

Desenvolvimento de um Software Web para Controle de Itens Pertencentes a Uma Instituição de Ensino

Eduardo Rossato Alessio¹, Gustavo Stangherlin Cantarelli¹

¹Curso de Sistemas de Informação – Universidade Franciscana
CEP 97010-032 – Santa Maria – RS – Brasil

alessioeduardo98@gmail.com, gus.cant@gmail.com

Abstract. *Aware of the many benefits that information technology can provide in the context of process management and mastery, this paper aims to develop web software that will contribute to the control of loans and item returns (keys, controls of air conditioning) belonging to an educational institution. To achieve this goal, good practices of the FDD agile development method were employed, as well as ASP.NET MVC technologies, Entity Framework, C# programming language, Bootstrap, Visual Studio development environment, and the SGBD SQL Server. In view of this, the proposed software was successfully developed.*

Resumo. *Ciente dos diversos benefícios que a tecnologia da informação pode proporcionar no contexto de gestão e domínio de processos, este trabalho tem por objetivo, desenvolver um software web, o qual contribuirá de forma propícia no controle de empréstimos e devoluções de itens (chaves, controles de ar-condicionado) pertencentes a uma instituição de ensino. Para atingir essa meta, foi empregado boas práticas do método de desenvolvimento ágil FDD, além das tecnologias ASP.NET MVC, Entity Framework, linguagem de programação C#, Bootstrap, ambiente de desenvolvimento Visual Studio e o SGBD SQL Server. Em vista disso, obteve-se êxito no desenvolvimento do software proposto.*

1. Introdução

Os sistemas de informação podem ser utilizados para controle e organização de vários processos em uma instituição de ensino. Com os constantes avanços tecnológicos, cada vez mais o ser humano vem se tornando dependente das novas tecnologias [Alfieri e Schutz 2012].

Muitas pessoas se sentem inseguras ou tem medo de mudanças pelo fato de já estarem acostumadas com o método de trabalho estabelecido a elas. O uso do papel, por exemplo, é um dos recursos ainda adotado por certas empresas ou instituições, contudo, não proporciona agilidade na execução dos processos, bem como não auxilia na gestão de informações, que são fundamentais para a administração das mesmas.

A mudança citada refere-se à utilização de sistemas de informação que, ao contrário de prejudicar a vida dos usuários, tem como propósito resultar em um trabalho facilitado, eficiente e que contribua para a evolução de instituições no âmbito da tecnologia.

Em consideração aos fatos iniciais, surgiu a concepção de desenvolver um sistema web que permitirá aos usuários do sistema realizar o controle de empréstimo de

itens, tais como: chaves, controles de ar-condicionado, entre outros. Esses itens pertencentes a uma instituição de ensino, criando mecanismos que não só tragam agilidade, mas também controle e eficácia na execução do processo.

1.1. Justificativa

Observado de fato o uso ineficiente de um caderno para realizar o controle de empréstimos e devoluções de itens em uma instituição de ensino, considerando também os equívocos ocasionados devido à grande quantidade de informações, surgiu a ideia de aperfeiçoar/automatizar esse processo utilizando tecnologias que possam facilitar e agilizar o mesmo.

1.2. Objetivo geral

Este trabalho tem como objetivo planejar e desenvolver um software web que favoreça o controle de empréstimos de itens em uma instituição de ensino e que conforte os responsáveis por essa função de modo a beneficiar seu serviço.

1.3. Objetivos específicos

Para este trabalho, os objetivos específicos encontrados são:

- Usufruir do ambiente de desenvolvimento integrado (IDE), Microsoft Visual Studio;
- Aplicar a plataforma ASP.NET MVC, a qual possibilita a implementação de aplicativos e sistemas web;
- Adotar boas práticas dos *frameworks*¹ *Entity Framework* e *Bootstrap*;
- Fazer uso do SGBD (Sistema Gerenciador de Banco de Dados) SQL Server a fim de armazenar os dados;
- Desenvolver um software que permita gerenciar informações de empréstimos e devoluções de itens e, também, pessoas, locais, itens, turmas, disciplinas, horários, cursos, entre outros;
- Possibilitar a geração de código de barras referente aos itens registrados no sistema;
- Possibilitar que os registros de empréstimos sejam feitos por meio da leitura do código de barras aderidos nos itens e nos crachás das pessoas vinculadas à instituição, a fim de facilitar/viabilizar o processo de empréstimos dos mesmos;
- Realizar testes de todos os módulos que compõem o software.

2. Referencial Teórico

Nesse segmento, sucederá a temática e as tecnologias que se fizeram imprescindíveis para o desenvolvimento do presente trabalho. A temática refere-se aos Sistemas de Informação para Controle e Gestão de Processos em uma Instituição de Ensino. Em relação às tecnologias, são ressaltados conceitos do ambiente de desenvolvimento

¹ *Framework* é um conjunto de técnicas, ferramentas ou conceitos pré-definidos utilizados para resolver um problema de um projeto ou domínio específico [Santos e Carvalho 2019].

Microsoft Visual Studio, da plataforma ASP.NET MVC, além do *Entity framework*, linguagem de programação C#, Bootstrap e o SGBD SQL Server. Outro tópico significativo define o padrão de arquitetura de software MVC.

2.1. Sistemas de informação para controle e gestão de processos em uma instituição de ensino

O desenvolvimento do software para a gestão/controle de empréstimos de itens enfatiza melhorar a agilidade e a consistência do processo, desse modo, possibilitar aos usuários maior praticidade e domínio de informações.

Segundo Ensinc (2017), a gestão eletrônica de documentos e a automatização de processos ressaltam um novo aspecto e uma nova imposição às instituições de ensino superior (IES). Tais inovações tecnológicas têm carregado inúmeras oportunidades para mudar, de forma a incrementar o ambiente organizacional.

Apesar disso, a inovação tecnológica gera um maior agregado e experiência aos usuários, além do êxito nos processos internos, redução de custo e eficiência, a fim de ser referência.

2.2. Arquitetura MVC (*Model, View, Controller*)

O padrão de arquitetura MVC separa uma aplicação em três grupos de componentes principais: *Models*, *Views* e *Controllers*. Empregando esse padrão, as solicitações de usuário são encaminhadas para um *Controller*, que é responsável por trabalhar com o *Model* para executar as ações do usuário e/ou recuperar os resultados de consultas. O *Controller* escolhe a *View* a ser exibida para o usuário e fornece-a com os dados do *Model* solicitado [Microsoft 2019e].

2.3. Tenologias

Esta seção apresentará as tecnologias computacionais utilizadas no desenvolvimento do sistema proposto.

2.3.1. Visual Studio

O Visual Studio é um ambiente de desenvolvimento integrado (IDE) da Microsoft destinado ao desenvolvimento de softwares. Ele permite que o editor de código e o depurador ofereçam suporte a inúmeras linguagens de programação, desde que exista um serviço específico para as mesmas. As linguagens internas incluem C# (Anexo A), C++, VB (Visual Basic), Python, JavaScript, entre outras. Também é um produto de desenvolvimento na área web, empregando a plataforma do ASP.NET MVC, como websites, aplicativos web, serviços web e aplicativos móveis [Microsoft 2019b].

2.3.2. ASP.NET MVC

Voltado à web e *open source*, o ASP.NET MVC foi criado pela Microsoft com o propósito de possibilitar a implementação de aplicativos e sistemas web modernos. Também é multiplataforma e pode ser executado no Windows, Linux, macOS e Docker [Microsoft 2019f].

2.3.3. Entity Framework

O *Entity Framework* é um *framework* ORM para .NET (Anexo B) desenvolvido pela Microsoft. Resumidamente, esse *frameworks* viabiliza o mapeamento das tabelas do banco em classes do projeto. Isto é, a partir de uma tabela “Pessoas” no banco de dados, um *framework* ORM facilita o mapeamento da mesma com uma possível classe “Pessoas” [Microsoft 2019a].

Além do mais, fornece uma série de recursos que aumentam significativamente a produtividade no desenvolvimento de aplicações persistentes.

2.3.4. Bootstrap

Considerado um *framework front-end*, o Bootstrap é uma ferramenta bastante usada para a criação de websites. Isso porque seus padrões seguem as convicções de usabilidade e ilustres tendências de design para interfaces [Leone 2017].

O *framework* Bootstrap conta com diversas classes em CSS (*Cascading Style Sheets*) prontas, além de plugins em JavaScript (jQuery) para implementar recursos - como *dropdowns*, carrosséis, *slideshows*, entre outros - de maneira acessível e com códigos otimizados [Barbiere 2017].

Apesar disso, sua padronização permite que os sites tenham um melhor aspecto pelo fato de suceder formas esteticamente agradáveis para a criação de páginas, resultando em maior satisfação dos usuários [Leone 2017].

2.3.5. SQL Server

O SQL Server é um SGBD (Sistema de Gerenciamento de Banco de Dados) relacional desenvolvido pela Microsoft para plataforma Windows. O SQL Server oferece os mais elevados níveis de segurança, confiabilidade e escalabilidade para garantir que aplicações e negócios estejam sempre em execução [Microsoft 2019c].

Distribuído em diferentes versões e com várias ferramentas integradas, esse SGBD é capaz de atender às demandas, desde os mais simples negócios até os mais complexos cenários, os quais operam grande volume de dados.

3. Trabalhos Correlatos

As pesquisas a seguir denotadas tiveram como foco principal trabalhos e/ou softwares que evidenciassem algum vínculo e que, de alguma forma, contribuíssem para o desenvolvimento do projeto.

3.1. Sistema web para reserva dos laboratórios de informática do Unasp (Centro Universitário Adventista de São Paulo)

O projeto proposto por Alfieri e Schutz (2012) teve como objetivo apresentar um sistema para automatizar o processo de reservas de laboratórios para os docentes, proporcionando um melhor controle e praticidade aos mesmos.

Constatou-se que, com o auxílio do sistema, o processo de reservas dos laboratórios tornou-se mais eficiente, expondo também segurança, clareza e controle na reserva dos mesmos. As duplicidades e colisões de horários foram extintas, visto que o

sistema não permite que sejam alocadas reservas para o mesmo horário [Alfieri e Schutz 2012]. As tecnologias utilizadas nesse projeto foram PHP, MySQL e HTML.

3.2. Software Wcond: controle de chaves - empréstimo e devolução

O wCond é um software web desenvolvido para a administração de condomínios, visto que disponibiliza diversos recursos para facilitar o dia a dia de administradores, síndicos, moradores e proprietários, sendo satisfatório pelo fato de unir todas as informações em um único ambiente de trabalho [Wcond 2017].

O software está implantado perante a infraestrutura de computação em nuvem, concedendo aos clientes uma melhor capacidade de processamento, disponibilidade e segurança nos serviços oferecidos. Além disso, uma política de backup automatizada e diária garante a integridade dos dados dos clientes.

Um dos recursos disponíveis na plataforma é o controle de chaves, o qual permite que o administrador realize o controle das chaves do condomínio, monitorando e acompanhando quem solicitou, quando foi emprestada e devolvida.

3.3. Software linear HCS Control Guarita: controle de chaves

O Control Guarita é um software desenvolvido exclusivamente para condomínios residenciais, ele permite gerenciar, de modo seguro e prático, todo o acesso de moradores, veículos e chaves, entre outros recursos.

O sistema possui várias ferramentas que facilitam as tarefas do dia a dia de um condomínio, além disso, conta com a integração da linha de produtos da Linear-HCS que garantem o controle de acesso, priorizando a segurança por meio de dispositivos de acionamento, como: cartão de proximidade, chaveiro, tag passivo e ativo, controle remoto, senha e biometria [Controlguarita 2017].

Um dos recursos disponíveis é o controle de entrega de chaves para moradores e prestadores, esse processo é muito importante, já que possibilita ter domínio absoluto sobre as retiradas e devoluções de chaves.

3.4. Conclusão sobre os trabalhos relacionados

Os trabalhos escolhidos se correlacionam com este projeto pelo fato de ressaltarem o desenvolvimento de sistemas para facilitar o controle e gestão de processos. O projeto de Alfieri e Schutz (2012), juntamente com os softwares Wcond (2017) e Control Guarita (2017), foi fundamental para a evolução deste projeto que objetiva por utilizar a prática do código de barras (aderidos nos itens, carteirinhas dos estudantes e crachás dos funcionários), no qual serão armazenados dados e realizada a leitura dos mesmos. Dessa forma, eleva a segurança e rastreabilidade dos itens e fornece aos usuários maior rapidez e acessibilidade na execução do processo (empréstimos e devoluções de itens).

4. Metodologia

Com propósito de alcançar bons resultados no desenvolvimento deste projeto, optou-se por usar boas práticas da metodologia FDD - *Feature Driven Development* (Anexo C). Um dos motivos dessa escolha vincula-se à conveniente técnica de modelagem e elaboração das *features*.

Nas seções posteriores, serão detalhadas as fases/etapas usadas na concepção desse software.

4.1. Desenvolvimento de modelo abrangente

Na fase inicial, realiza-se um estudo dirigido sobre o escopo do sistema e sua composição. Posteriormente, para cada processo a ser modelado, é realizado um detalhamento mais esclarecedor em relação ao domínio do negócio, resultando, assim, em um modelo que satisfaça o objeto do estudo.

Na Figura 1, é apresentado o fluxo de empréstimos de itens atualmente realizado em uma instituição de ensino.

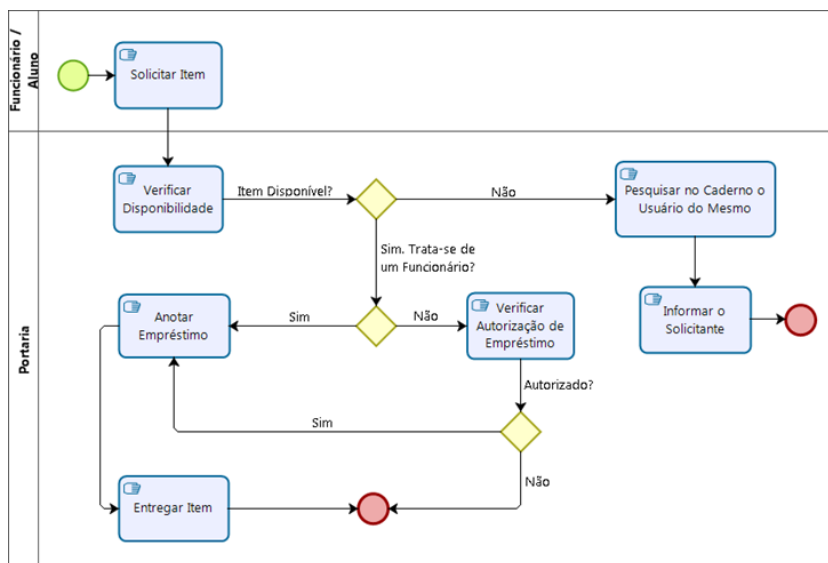


Figura 1. Fluxo empréstimos de itens

Complementado o fluxo apresentado anteriormente, na Figura 2 é apresentado o fluxo de devolução de itens.

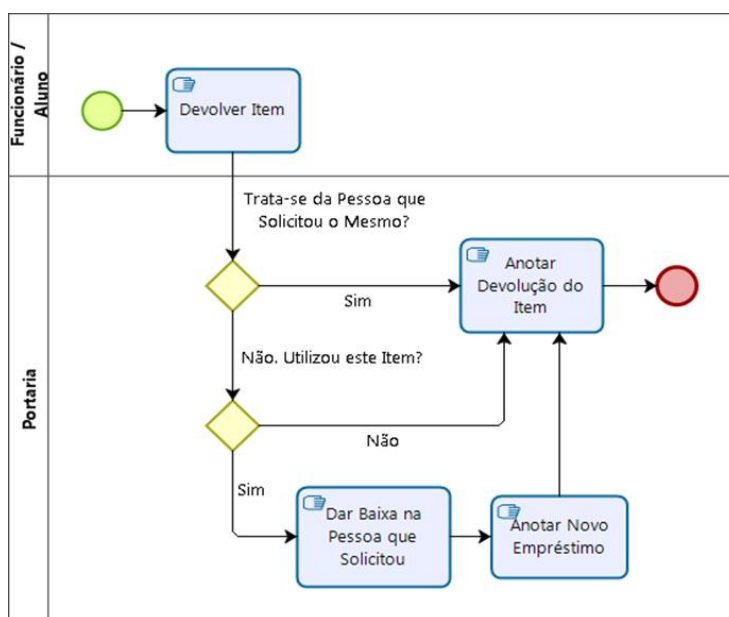


Figura 2. Fluxo devoluções de itens

4.2. Construção da lista de funcionalidades

Nesta fase, foram listadas todas as funcionalidades que atendem aos requisitos os quais irão compor o sistema a ser implementado. Esses requisitos são classificados em funcionais, isto é, descrevem o que o sistema fará, ou não funcionais, que retratam como o sistema fará.

Na Tabela 1, são listados os requisitos funcionais (RF), os quais especificam as funções e comportamentos do software.

Tabela 1. Requisitos funcionais (RF)

ID	Descrição	Relevância	Complexidade
RF1	Efetuar Login	Essencial	Baixa
RF2	Gerenciar Itens RF2.1 Pesquisar Itens RF2.2 Registrar Itens RF2.2.1 Gerar Código de Barras RF2.3 Alterar Itens RF2.4 Excluir Itens	Essencial	Média
RF3	Gerenciar Empréstimos de Itens RF3.1 Pesquisar Empréstimos RF3.2 Registrar Empréstimos	Essencial	Média
RF4	Gerenciar Devoluções de Itens RF4.1 Pesquisar Devoluções RF4.2 Registrar Devoluções	Essencial	Baixa
RF5	Gerenciar Autorizações de Empréstimos de Itens RF5.1 Pesquisar Autorizações de Empréstimos RF5.2 Registrar Autorizações de Empréstimos RF5.3 Alterar Autorizações de Empréstimos	Essencial	Média
RF6	Gerenciar Locais RF6.1 Pesquisar Locais RF6.2 Registrar Locais RF6.3 Alterar Locais RF6.4 Excluir Locais	Essencial	Baixa
RF7	Gerenciar Horários RF7.1 Pesquisar Horários RF7.2 Registrar Horários RF7.3 Alterar Horários RF7.4 Excluir Horários	Essencial	Baixa
RF8	Gerenciar Disciplinas RF8.1 Pesquisar Disciplinas RF8.2 Registrar Disciplinas RF8.3 Alterar Disciplinas RF8.4 Excluir Disciplinas	Essencial	Baixa
RF9	Gerenciar Cursos RF9.1 Pesquisar Cursos RF9.2 Registrar Cursos RF9.3 Alterar Cursos RF9.4 Excluir Cursos	Essencial	Baixa

(Continua)

(Conclusão)

ID	Descrição	Relevância	Complexidade
RF10	Gerenciar Turmas RF10.1 Pesquisar Turmas RF10.2 Registrar Turmas RF10.3 Alterar Turmas RF10.4 Excluir Turmas	Essencial	Baixa
RF11	Gerenciar Tipo Locais RF11.1 Pesquisar Tipo Locais RF11.2 Registrar Tipo Locais RF11.3 Alterar Tipo Locais RF11.4 Excluir Tipo Locais	Essencial	Baixa
RF12	Gerenciar Pessoas RF12.1 Pesquisar Pessoas RF12.2 Registrar Pessoas RF12.3 Alterar Pessoas RF12.4 Excluir Pessoas	Essencial	Baixa
RF13	Gerenciar Usuários RF13.1 Pesquisar Usuários RF13.2 Registrar Usuários RF13.3 Alterar Usuários RF13.4 Excluir Usuários	Essencial	Baixa
RF14	Gerar Relatórios RF14.1 Gerar relatórios de Empréstimos e Devoluções de Itens RF14.2 Gerar relatórios de Autorizações de Empréstimos de Itens	Desejável	Média

Dando continuidade, na Tabela 2 foram descritos os requisitos não funcionais (RNF), essenciais para o desenvolvimento do sistema.

