



Software de gerenciamento de investimento em recursos para desenvolvimento de aplicações no Banco Mercantil do Brasil*

Resource investment management software for application development at Banco Mercantil do Brasil

André Luiz Mendes Corrêa¹
João Caram Santos de Oliveira²

Resumo

O Banco Mercantil, assim como os bancos mais modernos, utiliza muitos recursos tecnológicos para o funcionamento da instituição. O banco possui dificuldade na demonstração de investimentos em licenças de *software* para apoiar o processo de desenvolvimento para auditoria. Este problema está relacionado com os gastos indevidos que prejudicam os investimentos corretos dos recursos do banco. O objetivo deste estudo é desenvolver uma aplicação que gerencia a alocação de recursos de *software* no desenvolvimento de sistemas com o propósito de demonstrar a apropriação de custos de licenciamento para fins de auditoria e estudar uma solução para o problema. Para isso, a primeira etapa é a elucidação dos principais conceitos relevantes ao estudo, seguido da listagem das etapas de desenvolvimento presentes no banco assim como os recursos disponíveis, servindo para especificação e desenvolvimento da aplicação para solucionar o devido problema. Foram feitos testes de validação na aplicação, finalizando o estudo com a análise e conclusão. Ao todo, foram utilizados treze estudos que possibilitaram um maior entendimento dos conceitos a serem utilizados e as situações que já foram enfrentadas por outros atores. O resultado é um sistema capaz de gerar um documento padronizado com as informações que ajudam a área de auditoria interna do banco a reconhecer utilização de licenças de *software* pelos desenvolvedores.

Palavras-chave: Documentação. Software bancário. Auditoria. Listagem de recursos.

*Submetido em 06/12/2022 - Aceito em 07/12/2022

¹Bacharelado em Sistemas da Informação da PUC Minas, Brasil– acorrea@sga.pucminas.br

²Professor orientador– caram@pucminas.br

Abstract

The Banco Mercantil do Brasil, like most modern banks, uses a lot of technological resources for its functioning. The bank has difficulty demonstrating investments in software licenses to support the audit development process. This problem is related to the undue expenses that harm the correct investment of its resources. The aim of this study is to develop an application that manages the allocation of software resources in systems development to demonstrate the appropriation of licensing costs for auditing purposes and study a solution to the problem. For this, the first step is the elucidation of the main concepts relevant to the study. Next, listing the development stages present in the database as well as the available resources, serving for the specification and development of the application to solve the aforementioned problem. Validation tests were performed on the application, finishing the study with the analysis and conclusion. Altogether, thirteen studies were used; they enabled a deeper understanding of the concepts to be used and the situations that had already been faced by other actors. The result is a system capable of generating a standardized document with information that helps the bank's internal audit area to recognize the use of software licenses by developers.

Keywords: Documentation. Banking software. Audit. Resource listing.

1 INTRODUÇÃO

O Banco Mercantil do Brasil, assim como os demais bancos no país, depende de muitos recursos tecnológicos para seu funcionamento. Cada sistema é responsável por uma lógica de programação e regras de negócio, contendo diversos componentes que foram desenvolvidos ao longo do tempo, sendo que foram necessários muitos recursos para serem implementados, seja da área de TI ou de outras áreas de negócio.

Esses componentes, muitas vezes, não possuem documentação adequada. Um tipo de documentação muito importante, e que geralmente é negligenciada, é a listagem de todos os recursos que foram utilizados ao longo do desenvolvimento, pois esse levantamento é fundamental para auxiliar nas auditorias internas que o banco sofre.

A auditoria é um procedimento no qual ocorre uma investigação técnica por meio da coleta de dados para opinar em fins específicos, seja comprovação de fatos ou melhoria de processos (MONTOTO, 2018). No caso do Banco Mercantil do Brasil, a auditoria interna, feita pela área responsável do banco, tem o propósito de avaliar os processos internos. Uma parte que compõe esse processo é entender a utilização dos recursos gastos em cada área. Este procedimento é extremamente importante para identificar problemas em processos e sugerir melhorias e remodelagens, corrigir custos e contribuir com a melhoria contínua do banco.

A auditoria possui dificuldades para reconhecer os gastos desnecessários com licenças de *softwares* nos processos de desenvolvimento de componentes dentro do banco, levando em consideração que os desenvolvedores não apontam os recursos que foram utilizados para compor essa informação para a área. A partir do conhecimento dos recursos que são realmente necessários para o processo, são permitidos cortes com investimentos desnecessários e que podem ser aplicados em outros momentos.

Com isso, a falta de documentação de todos os recursos utilizados para implantação de componentes é o problema central, pois afeta uma etapa importante do processo de auditoria interna, que por sua vez tem o seu próprio valor. O principal motivo é a falta de tempo disponível no escopo do projeto, prejudicando aqueles que precisam da documentação, seja outros desenvolvedores ou os auditores.

O objetivo deste estudo é desenvolver uma aplicação que gerencia a alocação de recursos de *software* no desenvolvimento de sistemas a fim de demonstrar apropriação de custos de licenciamento para fins de auditoria. Ademais, facilitará o processo de documentação de recursos por um desenvolvedor de sistemas, seja um programador ou analista do Banco Mercantil do Brasil.

Para atingir esse objetivo, o primeiro objetivo específico é listar as etapas de implantação de componentes do banco, com o intuito de descobrir como a aplicação pode se encaixar no cotidiano dos programadores. O segundo objetivo específico é a construção da aplicação e capacidade dela de adicionar, gerenciar e listar as tarefas que o usuário esteja atuando e recursos que são utilizados por ele. O terceiro e último objetivo específico é a implementação da funcionalidade de gerar um relatório contendo as informações de uso de licenças como apoio

ao processo de auditoria interna.

Este trabalho está organizado da seguinte forma: na Seção 2 está o referencial teórico usado neste trabalho, a Seção 3 detalha os trabalhos relacionados, a 4 descreve os procedimentos metodológicos, a 5 demonstra os resultados obtidos e por fim a Seção 6 contendo as considerações finais.

2 REFERENCIAL TEÓRICO

Para uma maior compreensão do tema deste estudo, buscou-se na literatura os principais conceitos relacionados a: *softwares* bancários, auditoria e tecnologia, documentação de *software*, ciclo de desenvolvimento e método cascata.

