. 2023.

OUTSYSTEMS. ***Integration.*** *OutSystems*, [ca. 2022d]. Disponível em: <https://www.outsystems.com/pt-br/evaluation-guide/integration/>. Acesso em: 07 de Mai. 2023.

OUTSYSTEMS. ***OutSystems development and management tools.*** *OutSystems*, [ca. 2022e]. Disponível em: <https://www.outsystems.com/evaluation-guide/development-and-management-tools/>. Acesso em: 07 de Mai. 2023.

OUTSYSTEMS. ***Out-of-the-box performance optimization.*** *OutSystems*, [ca. 2022f]. Disponível em: <https://www.outsystems.com/pt-br/evaluation-guide/out-of-the-box-performance-optimization/>. Acesso em: 07 de Mai. 2023.

OUTSYSTEMS. ***OutSystems Security Overview.*** *OutSystems*, [ca. 2022g]. Disponível em: <https://www.outsystems.com/pt-br/evaluation-guide/outsystems-security-overview/>. Acesso em: 07 de Mai. 2023.

OUTSYSTEMS. ***Scalability.*** *OutSystems*, [ca. 2022h]. Disponível em: <https://www.outsystems.com/pt-br/evaluation-guide/scalability/>. Acesso em: 07 de Mai. 2023.

OUTSYSTEMS. ***Standard architecture with no lock-in.*** *OutSystems*, [ca. 2022i]. Disponível em: <https://www.outsystems.com/pt-br/evaluation-guide/standard-architecture-with-no-lock-in/>. Acesso em: 07 de Mai. 2023.

OUTSYSTEMS. ***What Is No-Code?***. *OutSystems*, [ca. 2022j]. Disponível em: <https://www.outsystems.com/glossary/what-is-no-code/#:~:text=No%2Dcode%20is%20a%20software,applet%2C%20or%20other%20software%20elements>. Acesso em: 11 de Abr. 2023.

OUTSYSTEMS. ***Welcome to OutSystems***. *OutSystems*, [ca. 2022k]. Disponível em: <https://www.outsystems.com/company/?sc_lang=en>. Acesso em: 07 de Mai. 2023.

PETLOVANA, Yana.***Traditional Coding vs No-code/Low-code Development.*** *Steelkiwi*. [ca. 2023]. Disponível em: <https://steelkiwi.com/blog/traditional-coding-vs-no-codelow-code-development/>**.** Acesso em: 11 de Abr. 2023.

PERUMALSAMY, Vijay. ***How to send an Email from OutSystems Reactive Web Application?.*** Medium. 09 de Jan. 2023. Disponível em: <https://medium.com/@aalam-info-solutions-llp/how-to-send-an-email-from-outsystems-reactive-web-application-90aa0743fe84>. Acesso em: 21 de Abr. 2023.

TALEB, Simo. ***Everything You Need To Know About Today's Low-Code Architecture***. *SYSTEMS*, [ca. 2023]. Disponível em: <https://www.esystems.fi/en/blog/technology/everything-you-need-to-know-about-todays-low-code-architecture>. Acesso em: 21 de Mai. 2023.

SILVA, Nelson. **Transformação Digital, A 4ª Revolução Industrial.** FGV. Agos. 2018 Disponível em: <https://fgvenergia.fgv.br/sites/fgvenergia.fgv.br/files/coluna_opiniao_-_transformacao_digital.pdf>. Acesso em: 17 de Mai. 2023.

WEBFLOW. ***No-code development: A simple guide to the no-code movement.*** WebFlow, [ca. 2023]. Disponível em: <https://webflow.com/no-code>. Acesso em: 11 de Abr. 2023.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR

Av. Universitária, 1069 ● Setor Universitário
Caixa Postal 86 ● CEP 74605-010
Goiânia ● Goiás ● Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br ● reitoria@pucgoias.edu.br

RESOLUÇÃO n° 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Márcio Antusa da Costa
do Curso de Ciência da Computação, matrícula 2018200280028-4,
telefone: 62 99152-8234 e-mail marcioantusa18@gmail.com,
na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do Autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado DESENVOLVIMENTO LOW-CODE: ESTUDO DE CASO COM PLATAFORMA OUTSYSTEMS - APLICAÇÃO SEVEN SHOP, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto(PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 17 de Junho de 2023.

Assinatura do autor: Márcio Antusa da Costa

Nome completo do autor: Márcio Antusa da Costa

Assinatura do professor–orientador:

Nome completo do professor-orientador: