



## **AVALIAÇÃO DE PLATAFORMAS LOW CODE E DESENVOLVIMENTO DE UM E-LOGBOOK HOTELEIRO USANDO OUTSYSTEMS<sup>1</sup>**

Eliseu Aguiar da Silva<sup>2</sup>

Francisco Assis Moreira do Nascimento<sup>3</sup>

### **RESUMO**

O low code é uma abordagem de desenvolvimento em crescimento na área da criação de softwares que tem em sua principal característica o fato da pouca utilização de código em sua utilidade, tornando-se uma maneira mais rápida de desenvolvimento. Este trabalho analisa três plataformas desse tipo de abordagem e utiliza uma delas para a criação de um software de logbook hoteleiro, que visa otimizar a comunicação interna hoteleira e, com isso, analisar a experiência da usabilidade da plataforma selecionada.

**Palavras-chave:** Plataforma low code, Logbook e hotelaria.

### **ABSTRACT**

Low code is a growing development approach in the area of software creation whose main characteristic is the fact that it uses little code in its usefulness, making it a faster way of development. This work analyzes the platforms of this approach and uses one of them to create a hotel logbook software, which aims to optimize internal hotel communication and, with that, analyze the usability experience of this platform.

**Keywords:** Low code platform, Logbook and hospitality.

## **1 INTRODUÇÃO**

O desenvolvimento de software viu, em 2014, se firmar uma nova ideia da prática de criação de software: o LCD (*Low Code Development*), que despertou bastante interesse dos engenheiros de software pela sua característica de abordagem com pouco código e satisfatória eficiência, pois os softwares mais atuais, que variam entre softwares mais simples até os mais complexos, comportam bastante sistemas em todos os domínios possíveis, com isso a sua produção foi se tornando mais cara e demorada (BUCAIONI,

---

1 Trabalho de Conclusão de Curso.

2 Acadêmico do curso de Sistemas de Informação das Faculdades Integradas de Taquara – Faccat/RS. E-mail: eliseuag@sou.faccat.br.

3 Professor orientador das Faculdades Integradas de Taquara – Faccat/RS. E-mail: assis@faccat.br

CICCHETTI e CICCOSZI, 2021, p.1). Bucaioni, Cicchetti e Ciccozzi (2021, p.1), ainda conceituam que:

O LCD pode ser descrito como uma abordagem visual e semiautomática para o desenvolvimento de software. A ideia básica é que usando o LCD um engenheiro seja capaz de abstrair e semi-automatizar cada etapa do ciclo de vida de um sistema de software e simplificando a entrega de vários sistemas em um domínio específico

Com base neste conceito, é possível compreender que o objetivo das LCDs é, de certa forma, trazer para o mundo da criação de softwares uma abordagem diferente, que otimiza muitas etapas do seu desenvolvimento, auxiliando o desenvolvedor, independente da complexidade do software em desenvolvimento.

Visto isso, este tipo de abordagem teve uma expansão de plataformas que vem sendo um tanto expressiva. Segundo o site da Forum Verkstads (2022), “[...] o analista Gartner prevê que, até 2023, 50% das médias e grandes empresas terão adotado uma LCD (*Low Code Development*) como uma plataforma de aplicativos estratégicos” . Neste artigo, são apresentadas três plataformas Low Code que terão alguns tópicos analisados: Outsystems, Mendix e Cronapp.

O protótipo em questão é um Logbook hoteleiro que vai servir para melhorar a comunicação interna de uma empresa de hospedagem, visto que o tratamento de informações, de forma organizada, gera uma melhor performance de atendimento e prestação de serviço ao cliente.

O objetivo principal do artigo é fazer uma avaliação das três plataformas low code, gerando um comparativo entre elas a partir de pontos relevantes elencados, que receberão uma certa pontuação conforme os critérios determinados, para escolher qual plataforma se adequa melhor no desenvolvimento do projeto. Logo após a escolha, o protótipo será desenvolvido utilizando a plataforma em questão e, com base nisso, serão feitas as análises sobre a experiência de usabilidade de uma plataforma low code.

O artigo é dividido em 8 seções. A seção 2 traz a fundamentação teórica relativa ao tema do projeto. A seção 3 apresenta o desenvolvimento das

análises feitas das plataformas Low Code, assim como suas definições. A seção 4 mostra a parte do desenvolvimento do sistema desde sua fase inicial até sua conclusão. A seção 5 tem como base as funcionalidades do logbook. A seção 6 faz comparação com os trabalhos correlatos, e por fim, a seção 7 discute as conclusões que foram colhidas no decorrer da pesquisa.

## **2 FUNDAMENTAÇÃO TEÓRICA**

Esta seção tem o intuito de discorrer sobre o tema proposto pelo projeto, como também as análises, metodologias e ferramentas utilizadas no mesmo.

### **2.1 Low Code**

O Low Code é um modelo de desenvolvimento de aplicativos e programas que segue o modelo Platform-as-a-Service (PaaS) como serviço de criação de software, ou seja, utiliza plataformas em nuvem para seu gerenciamento e aplicação, e tem por principal característica a dispensabilidade dos tradicionais blocos de código em seu desenvolvimento. Sendo assim, os usuários podem construir aplicativos completamente operacionais interagindo por meio de interfaces gráficas dinâmicas, diagramas visuais e linguagens declarativas (TISI et al., 2019, p. 1). Nos dias de hoje, o Low Code está ganhando campo no mercado de desenvolvimento; sua proposta polêmica e ousada de desenvolvimento rápido e com pouco código traz uma perspectiva desafiadora para os adeptos aos métodos tradicionais de criação de software.

O termo “Low Code” começou a ser utilizado no mercado em 2014 para denominar as plataformas com interfaces baseadas em modelagem gráfica, na forma de uma GUI (Graphical User Interface - Interface Gráfica com Usuário).

Para Tissi et al. (2019, p. 2),

LCDPs (Low Code Development Platform) são construídos com base nos avanços recentes na programação visual, geração de código matemático e infraestrutura em nuvem. LCDPs fornecem ao usuário um ambiente totalmente gerenciado para todo o ciclo de vida do aplicativo, incluindo desenvolvimento, implantação, execução e monitoramento.

Segundo Sanchis et al. (2019, p. 2), as empresas de tecnologia tendem a ter um olhar mais relevante a este tipo de abordagem, pelo fato da exigência emergencial do mercado em querer soluções mais rápidas a todo tempo e, com isso, a rapidez acaba sendo um fator muito importante no contexto atual.

## **2.2 Logbook**

Na área hoteleira e estabelecimentos do ramo de hospedagem, a comunicação interna é um fator essencial para uma gestão eficiente. Sousa (2016, p. 3), discorre que “[..] apenas com uma comunicação efetiva e clara entre os funcionários efetivos e gestores, será possível atingir um padrão de excelência, criando uma imagem positiva do empreendimento no imaginário dos clientes”. Hampton et al. (1990, pág. 68) vai mais além, e ilustra que o que a comunicação faz com uma empresa é como o que a corrente sanguínea faz com os organismos do corpo, suprindo todas as células com oxigênio, assim como a comunicação satisfaz todas as unidades/setores e funcionários de uma organização com informação. Com isso, compreende-se então que a comunicação interna precisa ter uma atenção especial neste segmento, pois ela tem o poder de fazer com que tudo funcione da forma que tem de funcionar e com as informações necessárias para acontecer.

Especificamente, o setor de recepção em hotelaria tem papel fundamental de prestar um serviço de excelente comunicação com o hóspede e com outros setores/funcionários do estabelecimento, fazendo papel de mediador entre estes os dois, o que torna um setor fundamental na prestação de serviços (MENEZES E LOPES, 2020, pág. 4). Dessa forma, dá-se a entender que o ponto central de comunicação de um estabelecimento de hospedagem se encontra na recepção, que é o lugar onde as informações mais importantes são gerenciadas.

Para o tratamento de tantas informações que são anotadas diariamente tem-se a definição de logbook, um livro de registro de eventos que pode ser usado para avaliar situações realizadas, assim como todos os atos relevantes, medidas a serem tomadas e relatados em descrições específicas (THAKUR, KULSHRESTHA e SINGH, 2020, p. 188). Portanto, nesse contexto de

comunicação interna de um estabelecimento de hospedagem, o sistema de logbook se encaixa perfeitamente, pois trabalha exclusivamente com a necessidade de gerenciamento destas informações que são captadas, sugeridas ou repassadas pela recepção, independentemente de seu tipo, podendo ser uma simples anotação de uma ocorrência até um pedido que tem de ser respondido o mais rápido possível.

Por ser um assunto que trata de informações sobre os acontecimentos, eventos e/ou sinistros é de suma importância se ter um sistema que gerencie de forma idônea, pois uma informação crítica que passa despercebida, ou só tem a atenção devida depois que ela já gera muitas consequências, pode acarretar situações de reclamações e/ou situações desconfortáveis para o colaborador e a empresa, que também tem de redobrar o cuidado quando se fala em dados sensíveis. Almeida, Sousa e Mello. (2010, p. 2) explicam que:

Para que os funcionários de uma organização se tornem comprometidos com a entrega de produtos e serviços de qualidade, precisam receber informações que lhes permitam compreender e aceitar seus papéis individuais e coletivos nesse processo, revelando ser essencial o papel da comunicação interna como um instrumento de promoção da qualidade.

Visto isso, pode-se então compreender a importância da comunicação em uma organização, que gera o comprometimento e a qualidade na entrega de um serviço, que é a ideia onde está apoiada a criação deste software. Sendo assim, os valores agregados dessa qualidade de serviço têm um reflexo significativamente positivo na entrega desse atendimento, pois as informações tendem a serem mais tratadas e mais gerenciadas desde a anotação para simples informação de todos até a mais criteriosa.

Também entra em pauta a questão de custos de operação em empreendimentos de pequeno porte, onde um investimento em um sistema de gestão empresarial (ERP - Enterprise Resource Planning) de gestão hoteleira nem sempre é possível. Porém, no segmento hoteleiro, não se pode abdicar desse processo de anotações diárias pelo fato dessas informações serem importantes a qualquer hora do dia. Dessa forma, o projeto auxiliará as empresas de médio/pequeno porte a terem um sistema de comunicação de

forma eficiente com um custo mais em conta pelo fato de não estar intrínseco em um sistema de gestão integral hoteleira.

Obviamente, para se afirmar algo teria que se fazer uma pesquisa com uma quantidade expressiva de usuários. Porém, baseado na experiência do Autor deste projeto, que trabalhou durante 9 anos no setor hoteleiro e que observou como os recepcionistas utilizavam planilhas Excel de forma bastante ineficiente para fazer essas anotações, pode-se afirmar que existe uma grande deficiência na comunicação envolvendo a recepção em hotelaria, pois o uso destas planilhas possibilita pouco tratamento para as informações e seu uso para melhor gestão.

### **3 AVALIAÇÃO DE DESEMPENHO DAS PLATAFORMAS LOW CODE OUTSYSTEMS, MENDIX E CRONAPP**

Para o desenvolvimento do projeto foram escolhidas três plataformas para serem analisadas e avaliadas tanto suas opções quanto às suas funcionalidades para verificar qual seria a melhor solução entre os tópicos apontados. As plataformas escolhidas foram: Outsystems, Mendix e Cronapp. A seguir temos a descrição das plataformas:

- **Mendix**: Plataforma holandesa que foi adquirida pela Siemens em 2018 e, desde então, se destaca no mercado com recursos como *Mendix data hub Catalog*, um repositório de metadados aberto e baseado em padrões que permite que o usuário permeie e encontre recursos de dados no ambiente que está conectado com opções de executar em qualquer ambiente, qualquer sistema operacional, nuvem ou local sendo containerizado por padrão e ainda tem mapeador de dados, Rest nativo e conectores de banco de dados (MENDIX TECHNOLOGY BV, 2021).
- **OutSystems** : OutSystems é uma plataforma nativa em nuvem que utiliza um ambiente de desenvolvimento visual orientado por modelo com assistência baseada em Inteligência Artificial. A OutSystems trabalha com todas as tecnologias existentes (aplicativos, infraestrutura, nuvem e SaaS). A plataforma permite integração com bancos de dados externos, como SQL Server, Oracle, MySQL e iDB2, bem como uma gama de sistemas corporativos de registro (SoR), incluindo SAP,

Salesforce, Microsoft Dynamics 365, Microsoft Dataverse e Sharepoint Online (OUTSYSTEMS, 2021).

**Cronapp:** Plataforma brasileira que trabalha com blocos de programação pré-definidos, que permitem a criação de aplicativos e sistemas com a metodologia de desenvolvimento Low-Code e High-Code. Também conta com diagramação de dados, tecnologia open source, diferentes linguagens de programação disponíveis e licença válida para variados projetos (CRONAPP, 2022).

Como visto no conteúdo das descrições, as três plataformas trazem consigo algumas opções bem relevantes para sua utilização no desenvolvimento de software. Segundo a Gartner (2021), uma empresa de consultoria especializada em tecnologia que faz algumas pesquisas para avaliar líderes de mercado e assim auxiliar em tomadas de decisões, a plataforma OutSystems e a Mendix estão entre as líderes neste segmento, contemplando uma fração relevante desse mercado e, juntamente com elas, temos a plataforma Cronapp, uma empresa nacional promissora e em crescimento neste meio de desenvolvimento tecnológico.

Como componentes para as análises do projeto, foi utilizado um notebook, um desenvolvedor júnior, internet de fibra e três plataformas Low code, já descritas no artigo. Para parâmetros de sistema, foi utilizado o notebook com 4Gb de RAM, SSD de 120Gb e Processador Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz juntamente com o desenvolvedor, o navegador Chrome e a internet de até 600Mb. Para os fatores de sistema, os elementos aplicados foram as três plataformas Low code determinadas no artigo.

Na parte de parâmetros de carga, foi aplicado um CRUD <sup>4</sup> de anotações que teriam todos os campos, devidamente, preenchidos e cadastrados, e nos fatores de carga as próprias inserções que foram feitas aleatoriamente.

As medidas avaliadas foram:

- Tempo 1: tempo (em horas) de preparação do desenvolvedor para uso da plataforma através de cronometragem de trabalho realizado;

---

<sup>4</sup> Acrônimo para Create, Update e Delete, que são as quatro operações básicas utilizadas em uma base de dados.

- Tempo 2: tempo (em horas) de trabalho total do desenvolvedor para implementar a carga de teste (CRUD de anotações);
- Criação do CRUD: Botões de Rádio, Combo, Checkbox e criação automática de CRUD com todas as opções implementadas.
- Layout: quantidade de botões disponíveis, quantidade de tipos de menus e quantidade de templates;
- Recursos disponíveis: organização Documentação, Workflow com Saída modelo BPMN<sup>5</sup>, tipos de arquivos JSON<sup>6</sup>, XML<sup>7</sup> e Xlsx<sup>8</sup> suportados, API aberta, integração com SAP, plugin de alertas e transferência de arquivos, opções de APIs, versionamento, utilização criptografia e suporte Banco de Dados.
- Responsividade: tela máxima, mínima e tela padrão.

Os valores de métricas seguiram o raciocínio de que as métricas de tempo terão uma análise de quanto menor, melhor e ganhará um peso 1 o menor tempo. Em relação aos tópicos quantitativos de botões, menus, templates e quantidade de APIs terão a métrica de quanto mais, melhor e um peso de valor 1 no que representar mais quantidade e 0 nos demais. Os tópicos com valores binários de sim ou não receberão o valor 1 se a resposta for positiva e valor 0 se for negativa. No quesito de organização de documentação, a avaliação se dará em 3 opções: 0 - se for desorganizada e, conseqüentemente, complexa de se entender e receberá valor 0; 1 - se for organizada, porém com um nível de complexidade menor que a anterior e receberá valor 1; e 2 - se for organizada e não for tão complexa de se entender, ficará com valor 2. Na questão de suporte a banco de dados, quem tiver um suporte que comporta o maior número de bancos de dados atendidos ganhará peso 1. Em responsividade, temos 3 opções para cada teste, sendo que a opção 0 representa que não tem nenhuma organização na tela e não tem nenhum peso, a opção 1 representa uma organização com alguns pontos sobrepostos e/ou desorganizados e valor 1, e a opção 2 significa que está tudo

---

5 Business Process Model and Notation. Representação gráfica feita a partir de ícones que simbolizam o fluxo de processo.

6 Acrônimo de JavaScript Object Notation, é um formato compacto, de padrão aberto independente, de troca de dados simples e rápida entre sistemas.

7 Extensible Markup Language. É uma representação gráfica que obedece os critérios de escrituração fiscal para a NF-e

8 Arquivos de extensão de excel.



adequado a página e a nota tem valor 2 também. No final, será feita uma somatória e a plataforma melhor ranqueada será a escolhida. Em caso de empate, a métrica de preparação do desenvolvedor para uso da plataforma será o fator relevante.

A técnica utilizada será de Medição (JAIN, 1991), pelo fato da avaliação exigir um fator de tempo e quantidades de recursos para sua validação.

Com as análises feitas foi gerado o quadro 1 a seguir.

**Quadro 1 - Quadro de análise dos tópicos das plataformas Low Code.**

	<b><i>Outsystems</i></b>	<b><i>Mendix</i></b>	<b><i>Cronapp</i></b>
<b>Tempo 1</b>	01:37 min (0)	00:16 min (1)	01:56 min (0)
<b>Tempo 2</b>	01:36 min (0)	00:46 min (0)	00:22 min (1)
<b>Desenv. CRUD</b>			
<b>Botão Rádio, Combo e Checkbox</b>	Sim (1)	Sim (1)	Sim (1)
<b>Criação automática com todas opções</b>	Não (0)	Sim (1)	Sim (1)
<b>Layouts</b>			
<b>Botões</b>	4 (0)	9 (1)	6 (0)
<b>Menus</b>	4 (1)	2 (0)	2 (0)
<b>Templates</b>	38 (1)	32 (0)	20 (0)
<b>Recursos</b>			
<b>Organização Documentação</b>	2	1	2
<b>Workflow com Saída modelo BPMN</b>	Sim (1)	Sim (1)	Sim (1)
<b>Suporta arquivos Json, XML e Xlsx</b>	Sim (1)	Sim (1)	Sim (1)
<b>Possui API aberta</b>	Sim (1)	Sim (1)	Sim (1)
<b>Integração com SAP</b>	Sim (1)	Sim (1)	Não (0)
<b>Plugin para Alertas</b>	Sim (1)	Sim (1)	Sim (1)

<b>Plugin Transferência de arquivos</b>	Sim (1)	Sim (1)	Sim (1)
<b>Quantidade opções API nativas</b>	17 (0)	13 (0)	16 (1)
<b>Versionamento</b>	Sim (1)	Sim (1)	Sim (1)
<b>Opções nativas de utilização de Criptografia</b>	Sim (1)	Não (0)	Sim (1)
<b>Suporte a banco de dados</b>	DB2 iSeries, PostgreSQL e Aurora PostgreSQL, MySQL, Oracle, SQL do Azure e SQL Server e MongoDB (1)	DB2, MySql/MariaDB, Oracle e SAP HANNA (0)	Amazon RDS ( MySQL, PostgreSQL, MariaDB, Oracle BYOL ou SQL Server). (0)
<b>Responsividade</b>			
<b>Tela máxima</b>	2	2	2
<b>Tela mínima</b>	2	1	1
<b>Tela padrão</b>	2	2	2
<b>Resultado</b>	<b>20</b>	<b>17</b>	<b>18</b>

fonte: Autor, 2022.

Como descrito na tabela acima, da perspectiva do Autor, a plataforma que mais atendeu aos tópicos elencados foi a plataforma OutSystems que, por sinal, é uma das plataformas líderes do mercado deste modelo de programação. Ela se mostrou bem completa em relação às opções oferecidas para ser utilizada no projeto de Logbook.

A plataforma Mendix tem uma vasta opção também, porém em tópicos de documentação, onde o desenvolvedor encontrou alguma dificuldade em encontrar conteúdos objetivos, e em responsividade de tela mínima, onde alguns campos não foram totalmente organizados para serem exibidos em sua totalidade por exemplo, teve a sua pontuação afetada negativamente.

A Cronapp se mostrou com bastante recursos, perdendo alguns pontos, por exemplo, o fato de sua responsividade, nas opções de bancos de dados suportados e complexidade de UX da plataforma .

## **4 DESENVOLVIMENTO DO SISTEMA**

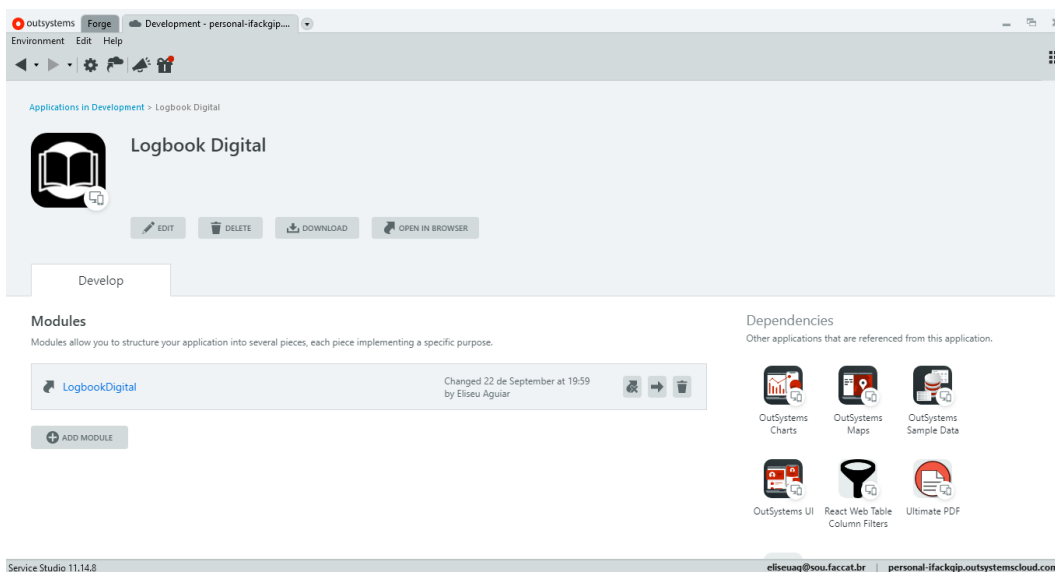
Neste capítulo, são apresentados as especificações e o detalhamento do software desenvolvido, detalhando as suas características, técnicas e ferramentas utilizadas no seu desenvolvimento, a operacionalidade da implementação e os resultados obtidos.

### **4.1 Ferramentas e tecnologias do desenvolvimento do sistema**

Para o desenvolvimento do projeto, foi utilizado o serviço em nuvem da Outsystems para cadastro e configurações iniciais e o Outsystems Service Studio 11 (Rapid Application Development with OutSystems Create applications with OutSystems), que é o ambiente de desenvolvimento Low Code (IDE) desktop da plataforma, totalmente orientado ao paradigma de desenvolvimento visual e é acessada através das credenciais de servidor e login e senha. Este ambiente de desenvolvimento permite criar aplicativos e módulos no servidor, definir os modelos de dados e processos de negócios, assim como temporizadores, conectar com banco de dados externos e também cria os códigos C# (Aprenda programar com C#) e .Net (Desenvolvimento Web com Asp.net MVC) nativo para a usabilidade em sistemas legados; também faz a depuração e compilação com base nos códigos em que abrange, como HTML, CSS, JavaScript.

A Outsystems também oferece uma ferramenta que permite estender a plataforma com código C# / .NET personalizado para a efetuar integrações com maior grau de complexidade e não padronizado, a Outsystems Integration Service. Contudo suas criações ficam acessíveis para serem utilizadas como dependências no Service Studio (PEREIRA, 2022, pg 44). Na figura 1, é possível visualizar a tela de inicialização da interface do Outsystems Service Studio 11 com o módulo do logbook digital.

**Figura 1 - Interface Outsystems Service Studio 11**



fonte: Autor, 2022.

Como base para gerenciamento de etapas do projeto, foi utilizada a plataforma do Trello (TRELLO, 2022), que oferece opções de organização de informações para esse fim.

## 4.2 Gestão do projeto usando Scrum Solo.

Scrum solo é uma adaptação da tradicional metodologia Scrum (PRESSMAN, 2011), que normalmente envolve mais de uma pessoa no seu desenvolvimento, pois há papéis de responsabilidades diferentes a serem desempenhados no processo natural dessa metodologia ágil. A metodologia do Scrum solo, no entanto, foi pensada para ser utilizada por uma pessoa somente, pois há muitos desenvolvedores que atuam sozinhos em seus projetos, sendo responsáveis por 10% da produtividade total desse mercado (PAGOTTO, FABRI E L'ERÁRIO, 2016, p. 1). Conforme Pagotto, Fabri e L'erário (2016, p. 3), o Scrum Solo é um método iterativo e incremental que combina as melhores práticas definidas no Processo de Software Pessoal (PSP) — um processo de melhoria que foi criado com intuito de ajudar o desenvolvedor a administrar melhor a construção de software — e pela metodologia Scrum. Sendo assim, o cenário se encaixa perfeitamente com o

contexto do projeto em questão. Com base na figura 2, a seguir, é possível analisar algumas relações do Scrum solo ao processo Scrum tradicional.

**Figura 2. Ilustração processo Scrum solo.**



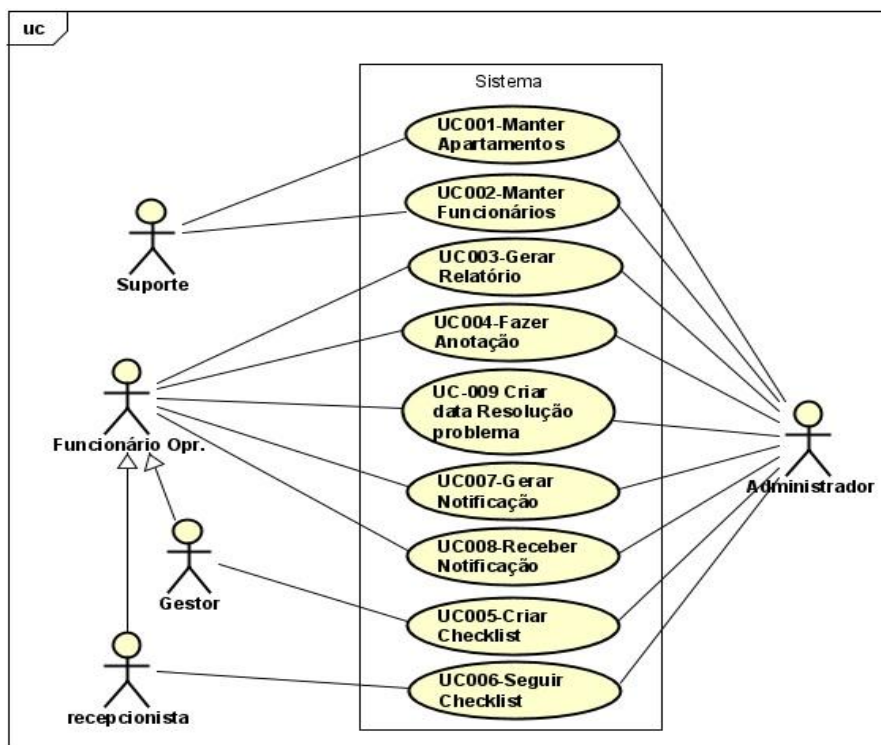
fonte: PAGOTTO, FABRI e L'ERÁRIO, 2016.

O Scrum solo tem muitas semelhanças com o tradicional Scrum. O Product Backlog e a Sprint, por exemplo, são iguais. Este tipo de metodologia procura otimizar o processo de utilização que tem como atividades principais quatro etapas: definição de requisitos, sprint, implantação e gerenciamento; onde a definição de requisitos estabelece o escopo do produto, o backlog e um protótipo de software; todas essas etapas serão alcançadas com a ajuda do cliente e de um consultor, que auxilia o desenvolvedor para concluir seu objetivo. A sprint, por sua vez, é reduzida a uma semana geralmente, e comporta um gerenciamento de status que é analisada e discutida pelos envolvidos com o objetivo de gerar uma estrutura analítica do projeto com cronogramas e tabela de custos. Por fim, na implantação o produto é disponibilizado ao cliente para utilização (BRITO, REBELO e BRITO, 2020, p. 3 - 5).

### 4.3 Análise orientada a objetos

Nas imagens das figuras 3 e 4, a seguir, tem-se os diagramas de casos de uso e de domínio do protótipo Logbook.

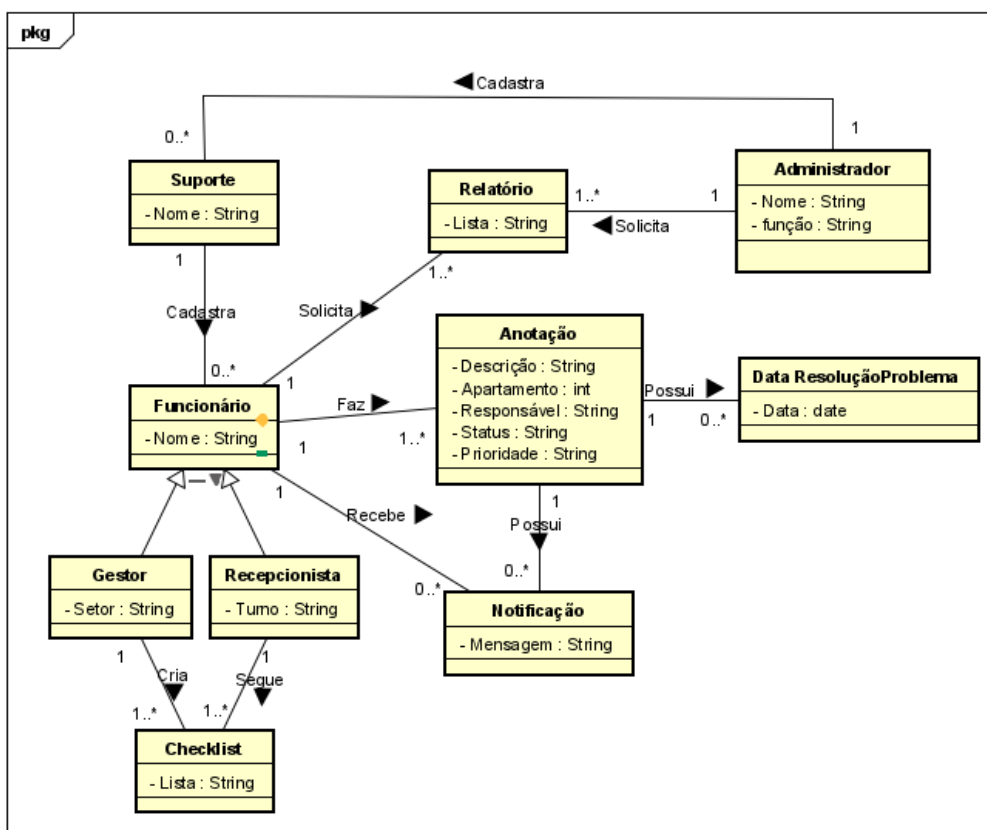
Figura 3. Diagrama de casos de uso.



fonte: Autor, 2022.

Na figura 3, tem-se a indicação de acessos de cada personagem do sistema que comporta a função de administrador do sistema, onde teria o empresário do estabelecimento como exemplo; o suporte, exemplificado pelo setor de TI da empresa; o gestor, que administra uma equipe no ambiente empresarial e o recepcionista, que executa a função de inserção e manipulação das informações no sistema.

Figura 4. Diagrama de domínio.



fonte: Autor, 2022.

O diagrama de domínio (LARMAN, 2007) ilustra a associação que os atores do sistema tem com as funcionalidades oferecidas pelo protótipo. Posto isso, pode-se identificar que a pessoa do administrador tem um papel inicial de cadastro, que logo após, pode ser exercida também pela pessoa do suporte, visto que o suporte técnico, em contato com administrador, cadastra os funcionários e apartamentos em atividade. Logo mais, é possível ver que a principal função de descrever anotações e suas opções, pode ser acessada pelo recepcionista e pelo gestor, que também tem acesso aos relatórios e pesquisas de anotações. Por fim, para a ilustração do papel do gestor tem-se a criação das checklists de atividades e o recepcionista seguindo e concluindo a mesma.

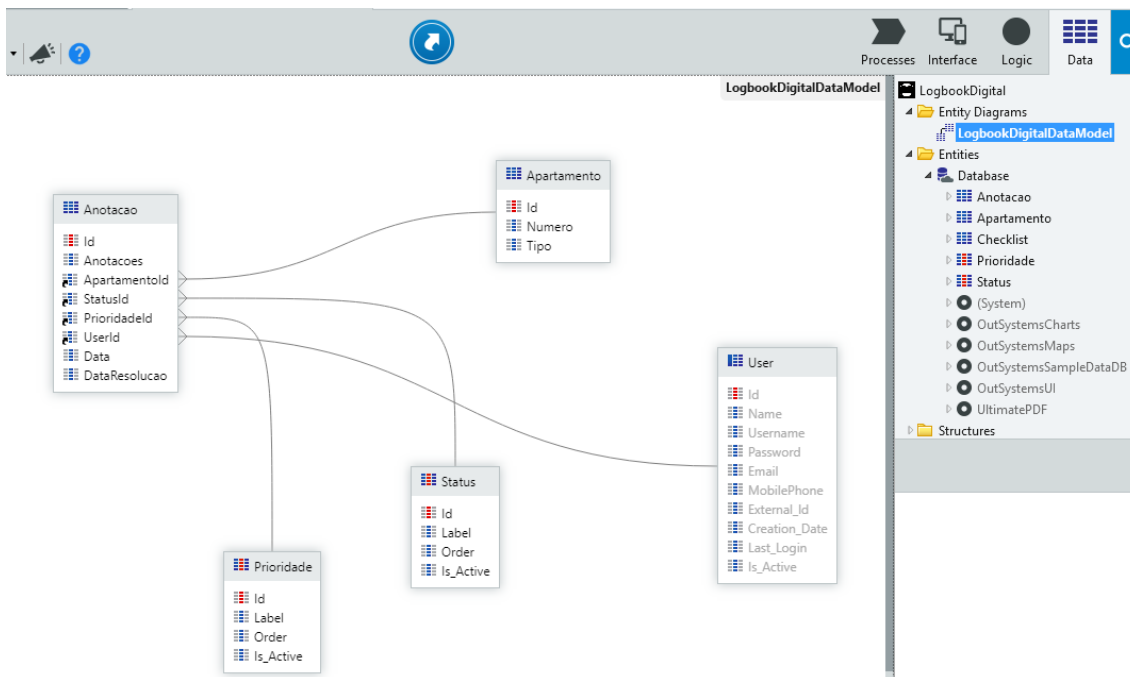
#### 4.4 Projeto usando a plataforma Outsystems

A criação da base de dados na plataforma é um processo muito otimizado, pois o modelo de desenvolvimento drag-and-drop, que é utilizado

pela plataforma, oferece uma maior facilidade para modelar os dados do sistema.

Logo mais abaixo, a figura 5 mostra a modelagem dos dados feita na plataforma para a organização e população de dados do protótipo Logbook.

**Figura 5. Ilustração da montagem da base de dados do protótipo Logbook.**



fonte: Autor, 2022.

Com a montagem das tabelas para o Banco de dados criadas na aba *Data*, a plataforma Outsystems disponibiliza um campo central, onde se pode visualizar melhor as tabelas, com seus atributos, e os relacionamentos feitos entre elas. Para algumas situações de população de dados massivamente de forma externa, a Outsystems oferece recursos de aceleração, que consiste em preencher as entidades, ou até mesmo criá-las, a partir de um determinado arquivo excel, por exemplo (PEREIRA, 2022, pg. 175).

#### 4.5 Construção usando os recursos do Outsystems

Nem sempre todos os comportamentos desejados para se ter em um aplicativo web/mobile se encontra em uma plataforma Low Code, é então que



entra o sentido literal do prefixo do tipo da abordagem da plataforma, pois alguns pontos mais específicos do projeto, muitas vezes, tem de ser contemplados e não se tem a pronta entrega. Pereira (2021, p. 295), discorre que, “[...]Podemos criar nosso próprio código personalizado para um determinado evento exatamente no fluxo que precisamos em nossas ações de cliente[...]. Com isso, tem-se um poder muito maior para lidar com os comportamentos esperados pelo aplicativo, aprimorando e ajustando a UX de nossos aplicativos para atender às necessidades dos usuários”. Também, como descrito no tópico 4.1, tem-se o Outsystems Integrations Service, onde podem ser criados/importados plugins/ferramentas para serem utilizadas como dependência pelo Studio Service no desenvolvimento.

No protótipo do sistema Logbook, foram utilizados alguns blocos de códigos Javascript com objetivos distintos, como de atualizar a tela, pois ela tem a necessidade de atualizar de forma simultânea em todas as páginas abertas e para a funcionalidade de filtros na parte de relatórios. Os condicionadores if/else também foram utilizados, porém para essas situações a plataforma disponibiliza esses condicionais em ícones e uma visualização da página com os elementos mostrados e segmentados em modelo de árvore, sendo assim a alocação das condições se dá de maneira mais simples.

#### **4.6 Testes usando o Outsystems**

Esta subseção apresenta os testes de software (SOMMERVILLE, 2011) que foram realizados com as funcionalidades do sistema de forma manual.

No teste funcional primário, que foram os testes funcionais de cadastros, apartamentos e funcionários, os apartamentos foram cadastrados conforme seu número e tipo, armazenados através da tela de detalhes do apartamento e disponibilizados na tela de cadastro de apartamentos, onde se tem a opção de edição ou exclusão do mesmo. Na sequência de cadastro, os funcionários foram cadastrados através da tela de detalhes também e as permissões que cada um vai ter dentro do sistema foram dadas a partir da tela de administração do sistema. Assim como a tela de apartamentos, os funcionários ficaram armazenados e disponibilizados na tela de cadastro de funcionários

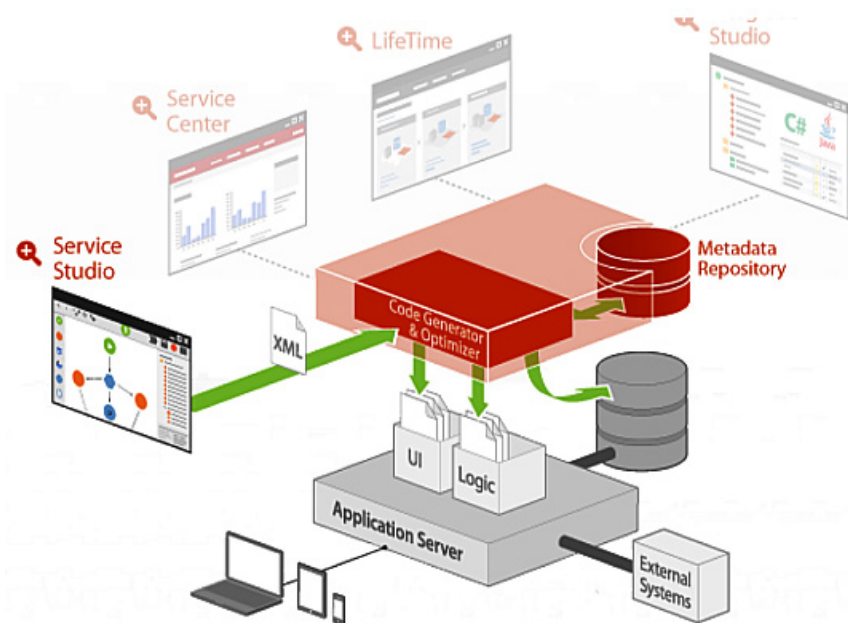
com opção de edição e exclusão. Todos os testes de cadastro obtiveram resultados positivos.

Na parte de anotações, foi feita inserção de informações na página de anotações de forma sequencial, os dados foram armazenados através de todas as informações da página de detalhes das anotações, acessados pelo botão de inserção de anotações presente na página, e as informações ficaram disponíveis na página de anotações. Na página de relatórios, foram feitos testes de pesquisas de informações em todos os filtros disponíveis. Por fim, na parte de Checklist, se teve a criação de mais de um tópico de checagem feita pelo gestor e a visualização e conclusão feita pelo recepcionista, todos obtendo resultados positivos.

#### 4.7 Implantação

Tendo em vista que a plataforma OutSystems otimiza e torna menos complexa algumas atividades de desenvolvimento, com a implantação não seria diferente. Na figura 6, é ilustrado como se dá a publicação de aplicativos na plataforma OutSystems.

Figura 6 - Fluxo de publicação de aplicativos OutSystems do Service Studio



fonte: PEREIRA, 2022.

Como mostrado na figura 6, durante o desenvolvimento de um aplicativo, assim como na hora de sua conclusão, o Service Studio disponibiliza um botão acima da área de trabalho da plataforma para ser publicado o software do projeto em desenvolvimento. Acionado o botão de publicação, o processo de depuração e compilação é iniciado, se tudo estiver correto, o botão modifica seu ícone para mostrar ao desenvolvedor que tudo está pronto e que a publicação pode ser feita.

Neste contexto, a mágica da montagem da página na tela, via browser, é feita com os dados sendo levados ao gerador e otimizador de códigos. A plataforma consegue ainda oferecer esta funcionalidade de publicação com 100% de disponibilidade, com o chamado tempo de inatividade zero. Com isto, os aplicativos não ficam com interrupções durante o processo de publicação, pois os arquivos necessários para publicação são armazenados em um diretório virtual. Somente quando a operação é concluída, o diretório antigo é substituído totalmente pelo novo diretório.

Para ter esse desempenho, a Outsystems tem um servidor denominado Servidor de controlador de implantação. A plataforma da Outsystems é composta por um conjunto de servidores (que podem ser instalados na nuvem ou como on-premises) que realizam todas as operações importantes para que as aplicações desenvolvidas sejam oferecidas com qualidade aos usuários.

Na figura ilustrada, nota-se também a existência do *Service Studio*, que é um ambiente que oferece alguns monitoramentos do software, assim como logs para se entender e interpretar possíveis erros que possam ocorrer no processo de publicação do software, com o intuito de ser mais eficaz em sua depuração. Além disso, tem-se a *Lifetime*, um console da Web que controla, monitora e configura o ciclo de vida integral dos aplicativos nos ambientes implementados, permitindo, entre outras coisas, o rastreamento de versões, inúmeras verificações de integridade para garantir a robustez do ambiente e personalização de Tags no momento da implantação (PEREIRA, 2022, p. 55-118-139).

## 4.8 Manutenção do sistema

Para qualquer alteração referente ao protótipo, o administrador da conta tem as credenciais necessárias para acessar a conta em nuvem e, através do service studio, que disponibiliza todos projetos disponíveis para o plano da conta do administrador, fazer quaisquer alterações.

## 5 UTILIZAÇÃO DO LOGBOOK

A seção a seguir descreve o uso das funcionalidades do protótipo Logbook.

### 5.1 Cadastros

As figuras 7 e 8 demonstram as páginas de cadastro de apartamentos e funcionários, respectivamente.

Figura 7-Página cadastro funcionários.

A captura de tela mostra a interface de usuário para o cadastro de funcionários. No topo, há um menu de navegação com os itens: LogbookDigital, Cadastro Apartamentos, Cadastro Funcionários e Painel Administrativo. O título principal da seção é "Cadastro funcionário". Abaixo, há um formulário com os seguintes campos: "Nome" com o valor "Angelo", "Username" com o valor "Angelo@hotel", e "Senha" com caracteres ocultos por pontos. Há também uma opção "Usuário Ativo" com uma caixa de seleção marcada. Um botão "Salvar" está localizado na base do formulário.

fonte: Autor, 2022

Figura 8-Página cadastro apartamentos.

A captura de tela mostra a interface de usuário para o cadastro de apartamentos. No topo, há um menu de navegação com os itens: LogbookDigital, Cadastro Apartamentos e Cadastro Funcionários. O título principal da seção é "Cadastrar Apartamento". Abaixo, há um formulário com os seguintes campos: "Número" com o valor "314" e "Tipo" com o valor "Master". Há dois botões: "Voltar" e "Salvar".

fonte: Autor, 2022

Nas informações adicionadas de apartamentos, tem-se o número do apartamento e o tipo de sua categoria. Já no cadastro de funcionários, as

informações a serem armazenadas são as do nome, para a sua identificação no sistema; username e senha, para ser utilizado no login da software e a opção indicando que é um usuário ativo, pois pode ser que futuramente o administrador do sistema não queira excluir o funcionário, mas ele, por algum motivo, não pode mais acessar o sistema; sendo assim o administrador tem a opção de deixar este funcionário inativo e pode ativar novamente a qualquer momento.

## 5.2 Relatórios

A figura 9 ilustra a parte de pesquisa de informações e relatórios.

**Figura 9 - Página Relatórios.**

Anotações	Apartamento	Responsável	Status	Prioridade	Data
Feito teste do celular e foi visto**44		Assis	Andamento	Alta	18 Jul 2022
Manutenção pediu par nivelar as agua do Boiler	325	Kaike R.	Fechado	Media	17 Jul 2022
teste8	159	fulano	Andamento	Media	23 Jul 2022
teste TCC	259	-	Andamento	Alta	23 Jul 2022
testetacc rapido	159	Assis	Andamento	Media	23 Jul 2022

fonte: Autor, 2022.

Na aba de filtros e relatórios do sistema, o usuário pode pesquisar uma determinada informação por cada coluna em particular e baixar em arquivo PDF o resultado da busca, com isso, tem o acesso ao relatório de anotações de uma determinada data, ou por determinado apartamento, por exemplo.

## 5.3 Anotações

Logo mais abaixo, na figura 10, temos a ilustração da página principal do protótipo, que é a de anotações, onde o autor da anotação tem que preencher

todos os dados dos campos disponibilizados para que a informação esteja completa.