2.1 Softwares bancários

Assim como descrito por Haralayya (2021), na era pré internet as instituições bancárias demoravam mais de dois dias para realizarem uma simples transferência. Com a evolução da tecnologia, os bancos passaram a assimilar inovações tecnológicas mais rápido e frequentemente para estarem sempre similar à concorrência e dispostos a entregar um serviço que não sofra com interrupções. E com isso se tornaram berços de grandes conhecimentos sobre *softwares* bancários e como administrá-los para o bom funcionamento da instituição. De acordo com a Pesquisa Febraban de Tecnologia Bancária, a indústria bancária é a que mais investe em tecnologia, sendo a expectativa de orçamento para 2022 de R\$35,5 bilhões (FEBRABAN, 2022).

As instituições bancárias modernas são grandes dependentes de tecnologia (MALANOVICZ, 2018), pois automatizam cada vez mais os seus processos para que executem sem erros. *Softwares* bancários são os processos que compõem os sistemas de informação utilizados dentro das agências, nos principais produtos do banco (MALANOVICZ, 2018). Geralmente esses são dotados de cálculos complexos e que envolvem total discricção dos dados que são submetidos a cada tarefa.

2.2 Auditoria e tecnologia

Segundo Costa (2019), a auditoria é o processo sistêmico no qual são obtidas provas como dados para a comprovação de informações. O responsável pela coleta, verificação, construção e distribuição da análise destes dados é o auditor, podendo ele ser interno, da própria empresa, ou externo, como um agente do governo. O auditor interno é um dos responsáveis pelo sucesso da organização por meio da distribuição das informações geradas por seu trabalho. Contudo, ele deve estar a parte de todas as informações do negócio. Ainda para o autor, as tec-

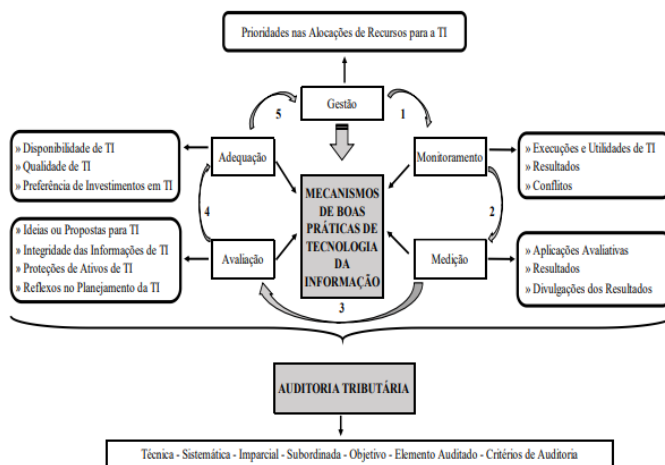
nologias são ferramentas indispensáveis para auxiliar a realização dos trabalhos relacionados às auditorias, sendo denominadas por Pedrosa (2017), como *Computer Assisted Audit Tool and Techniques*, as CAATTs.

Para Costa (2019) à luz de Pedrosa e Costa (2012), os CAATTs são definidos por: "o uso de determinados *softwares* que podem ser usados pelo auditor para realizar auditorias e para atingir os objetivos de auditoria"

Dantas (2019), sintetizou as boas práticas de TI na auditoria fiscal em um modelo representado na Figura 1, esse modelo é aplicado para as análises de dados coletados e seus resultados, conforme os seguintes critérios:

- a) Gestão de TI: investiga-se que a TI tenha recursos para atender as necessidades das auditorias;
- b) Monitoramento da TI: investiga-se as execuções da TI e os resultados gerados nas auditorias;
- c) Medição da TI: investigam-se as utilizações das aplicações da TI nas auditorias;
- d) Avaliação: avaliação das ideias propostas por gerentes, além das informações das auditorias;
- e) Adequação: verificada a disponibilidade e qualidade da TI utilizada nas auditorias.

Figura 1 – Modelo Boas práticas de TI na auditoria



Fonte: (DANTAS, 2019)

2.3 Documentação de softwares

De acordo com Aghajani et al. (2019), a documentação é responsável por fornecer informações relevantes a respeito do *software*, como seu funcionamento e principais características

necessárias a quem precisa. Apesar de seus benefícios, tem se notado um mau uso da ferramenta, acarretando assim em uma documentação negligenciada e inadequada. Em muitos casos, a documentação precisa ser gerada automaticamente, utilizando apenas a entrada de alguns parâmetros do usuário, para gerar um relatório final ou documento útil.

Para Robillard et al. (2017) é por meio da documentação técnica que o conhecimento é transmitido para aquele que não participou no desenvolvimento do componente. Esses documentos são compostos com informações ou comentários explicativos sobre cenários, comparações, códigos de exemplo, construção e dados técnicos que são essenciais para a vitalidade do projeto.

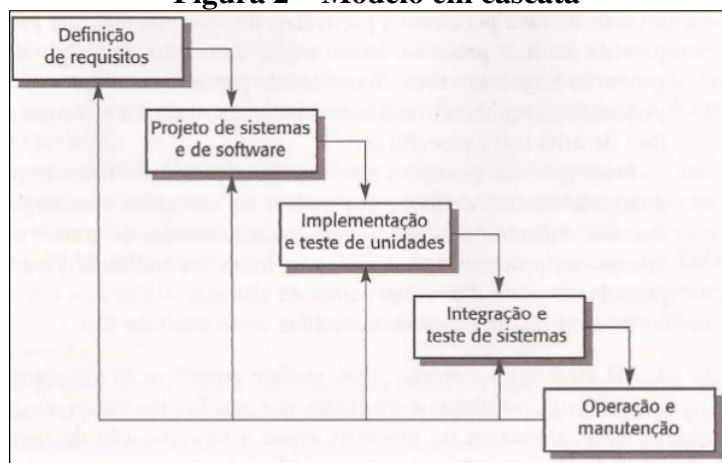
Aghajani et al. (2019) afirma que para a obtenção de uma ferramenta que de fato, entregue resultados satisfatórios são necessários conhecimentos mais aprofundados em *softwares* que auxiliem nesse objetivo.

2.4 Ciclo de desenvolvimento e método cascata

O ciclo de vida de um *software* indica as tarefas que compõem o desenvolvimento da aplicação, assim como os processos gerados. Um dos ciclos mais clássicos de desenvolvimento de *softwares* é a utilização de três etapas básicas, sendo elas: a definição, o desenvolvimento e a operação. Este projeto é implementado por meio do gerenciamento em cascata (SOMMERVILLE, 2003).

O método cascata tem como etapas fundamentais: análise e definição de requisitos, projeto, implementação, teste e integração como visto no modelo (Figura 2). Ele apresenta a facilidade de encaixar as tarefas enfileiradas de forma a se tornar um processo de desenvolvimento linear e contínuo. Assim, além de ter uma fácil compreensão, uma tarefa só é iniciada quando a anterior é finalizada, mantendo um maior controle nas conclusões de atividades (DEV MEDIA, 2011).

Figura 2 – Modelo em cascata



Fonte: (SOMMERVILLE, 2003)

3 TRABALHOS RELACIONADOS

Foram encontrados poucos artigos que trazem como objetivo o desenvolvimento de aplicações voltadas para documentação em instituições bancárias e menos ainda são aqueles que trazem uma ferramenta de documentação de recursos utilizados para desenvolvimentos de componentes. Isso se dá pelo fato de aplicações desenvolvidas internamente em bancos são bem específicas para atendimento das necessidades somente daquela empresa, além disso, a divulgação poderia colocar em risco a segurança e inteligência da instituição. Visando isso, essa seção apresenta conceitos e exposições que serviram para guiar e motivar o devido estudo.

O artigo de Malanovicz (2018), à luz de Tonolli Jr (2012), relata como tem sido difícil para instituições bancárias o desenvolvimento de produtos internos devido a alta demanda de serviços e prazos cada vez mais curtos, dificultando a inovação em muitos dos processos. Em seu estudo, são analisadas alternativas de remodelagem para os processos de desenvolvimento em uma instituição bancária. Por meio de entrevistas com a equipe de desenvolvedores, análise de documentos e observação nos projetos, foi capaz de descrever o processo atual, propor uma remodelagem e destacar os benefícios que a empresa poderia usufruir.

Já em relação ao gerenciamento de projetos de *softwares* bancários, Pinheiro (2017) aborda a importância dessa prática para as empresas, através de uma análise comparativa entre a instituição bancária e o guia de boas práticas de gerenciamento de projetos. Seu estudo realizou uma revisão bibliográfica das boas práticas apresentadas pelo Guia PMBOK em relação a gerenciamento de projetos e áreas de conhecimento. Por fim, ele conclui apresentando uma comparação entre os processos descritos no Guia PMBOK e as atividades que constam no documento da instituição que sofre com entradas incompletas e saídas não explícitas, o que dificulta a clareza de transmissão do conhecimento dos processos.

Relacionados a documentação de processos, Buchgeher et al. (2018) trazem o estudo de caso de uma fábrica de manufatura que sofria com problemas de documentar seus componentes. A proposta do trabalho desses autores era observar, avaliar e entregar um novo processo de documentar código na empresa, por meio de um *software* personalizado feito por eles. Mesmo trazendo métodos de documentação atualizada em tempo real, processos da fábrica ainda apresentavam problemas, demonstrando como é falho documentar com o passar do tempo, pois é preciso muitos recursos humanos para criar e manter a documentação e a falta de ferramentas de documentação de *software* dedicadas que apoie os desenvolvedores nas atividades de documentação.

Os trabalhos apresentados nesta seção foram de importância para elucidar as principais dificuldades que os autores enfrentaram, assim como trazer conceitos relevantes. Com isso, contribuíram para demonstrar como podem ser fundamentais estudos como este, que se aprofundem em resolver um problema pertinente em relação a documentação e a falta de estudos mais acadêmicos em instituições bancárias, podendo ser de grande ajuda futuramente para outros desenvolvedores e empresas.

4 METODOLOGIA

Para elaboração do projeto, a metodologia adotada se baseia em cinco etapas apontadas no diagrama (Figura 3) e explicadas posteriormente.

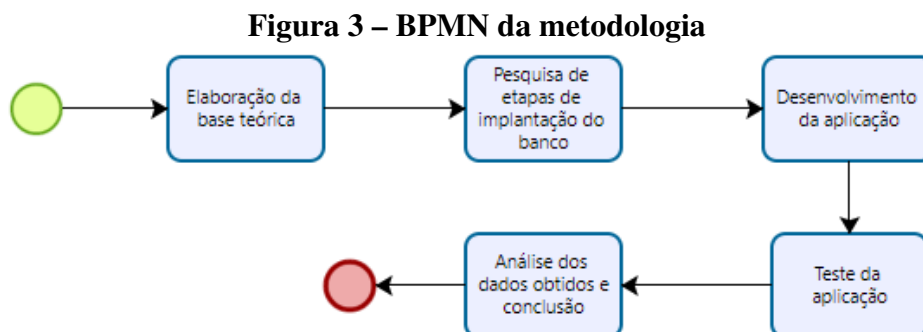


Figura 3. Desenvolvido pelo autor

4.1 Elaborar a base teórica

Foi feita a elaboração de uma base teórica como meio de produção de conhecimento e contextualização dos assuntos abordados, como desenvolvimento de *softwares* e documentação, produção de *softwares* bancários e a tecnologia no processo de auditoria bancária.

Para compreender melhor o tema e elucidar os conceitos, a principal técnica utilizada foi a revisão bibliográfica. Por meio do acesso ao conteúdo do Portal de Periódicos da CAPES (CAPES, 2021), foram levantados os estudos que embasam o presente artigo.

4.2 Levantar informações necessárias para a área de auditoria

Para iniciar a especificação dos requisitos funcionais, foi necessário antes entender quais dados a aplicação precisava entregar para que gerasse valor e pudesse contribuir com a área de auditoria para geração de informações.

Por meio de uma entrevista realizada com um analista sênior de segurança da informação, o qual participa diretamente no processo de auditoria interna do banco, os dados gerados pela aplicação poderiam contribuir para levantamento de utilização de licenças com *softwares* para certos sistemas do banco.

A entrevista foi realizada por meio do *chat* do *Google Workspace*, com duração aproximada de trinta minutos, sendo pontuadas pelo analista questões relevantes como a possibilidade da aplicação gerar um relatório que pudesse evidenciar as horas de uso de recursos (licença de ferramentas digitais, de recursos terceiros e humanos) e também filtrar por sistema e intervalo de tempo (levando em consideração o Banco Mercantil, o tempo definido seria de quatro meses visto que, é a duração das auditorias internas realizadas por ele).

4.3 Listar as etapas de implantação de componentes do banco

A listagem das principais etapas de implantação de componentes dentro dos ambientes do banco, foi feita por meio da experiência adquirida pelo autor e serve para ampliar o conhecimento necessário para o desenvolvimento das funcionalidades que estarão presentes no *software* proposto neste documento. O levantamento dos recursos foi realizado com base nas ferramentas disponíveis nas máquinas dos desenvolvedores do banco e na lista de *softwares* que podem ser requisitados via demanda para instalação. No Quadro 1 é demonstrado o resultado obtido:

Quadro 1 - Etapas de implantação de componentes do Banco Mercantil do Brasil

Etapa	Artefatos gerados	Recursos geralmente utilizados
Análise	Documentos de requisitos e engenharia de <i>software</i>	Pacote <i>Office</i> , <i>Meet</i> , tempo, funcionários de outras áreas
Modelagem	Documento de escopo do projeto	<i>Softwares</i> de Modelagem de processos (ex: <i>Bizagi</i>), tempo
Desenvolvimento	Componente e resultado dos primeiros testes em ambiente interno	Programadores alocados, IDEs de programação, <i>softwares</i> de testes, tempo
Homologação	Resultado de testes do componente com o usuário e alterações	<i>Meet</i> , tempo, funcionários de outras áreas
Implantação	Documentos de implantação	Tempo, ferramenta de controle de códigos
Acompanhamento de implantação	Relatório contendo as principais informações do atendimento a demanda	Tempo, <i>Meet</i> , IDEs de programação

Fonte: Elaborado pelo autor

Por meio deste levantamento e entendimento das principais etapas, além da pequena entrevista que proporcionou levantar as entradas e saídas esperadas, foi elaborado um documento contendo os requisitos funcionais da aplicação de listagem de recursos de *software*. O documento foi gerado no formato .pdf, utilizando um modelo de listagem de requisitos e anexado ao documento principal.

4.4 Desenvolver a aplicação

A principal etapa é composta pelo desenvolvimento da aplicação de listagem de recursos. O objetivo desta etapa é desenvolver uma aplicação *desktop*, um executável instalável, que contém as funcionalidades listadas na etapa anterior no documento de requisitos funcionais e não funcionais.

Para gerenciar o ciclo de desenvolvimento deste projeto foi escolhido o método de gerenciamento em cascata (SOMMERVILLE, 2003). A escolha se dá pela facilidade de implantar tarefas consecutivas em uma linha direta, sem a necessidade de maior complexidade na entrega das atividades.

4.5 Testar e analisar aplicação

Após o desenvolvimento da aplicação proposta neste documento, foram realizados os devidos testes para a validação de seu uso, sendo os principais aspectos analisados com os testes: eficácia; qualidade; tempo utilizado até a geração do relatório; acessibilidade e busca por falhas.

Devido a um prazo menor e a dificuldade de alocar funcionários do banco para trabalhos extras (como a etapa de testes) foi realizada somente a viabilização local, sem a participação do usuário final, se tornando parte de uma complementação futura deste trabalho.

Para isso, são utilizados os testes funcionais, também chamados de testes de caixa preta, uma metodologia que visa analisar as saídas levando em consideração as entradas da aplicação, sem atentar ao código fonte. A técnica de testes unitários das estruturas foi utilizada com o intuito de verificar as respostas de cada função implementada de forma isolada e em código nas suas respectivas classes, para garantir uma maior precisão das respostas que são obtidas.

Os testes unitários foram realizados por meio da utilização do *XUnit.net*, um pacote que permitiu testar os métodos de preenchimento de listas e resultados internos, principalmente aqueles que continham uma saída já esperada. Por exemplo, tendo uma entrada um número de demanda de serviço, o método de listar demanda por número deveria retornar apenas aquele registro específico.

Em relação aos testes de validação das telas foram feitas pelo próprio autor e levou em consideração testes de inserção de dados incorretos em diversos campos e utilizando os recursos exaustivamente (teste dos botões, telas, campos e funções de tela presentes no *Windows Forms*) com o intuito de encontrar possíveis falhas e principalmente se atendia corretamente os requisitos levantados anteriormente.

Após a realização dos testes foram validados a aplicação e os requisitos levantados, apontando os resultados obtidos e se a aplicação atende o principal objetivo deste estudo. A conclusão contém os principais pontos relevantes do estudo e se ele pode contribuir com o banco por meio de estudos futuros, com a finalidade de corrigir problemas levantados, sugerir

melhorias e a realização de testes adequados com o usuário final.

4.6 Ferramentas utilizadas

A organização, tanto do desenvolvimento da aplicação do projeto, quanto a do próprio texto, utilizou o método *Kanban*, por meio da ferramenta *Trello*. O *Kanban* é um método visual para gerenciar e conduzir o trabalho (MESH, 2020). Por meio de um quadro, cada etapa é documentada com os passos para conclusão da tarefa por meio de *cards* detalhando as funcionalidades a serem desenvolvidas através da visão do usuário final.

Para desenvolver a aplicação, é seguido um padrão do Banco Mercantil do Brasil em que as ferramentas internas, que visam serem utilizadas pelos usuários das áreas de tecnologia, são construídas utilizando a linguagem de programação C# para o *back end*, com o uso da IDE *Visual Studio* que contém os recursos e bibliotecas necessários para desenvolver aplicações *desktop Windows*.

Para o *front end* da aplicação, são utilizados os recursos do *Visual Studio* para compor as partes visuais da aplicação. Esses recursos utilizarão linguagem de marcação HTML, de estilização CSS que são suportados pela IDE e fazem parte de tecnologias *web* mais recentes.

Para os testes unitários dos métodos que compõem a aplicação, foi utilizado a ferramenta *XUnit.net*, um pacote com códigos que auxiliam os programadores a criarem testes nos quais são validados as saídas ou resultados de cada método, de acordo com as entradas passadas por parâmetro.

5 RESULTADOS OBTIDOS

Esse estudo teve como objetivo desenvolver uma aplicação que atendesse os requisitos levantados e pudesse apresentar uma possibilidade de melhoria para o processo já existente de documentação de recursos no Banco Mercantil do Brasil. Nesta seção são apresentados os artefatos gerados ao longo deste estudo.

5.1 Especificação de requisitos

5.1.1 Descrição geral do sistema

O sistema tem como objetivo melhorar o fluxo de, armazenando os dados manipulados em um documento local para acesso apenas na máquina. O sistema será instalado na máquina do usuário para utilização específica de cada usuário.

O sistema será desenvolvido em linguagem C#, e utilizará *Windows Forms* para o *front-*

end.

5.1.2 Descrição dos envolvidos

Programador / analista: os principais usuários serão os programadores / analistas do banco que irão poder adicionar, editar e excluir as tarefas e recursos que serão utilizados por eles durante a execução de seus trabalhos.

Auditor interno: o sistema será responsável por gerar relatórios a pedido dos auditores.

5.1.3 Descrição dos usuários

Nome	Descrição	Responsabilidade	Envolvido
Programador / analista	- Representa a pessoa que trabalha no desenvolvimento de componentes no banco	- Registrar as tarefas e recursos utilizados na execução de seus trabalhos - Gerar relatório de acordo com o pedido dos auditores internos	Este usuário corresponde a qualquer um que trabalhe no banco e utilize recursos de T.I. para executar suas atividades

5.1.4 Requisitos funcionais

5.1.4.1 [RF001] Manter tarefas

Descrição: O sistema permitirá o cadastro, alteração, inativação, consulta e pesquisa de tarefas em um documento local contendo os dados informatizado, seguro e organizado.

Atores: Programador / Analista.

Prioridade: Alta.

Entradas:

- SAMB: número da demanda referente a tarefa. [*int*]
- Nome: nome da tarefa em questão. [*string*]
- Sistema: sistema a qual pertence aquela tarefa. [*string(3)*]
- Data criação: a data em que foi criada a tarefa. [*datetime*]

Saídas:

- Exibir na tela os registros cadastrados.

5.1.4.2 [RF002] Manter recursos

Descrição: O sistema permitirá o cadastro, alteração, inativação, consulta e pesquisa de recursos em um documento local contendo os dados informatizado, seguro e organizado.

Atores: Programador / Analista.

Prioridade: Alta.

Entradas:

- Etapa: etapa em que foi utilizado o recurso. [*string*]
- Recurso: nome do recurso utilizado. [*string*]
- Tempo de uso: tempo total que foi utilizado o recurso. [*time(HH:mm)*]
- Outra área: se o recurso pertence a outra área. [*bool*]
- Quem: caso o recurso pertença a outra área, qual é essa área. [*string*]

Saídas:

- Exibir na tela os registros cadastrados.

5.1.4.3 [RF003] Gerar relatório

Descrição: O sistema permitirá a criação de um documento relatório contendo as informações de tarefas e recursos de acordo com o intervalo e sistema exigido.

Atores: Programador / Analista.

Prioridade: Alta.

Entradas:

- Data início: uma data para indicar o início do intervalo para buscar os registros. [*datetime(yyyy-MM-dd)*]
- Sistema: o sistema desejado para filtro dos registros retornados. [*string(3)*]

Saídas: - Um documento no formato .pdf contendo as informações de entrada e os registros de tarefas e recursos de acordo com o exigido.

5.1.5 Requisitos não funcionais

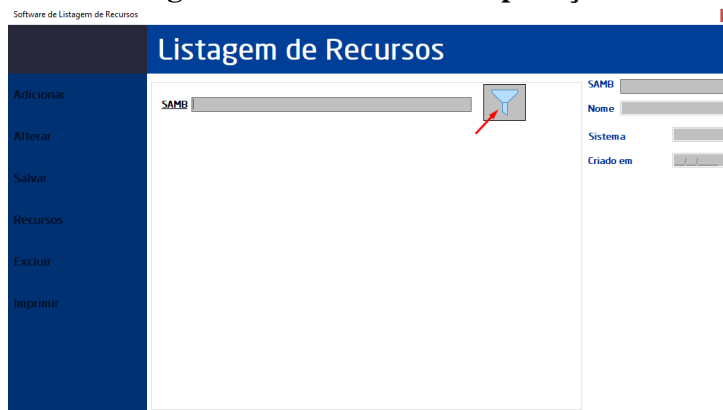
Número de ordem	Requisito	Descrição	Prioridade
RNF001	Implementação em C#	O sistema deve ser implementado na linguagem C#	Alta
RNF002	Tempo de validação	O sistema deve ser capaz de validar entradas em até 20s	Baixa
RNF003	Implementação em modelo MVC	O sistema será implementado através da <i>framework .NET Core</i>	Média
RNF004	Implementação de testes pelo <i>XUnit.net</i>	O sistema irá ser testado com a implementação do <i>XUnit.net</i> no desenvolvimento	Alta

5.2 Tela inicial e de listagem de tarefas

Os desenvolvedores e analistas do banco são os principais usuários da aplicação, sendo somente o relatório que seria de auxílio à área de auditoria. Portanto, a primeira funcionalidade deve ser permitir ao usuário registrar suas tarefas, que logo após, irão receber os recursos cadastrados por ele.

Ao iniciar a aplicação, o desenvolvedor irá se deparar com uma tela ainda sem dados preenchidos (Figura 4), posteriormente ele deverá apertar o botão de filtragem de resultados para que possa retornar todas as tarefas que o usuário já havia criado.

Figura 4 – Tela inicial da aplicação

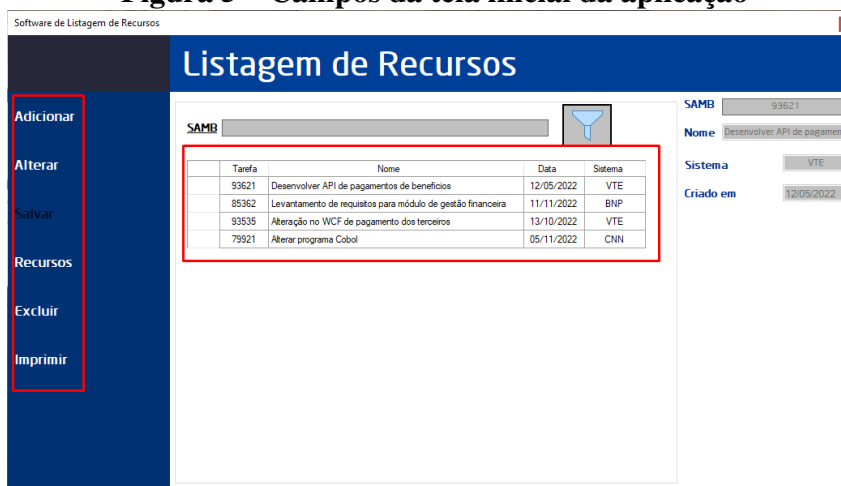


Fonte: Desenvolvido pelo autor

Após apertar o botão correspondente a filtragem, é carregado no meio da tela, uma tabela contendo as tarefas já criadas (Figura 5), assim como as respectivas informações de número da demanda, nome, data de criação e sistema responsável.

O menu lateral fica disponível com as opções de adicionar, alterar e excluir tarefas, assim como a possibilidade de exibir os recursos da tarefa escolhida e imprimir o relatório de uso de recursos.

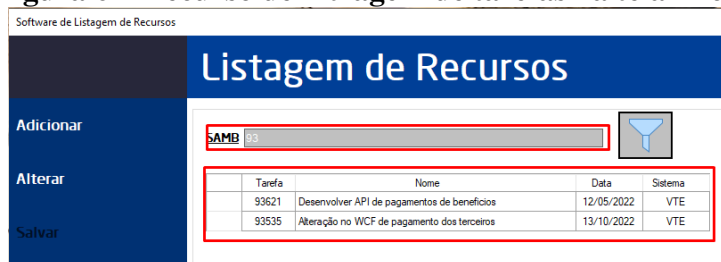
Figura 5 – Campos da tela inicial da aplicação



Fonte: Desenvolvido pelo autor

É possível filtrar as tarefas pelo número da demanda de serviço através da barra de filtragem (Figura 6), que deve ser preenchida com o número desejado. Logo após, o usuário deve apertar o botão de filtro e com isso é exibido o resultado logo abaixo com as tarefas pertencentes a pesquisa. Isso traz uma facilidade de busca caso a lista tenha diversos registros.

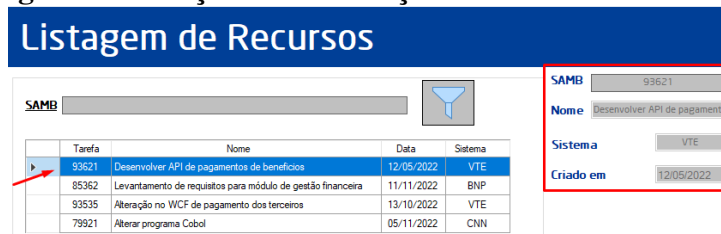
Figura 6 – Recurso de filtragem de tarefas na tela inicial



Fonte: Desenvolvido pelo autor

Ainda existe a possibilidade do usuário verificar as informações da tarefa na barra lateral direita assim que alguma é selecionada (Figura 7).

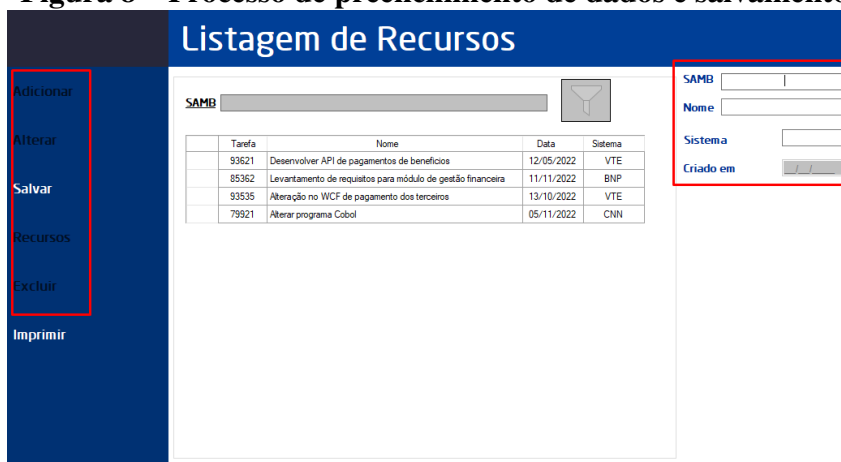
Figura 7 – Seleção de informação no menu lateral direito



Fonte: Desenvolvido pelo autor

Selecionando a opção de "Adicionar", no menu lateral esquerdo, os demais botões do menu são desabilitados, ficando habilitado somente o de salvar. Na lateral direita, irá habilitar os campos para preenchimento de uma nova tarefa, sendo possível vê-la na lista, logo após salvar (Figura 8).

Figura 8 – Processo de preenchimento de dados e salvamento



Fonte: Desenvolvido pelo autor

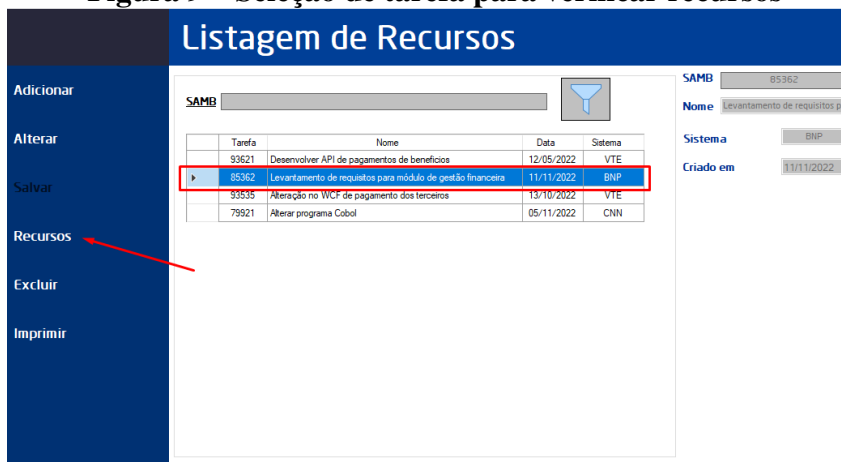
Com a implementação dessa funcionalidade de gerenciamento de tarefas, parte do segundo objetivo específico é atendido, com a possibilidade do usuário poder criar, editar, excluir e listar as tarefas nas quais ele esteja atuando. Além de permitir que seja consultado as tarefas que ele já tenha atuado, servindo de exemplos passados.

5.3 Tela de listagem de recursos

A segunda etapa irá contribuir para a conclusão do segundo objetivo específico e auxiliar no objetivo principal. Para isso, primeiro o usuário deve ser capaz de inserir nas tarefas uma lista de recursos que representam as ferramentas utilizadas por ele ao longo de seu trabalho naquela mesma tarefa.

Na tela inicial, o usuário deve selecionar uma tarefa que deseja verificar os principais recursos utilizados e escolher a opção de "Recursos" no menu lateral esquerdo (Figura 9).

Figura 9 – Seleção de tarefa para verificar recursos

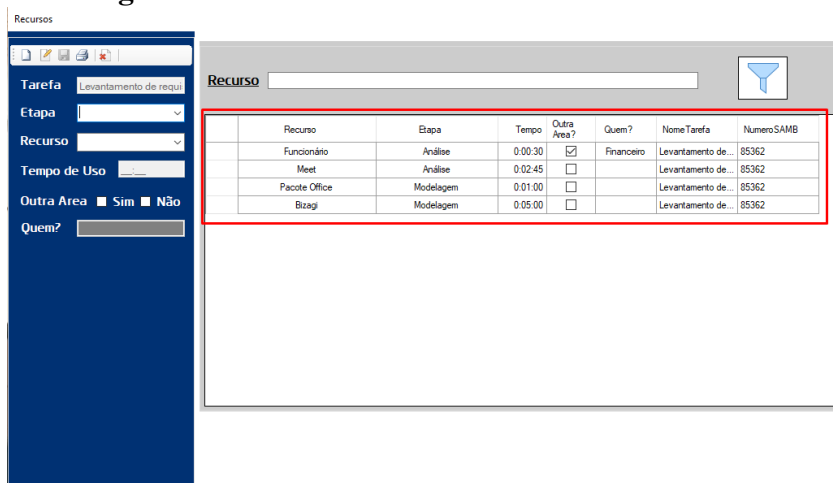


Fonte: Desenvolvido pelo autor

É aberta uma nova tela na qual é possível ver uma tabela contendo os recursos que já foram adicionados à tarefa (Figura 10) e suas informações como nome do recurso, etapa que foi

utilizada, tempo de utilização, se é de uma área diferente e se sim, qual área.

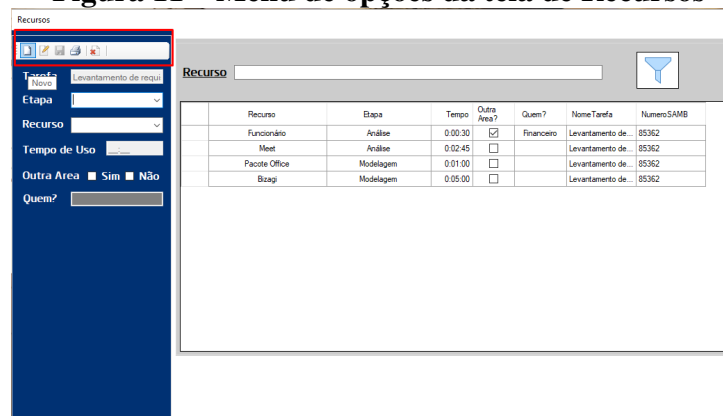
Figura 10 – Lista de recursos da tarefa selecionada



Fonte: Desenvolvido pelo autor

Nesta tela, o menu fica disponível na região superior do lado esquerdo e possui as opções de "adicionar, alterar e excluir um recurso"(Figura 11), sendo essas duas últimas somente quando estiver selecionado um já existente na lista.

Figura 11 – Menu de opções da tela de Recursos



Fonte: Desenvolvido pelo autor

Ao selecionar a opção de adicionar um novo recurso, os campos na lateral esquerda da tela ficam disponíveis para o usuário preencher (Figura 12). A lista de "Recurso" e "Etapa" são preenchidos de acordo com a disponibilidade no banco de dados que a aplicação consulta. Isso garante que somente os recursos permitidos e homologados pelo banco possam ser utilizados. Para adicionar um novo recurso, o usuário deverá abrir uma demanda pedindo para inserir na base de dados.

As listas que compõem os campos de "Recurso" e "Etapa" complementam a conclusão do primeiro objetivo específico de listar as etapas e recursos disponibilizados pelo banco. Já a funcionalidade de adicionar e gerenciar recursos às tarefas, completa o segundo objetivo específico deste artigo. Futuramente, é possível até mesmo detalhar melhor os itens que compõem essas listas com o intuito de trazer mais informações de uso e possibilidades.

Figura 12 – Salvar um novo recurso

Recurso	
Funcionário	
Meet	
Pacote Office	
Bizagi	

Fonte: Desenvolvido pelo autor

5.4 Relatório gerado

A última etapa e que atende ao objetivo principal deste projeto é a capacidade de gerar um documento relatando os recursos que foram cadastrados pelo usuário. Para a geração do relatório, primeiramente o usuário deve escolher a opção de “Imprimir” por meio do botão na tela inicial (Figura 13).

Figura 13 – Seleção da opção de Imprimir na tela inicial

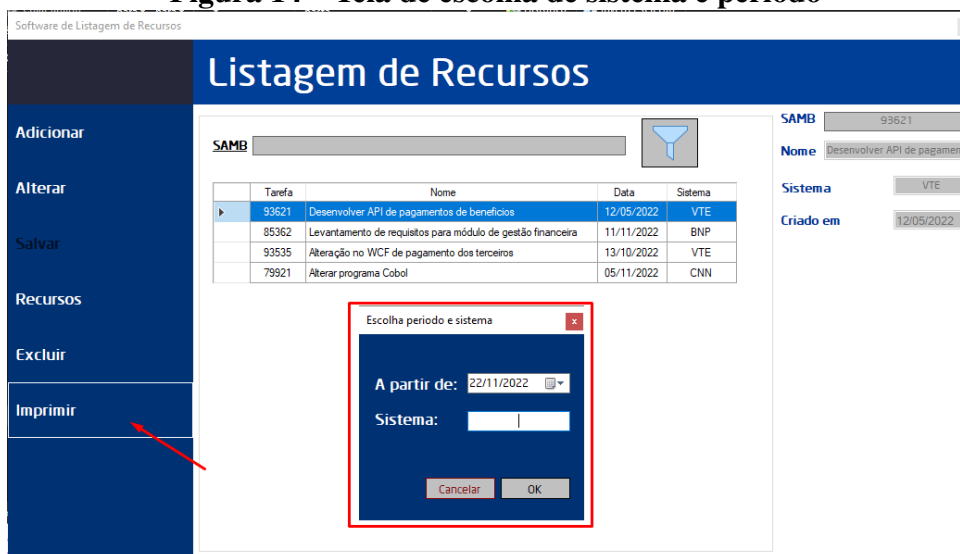
Tarefa	Nome	Data	Sistema
93621	Desenvolver API de pagamentos de beneficios	12/05/2022	VTE
85362	Levantamento de requisitos para módulo de gestão financeira	11/11/2022	BNP
93535	Alteração no WCF de pagamento dos terceiros	13/10/2022	VTE
79921	Alterar programa Cobol	05/11/2022	CNN

Fonte: Desenvolvido pelo autor

Após isso, é gerado um *pop-up* (Figura 14) no qual ele pode escolher qual o sistema responsável das tarefas e o intervalo de tempo que ele precisa.

O documento que é gerado pela aplicação apresenta na região superior uma tabela contendo o sistema e o intervalo de tempo escolhido pelo usuário para listagem das tarefas, seguidos pelo tempo total de uso dos recursos e a quantidade.

Figura 14 – Tela de escolha de sistema e período



Fonte: Desenvolvido pelo autor

Logo após essa primeira tabela, o documento detalha, separadamente por tarefas, os recursos utilizados (Figura 15), contendo as informações de nome do recurso, a etapa em que foi utilizada, o tempo de uso, se foram disponibilizados por outra área e por quem.

Durante a entrevista com o analista sênior de segurança da informação, foi levantado que o principal dado que deveriam constar no relatório gerado pela aplicação é o de horas totais de uso dos recursos. Além disso, o relatório atende também outros dois requisitos sendo eles o de filtragem por sistema e por data, preenchidos pelo usuário em tela e demonstrado no relatório para registro dos auditores. Com a evidência destes dados no relatório gerado, esperasse eles que ajudassem a trazer informações para a área de auditoria interna do banco, além de atender ao objetivo principal deste estudo.