**Figura 10 - Página anotações.**

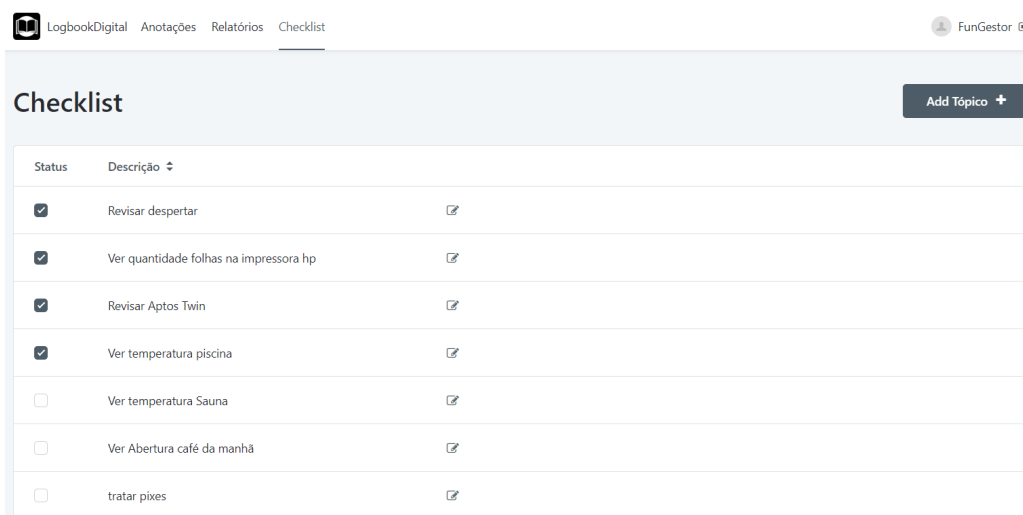
Anotações	Apartamento	Responsável	Status	Prioridade	Data	Data Resolucao
Feito teste do celular e foi visto**44		Assis	Andamento	Alta	18 Jul 2022	28 Oct 2022
Manutenção pediu par nivelar as agua do Boiler	325	-	Fechado	Media	17 Jul 2022	30 Oct 2022
Bouquet de flores chegou para apartamento x e o Sr. Fulano ficou de pegar antes d eir embora.	159	fulano	Andamento	Media	23 Jul 2022	15 Oct 2022

fonte: Autor, 2022.

Como visto na figura 10 acima, as anotações têm de ser descritas com indicação de apartamento, seja indicando onde o hóspede está localizado ou simplesmente onde a informação está direcionada; de responsável, indicando a pessoa que vai ter alguma ação perante aquela informação ou que está envolvida no sinistro descrito; de status, para o usuário poder saber se a situação descrita está em processo de resolução ainda ou já teve algum desfecho.

## 5.4 Checklists

Por fim, temos a página de checklists do protótipo, onde o recepcionista vai preencher os tópicos listados abaixo, criado pelo gestor, conforme vai concluindo as tarefas.

**Figura 11 - Página de Checklist.**

fonte: Autor, 2022.

Como visto, a página de checklist tem por objetivo elencar tópicos de ações para serem feitas antes, durante ou no fim de algum turno ou período previsto, para efeito de preparo e/ou organização do ambiente de trabalho. Ela tem como criador/editor a figura do gestor e como executor a figura do recepcionista.

## 5.5 CONSIDERAÇÕES FINAIS SOBRE O LOGBOOK

O logbook digital é um software pensado para atender uma demanda muito importante dentro de uma empresa que precisa tratar informações de forma instantânea. Seu desenvolvimento foi feito utilizando uma plataforma que auxilia muito o desenvolvedor em muitos aspectos de programação. Algumas das poucas dificuldades que foram encontradas, se deram pelo fato de comportamentos específicos importantes, que precisavam ser executados pelo software e não estavam na biblioteca das funcionalidades pré prontas, mesmo assim, a plataforma tem uma gama de plugins externos e uma comunidade muito ativa, que auxiliou em muitos momentos de complexidade.

## 6 TRABALHOS CORRELATOS

A proposta em questão do logbook sugerido tem o objetivo de melhorar a passagem de informações e a comunicação interna de uma recepção hoteleira. A partir de uma pesquisa na web, foi encontrada uma variedade bem expressiva de softwares de logbook em muitos segmentos, pois sua

característica de armazenamento e tratamento de informações é um diferencial muito importante para qualquer empreendimento. Dentre estes sistemas correlacionados, foram reunidos três deles, que se mostraram mais representativos. São eles:

- **LogBook, Software de diário de bordo:** Sistema de anotações diárias com funcionalidades que visam a comunicação de turnos do setor da indústria com maquinários robustos como montadoras e fábricas de produção em geral (LOGBOOK, 2021).
- **HubWorks, Zip Shift Book:** Sistema diário de bordo que segue o viés de automatização de gestão e serviços de restaurantes de todo tipo (HUBWORKS).
- **ShiftNote:** Sistema de Logbook gerencial, que foca na organização e otimização de serviços de gerentes de variados setores com funcionalidades de previsão de vendas, relatórios, lista de controle, entre outros (SHIFTNOTE, 2007-2020).

O parágrafo anterior, mostrou alguns softwares de logbook com intuito tratar informações e instigar a melhoria na comunicação também, como todo diário de bordo se propõe a fazer. Porém, cada um abrange um segmento particular. A título de exemplo, tem-se o primeiro item da lista Logbook, que tem o viés de auxílio a informações industriais, que busca estabelecer um padrão para entradas de log consistentes em toda a organização, para garantir as informações necessárias em uma auditoria ou inspeção, atividades comuns no ramo industrial.

O Zipshift tem uma visão mais em organizações gastronômicas para seu foco empresarial, com o objetivo de utilização de seus recursos para rastrear atividades diárias, reparos, mudança de horário de última hora, prevenção de acidentes e custos, enfim. Também conta com registros de feedbacks de clientes para análise de satisfação.

E, por último, o Shiftnote, que traz mais um olhar em gerências de vendas, onde o intuito é fazer gerentes e/ou responsáveis ficarem sincronizados por meio de notas e tenham rastreamento de estatísticas, bem como também criar cronogramas e evitar horas extras de seus colaboradores.

Todos sistemas agregam, e muito, seus respectivos segmentos. Contudo, em relação a hotelaria, o logbook proposto se destaca pelo fato do



tratamento diferenciado dado às informações, desde a descrição da situação, do status associado, do apartamento em questão, do responsável atribuído, da prioridade definida e do dia proposto a ser resolvido o sinistro; tópicos que não são relacionados acima pelos outros softwares e que auxiliam muito o trabalho do colaborador hoteleiro.

## 7 CONCLUSÃO

Em resumo, o desenvolvimento de um software através de uma plataforma low code oferece muita agilidade durante a criação de algumas funcionalidades, pelo fato de se ter uma gama de opções pré-configuradas para auxiliar na rapidez do desenvolvimento. Com isso, pode-se dizer, assim como o criador de conteúdo sobre low code, escritor e idealizador do Baixada Nerd, Léo Andrade (LEO ANDRADE, 2021) — especialista com quem o presente Autor teve uma conversa para compreensão de alguns tópicos relacionados ao tema — que este tipo de abordagem é um acelerador de desenvolvimento de software que, literalmente, oferece opções prontas de funcionalidades repetitivas de forma mais rápida e pontual. Além do mais, por ser Low code, a plataforma ainda disponibiliza opções de customização que podem ser inseridos no software através de blocos de códigos de linguagens específicas de programação, com isso trouxe uma melhor experiência ao desenvolvedor, que otimizou tempo em desenvolvimento de CRUD's e conexões com banco de dados, por exemplo.

Um quesito importante a deixar esclarecido também, que foi notado neste projeto, é que as plataformas Low Code exigem conhecimentos lógicos de programação para serem utilizadas, pois, a título de exemplo, seu desenvolvimento envolve a aplicação de tratamentos como modelagem de dados, uso de condicionais e noções de tipos de variáveis, entre outros.

Para trabalho futuro, fica o avanço em algumas funcionalidades tais como, notificações ao usuário e aprimoramento de relatórios, que são funcionalidades importantes.

Assim como mais um comparativo entre tópicos de desenvolvimento low code do protótipo com um desenvolvimento tradicional, com isso, trará parâmetros relevantes e irá agregar muito ao projeto, que infelizmente, não

pode atender a estas opções por questões de tempo disponível e do escopo ter se revelado extenso para um TCC.

## 8 REFERÊNCIAS

ALMEIDA, Lilian Maria de Souza; SOUZA, Luiz Gonzaga Mariano de; MELLO, Carlos Henrique Pereira. A comunicação interna como um instrumento de promoção da qualidade: estudo de caso em uma empresa global de comunicação. **Gestão & Produção**. Gest. Prod., São Carlos, v. 17, n. 1, p. 19-34, 2010. DOI: 10.1590/S0104-530X2010000100003 . Disponível em: <http://www.scielo.br/pdf/gp/v17n1/v17n1a03.pdf>. 2010. Acesso em: 21 Out. 2021.

BRITO, João N.; REBELO, Carlos; BRITO, Miguel A.. **Scrum solo application in a project with a strong integration component**. Publicado em Conference on System Sciences (HICSS'07), Waikoloa, HI, 2007, pp. 274a-274a, doi: 10.1109/HICSS.2007.180. 2020. Disponível em: <https://repositorium.sdum.uminho.pt/bitstream/1822/71816/1/36th%20BIMA%20Camera%20Ready%2030%20v4.pdf>. Acessado em: 26 Set de 2022.

BUCAIONI, Alessi; CICCHETTI Antonio; CICOZZI, Frederico. Modelling in low-code development: a multi-vocal systematic review. **Springer link**. Suécia, 19 janeiro de 2022. DOI:<https://doi.org/10.1007/s10270-021-00964-0>. 2022. Disponível em: <https://link.springer.com/article/10.1007/s10270-021-00964-0>. Acessado em 21 Set de 2022.

CRONAPP, **Funcionalidades**. © 2022 Cronapp. Disponível em:<https://www.cronapp.io/recursos>. Acesso em: 29 Ago 2022.

**Forum Verkstads**. LOW-CODE PLATFORMS – hot solutions expected to be doubled within two years: Mendix visionary market leader according to Gartner. Disponível em: <https://plm-erpnews.se/low-code-platforms-hot-solutions-expected-to-be-doubled-within-two-years-mendix-visionary-market-leader-according-to-gartner/>. 2021. Acessado em: 21 Set de 2022.

GARTNER, **Gartner**. © 2021 Gartner, Doc. Disponível em: <https://www.gartner.com/doc/reprints?id=1-27I1J5FE&ct=210923&st=sb>. Acesso em 30 Out. de 2021.

HAMPTON, David R; CASTRO, André Olímpio Mosselman Du Chenoy; KWSNICKA, Eunice Lakava. **Administração**: comportamento organizacional. São Paulo: McGraw-Hill, 1990.

HUBWORKS, **Hubworks**, © Hubworks [2021?]. Home Page .Disponível em: <https://hubworks.com/industries/hotel-logbook-app.html>. Acesso em: 21 de Out. de 2021.

LARMAN, Craig. **Utilizando UML e Padrões**: Uma Introdução à Análise e ao Projeto Orientados a Objetos e ao Desenvolvimento Iterativo . 3 ed . Bookman, 2007.

LÉO ANDRADE. **Crie aplicações que quiser, aprenda a programar se quiser**. ©2021 Leo Andrade. Home Page. Disponível em: <https://leoandrade.net/>. Acesso em: 13 de Out. de 2022

LOGBOOK. **Logbook**, Say “Good Bye” to papers logs!. ©2021. Página Inicial. Disponível em: <https://trylogbook.com/>. Acesso em: 21 de Out. de 2021.

MENDIX TECHNOLOGY BV 2021. **THE MENDIX LOW CODE PLATAFORM**. 2021. Disponível em: <https://www.mendix.com/platform/>. Acesso em: 17 Setembro 2021.

MENEZES, Paula Dutra Leão de; LOPES, Danila Kássia Lima. Políticas de gestão de pessoas aplicadas no setor de recepção de redes hoteleiras na cidade de João Pessoa - PB. **Cenário: Revista Interdisciplinar Em Turismo E Território**. João Pessoa-PB. Vol. 8, nº. 15, p. 41-63, 2020.

OUTSYSTEMS. **The Modern Application Platform for the Software That Makes the Difference**. Home Page. Disponível em: <https://www.outsystems.com/platform/>. Acesso em: 17 Setembro 2021.

PAGOTTO, Tiago; FABRI, José Augusto Fabri; L´ERARIO, Alexandre; GONÇALVES, José Antonio. **Scrum solo: Software process for individual development**. *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)* , 2016, pp. 1-6. DOI: 10.1109/ CISTI.2016.7521555. Disponível em: <https://ieeexplore.ieee.org/document/7521555>. Acesso em: 26 Set de 2022.

PEREIRA, Ricardo. **Rapid Application Development with OutSystems Create applications with OutSystems**: create applications with OutSystems up to seven times faster than with traditional technologies. 1. ed. Birmingham - Mumbai, 2022.

PRESSMAN, Roger S. **Engenharia de Software**: uma abordagem profissional. 7. ed. AMGH, 2011.

JAIN, Raj. **The Art of Computer Systems Performance Analysis**: Techniques for Experimental Design, Measurement, Simulation, and Modeling. 1 ed. Wiley, 1991.

SANCHES, fabricio; ALTHMANN, Marcio F. **Desenvolvimento Web com Asp.net MVC**. 1º. ed. Casa do Código, 2013.

SANCHIS, Raquel; GARCEUA-PERALES, Ócicatriz; FRAILE Francisco e POLER, Raul. Low-Code as Enabler of Digital Transformation in Manufacturing

Industry. **MDPI**, DOI: <https://doi.org/10.3390/app10010012>. Disponível em: <https://www.mdpi.com/2076-3417/10/1/12/htm>. 2019. Acesso em: 22 Ago, 2022.

SHIFTNOTE. **ShiftNote**, © 20007-2020. Pagina inicial. Disponível em: <https://www.shiftnote.com/free-trial-employee-scheduling-software-manager-log-book>. Acesso em: 21 de Out. de 2021.

SOMMERVILLE, Ian. **Engenharia de Software**. 9 ed. São Paulo. Pearson Prentice Hall, 2011.

SOUSA Antônio Fillippe de. **ESTRATÉGIAS DE COMUNICAÇÃO INTERNA NA HOTELARIA: O CASO DO OSCAR HOTEL**. Trabalho TCC publicado em repositório do Instituto Federal de Santa Catarina. Disponível em: <https://repositorio.ifsc.edu.br/bitstream/handle/123456789/872/Antonio%20Fillippe%20TCC%20hotelaria%202016-2.pdf?sequence=1&isAllowed=y>. Acesso em: 10 Set. 2021.

TISI, Massimo; MOTTU, Jean-Marie; KOLOVOS, Dimitrios S.; LARA, Juan de; GUERRA, Esther M.; DI RUSCIO, Davide; PIERANTONIO, Alfonso; WIMMER Manuel. **Lowcomote: Training the Next Generation of Experts in Scalable Low-Code Engineering Platforms**. Disponível em: <https://hal.archives-ouvertes.fr/hal-02363416/document>. 2019. Acesso em: 22 Ago 2022.

THAKUR, Shiwani.; DR. KULSHRESTHA, Ruchika.; SINGH, Vikas. **TOURISM AND HOSPITALITY Perspectives, Patterns and Practices**. Disponível em: [https://www.researchgate.net/publication/353547115\\_Issues\\_and\\_Challenges\\_of\\_Hospitality\\_Education\\_System\\_in\\_India](https://www.researchgate.net/publication/353547115_Issues_and_Challenges_of_Hospitality_Education_System_in_India). Acessado em: 6 Nov. 2021.

TRELLO. **O trello ajuda os times a avançar no trabalho**. Copyright © 2022 Atlassian. Disponível em: <https://trello.com/pt-BR>.

TRIGO, Antonio; HENRIQUES, Jorge. **Aprenda a programar com C# console, windows forms e SQL Server Microsoft ASP.NET Core | Azure | Xamarin**. 2°. ed. Zamboni Comercio De Livro Ltda, 2020.