6 CONCLUSÃO

Este artigo teve o intuito de estudar e entender o problema relacionado à documentação gerada pelos desenvolvedores, detalhando os recursos utilizados em cada etapa de implementação de um componente no banco Mercantil do Brasil. Assim, propondo uma solução que pudesse ajudar na otimização de tempo ao documentar os recursos utilizados por eles em seus trabalhos de codificação e análise e, ao mesmo tempo, gerar um documento padronizado que ajude o processo de auditoria interna do banco.

Para alcançar esse resultado foram utilizados estudos teóricos a fim de entender e aprofundar mais sobre os temas: documentação, auditoria e *softwares* bancários. Além disso, houve o levantamento das principais informações que contribuíram com a construção desta pesquisa e consequentemente para a concepção da aplicação de listagem de recursos.

Em virtude desse processo ser realizado de forma manual pelos desenvolvedores e analistas atualmente, a geração de dados nos documentos realizados por eles acaba sendo precária.

Figura 15 – Relatório gerado pela aplicação

Relatório de Recursos

Sistema: VTE
Período: 02/05/2022 a 22/11/2022
Tempo Total: 05:25:00

Tarefa SAMB Nº 93621

Recurso	Etapa	Tempo	Area	Quem
Meet	Desenvolvimento	02:00	Sim	Financeiro
Visual Studio	Homologação	02:25	Sim	Segurança

Tarefa SAMB Nº 93535

Recurso	Etapa	Tempo	Area	Quem
Visual Studio	Homologação	01:00	Não	

Fonte: Desenvolvido pelo autor

Tendo isso em vista, entende-se que a utilização da aplicação para listar as tarefas e recursos, pode trazer vantagens em relação a rapidez e maior exatidão no preenchimento das informações do relatório, além da obtenção de um documento padronizado.

Os relatórios gerados pela aplicação, contém os dados de entrada e saída levantados por meio da entrevista com um dos envolvidos pelos processos de auditoria interna no banco. Há ainda a possibilidade de entregar no relatório outros campos que auxiliem a área de auditoria a identificar melhor os recursos utilizados.

Dessa forma, entende-se que os objetivos do presente estudo foram alcançados e percebe-se ainda que a aplicação possui capacidade para receber novas funcionalidades e aprimoramentos futuros. Porém, é de se entender que os objetivos alcançados se relacionam à construção do sistema e não pelos benefícios da sua utilização pelos auditores, devido a ausência dos testes de usabilidade. Contudo, este estudo auxiliou de forma precisa as práticas de análise e desenvolvimento de sistemas, uma vez que possibilitou empregar os conhecimentos adquiridos ao longo do curso.

6.1 Trabalhos futuros

Os testes de usabilidade são essenciais para a validação do produto, pois, junto ao usuário final é possível observar e corrigir questões relacionadas à aplicação nos quesitos de navegação e interface. Neste estudo, devido a grande burocracia presente no banco para alocação de usuários em tarefas extra funcionais e a indisponibilidade de tempo no escopo do projeto para

os testes, esse procedimento ficará como parte de estudos futuros.

Outro ponto a ser desenvolvido é uma interface mais amigável nas telas de navegação, podendo conter *design flex* e responsivo, imagens e ícones mais modernos e utilizar as logomarcas do banco Mercantil do Brasil. Além de tornar o próprio relatório gerado pela aplicação mais agradável e melhor formatado.

Referências

AGHAJANI, Emad et al. Software documentation issues unveiled. In: **SOFTWARE DOCUMENTATION ISSUES UNVEILED**, 2012, Montreal, QC. **2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)**. Canada: IEEE, 2019.

BUCHGEHER, Georg et al. Providing technical software documentation as a service - an industrial experience report. In: **2018 25th Asia-Pacific Software Engineering Conference (APSEC)**. [S.l.: s.n.], 2018. p. 581–590.

CAPES. **Portal de Periódicos da CAPES**. 2021. Disponível em: <<https://www-periodicos-capes-gov-br.ez1.periodicos.capes.gov.br/index.php?>>

COSTA, Erick Rodrigues da. **As ferramentas de tecnologia de informação em auditoria para detectar fraude fiscal em Angola**. 2019. Dissertação (Mestrado) — IPC - Instituto Politécnico de Coimbra, Portugal.

DANTAS, Cantidiano Novais. **Mecanismos de boas práticas de tecnologia da informação nas auditorias tributárias das Fazendas Públicas estaduais após a inserção do sistema público de escrituração digital**. 2019. Dissertação (Mestrado) — Universidade Federal de Sergipe, São Cristóvão, SE.

DEVMEDIA. **Ciclos de Vida do Software**. 2011. Disponível em: <<https://www.devmedia.com.br/ciclos-de-vida-do-software/21099>>.

FEBRABAN. **Pesquisa FEBRABAN de Tecnologia Bancária 2022**. 2022. Disponível em: <<https://cmsarquivos.febraban.org.br/Arquivos/documentos/PDF/pesquisa-febraban-2022-vol-2.pdf>>.

HARALAYYA, Dr. Core banking technology and its top 6 implementation challenges. v. 4, p. 25–27, 01 2021.

MALANOVICZ, Aline Vieira. Análise de alternativas para o desenvolvimento de produtos em sistemas de informações para empresa do setor bancário. **Revista Electronica de Sistemas de Informação**, Campo Largo, v. 17, n. 2, mai./aug. 2018.

MESH, Janet. **Método Kanban: Guia detalhado e 5 modelos prontos para usar**. 2020. Disponível em: <[MONTOTO, E. Contabilidade geral e avançada esquematizado. São Paulo: Saraiva, v. 5, p. 25–27, 01 2018.](https://blog.trello.com/br/metodo-kanban#:~:text=O%20Kanban%20%C3%A9%20um%20m%C3%A9todo,a%20efici%C3%Aancia%20do%20fluxo%20atual.>></p></div><div data-bbox=)

PINHEIRO, Felipe Gonsalves. Gerenciamento de projetos de desenvolvimento de software em uma instituição bancária. v. 1, p. 03–22, 08 2017. Disponível em: <<https://bdm.unb.br/handle/10483/17602>>.

ROBILLARD, Martin P. et al. On-demand developer documentation. In: **2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)**. [S.l.: s.n.], 2017. p. 479–483.

SOMMERVILLE, Ian. Engenharia de software. **Revista Electronica de Sistemas de Informação**, São Paulo, 2003.