Tabela 2. Requisitos não funcionais (RNF)

ID	Descrição	Relevância	Complexidade
RNF1	O sistema será implementado com o uso do ambiente de desenvolvimento Visual Studio e suas tecnologias (ASP.NET MVC, <i>Entity Framework</i> , entre outras).	Essencial	Baixa
RNF2	O sistema será implementado com o uso da Linguagem de Programação C#.	Essencial	Média
RNF3	O sistema fará uso do SGBD SQL Server.	Essencial	Média

4.3. Planejamento por funcionalidade

Além do planejamento das funcionalidades (Apêndice A), as quais irão constituir o software, visando uma boa percepção das *features* da aplicação, foi construído o diagrama de casos de uso (Figura 3) o qual evidencia, de forma estruturada, todas as funcionalidades do software e os atores que as executam. Baseando-se nesse diagrama, foi elaborada uma descrição de dois casos de uso mais relevantes para o contexto deste software (Apêndice B).

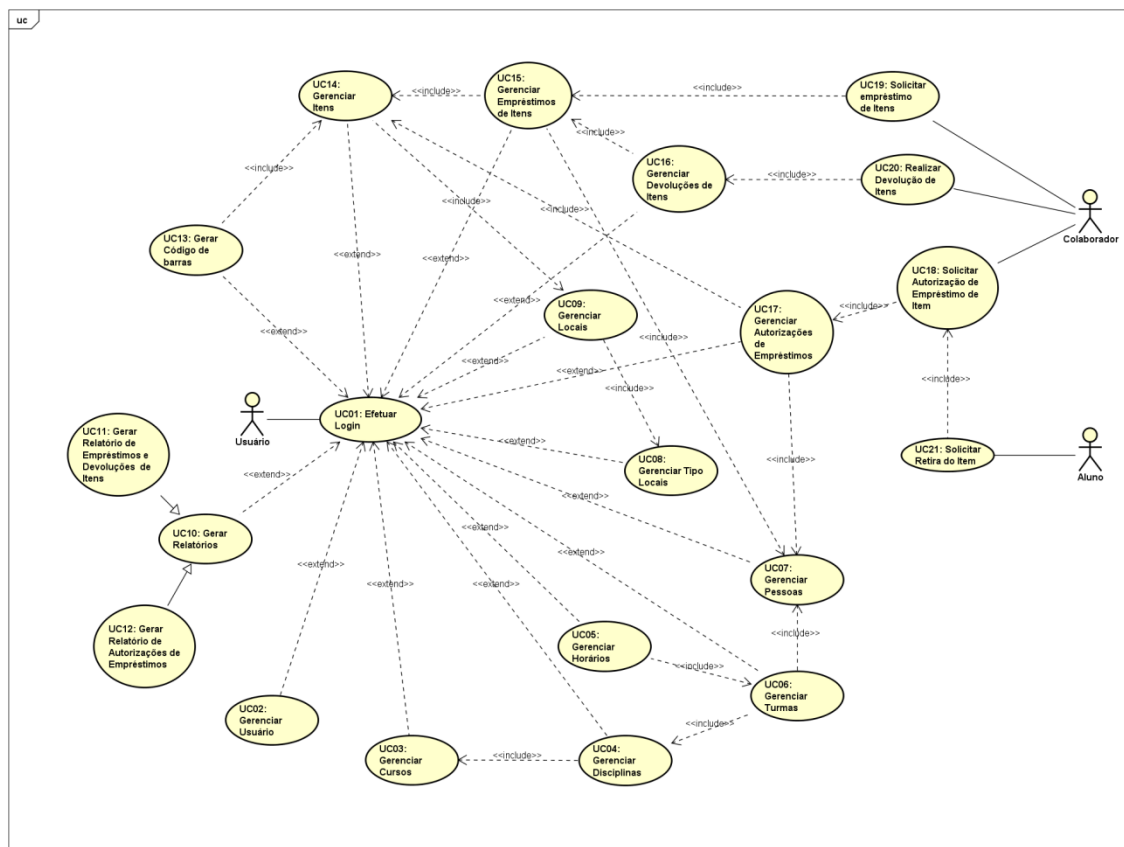


Figura 3. Diagrama de casos de uso

Outro aspecto significativo é compreender o fluxo de ações que serão executadas no sistema. Para isso, foram construídos diagramas de atividades responsáveis por representar tal finalidade, além de reproduzir o comportamento do software na realidade de negócio na qual irá submeter-se.

Na Figura 4, é apresentado o diagrama de atividades referente ao fluxo de gestão do sistema, o qual será realizado pelo usuário (administrador) do software.

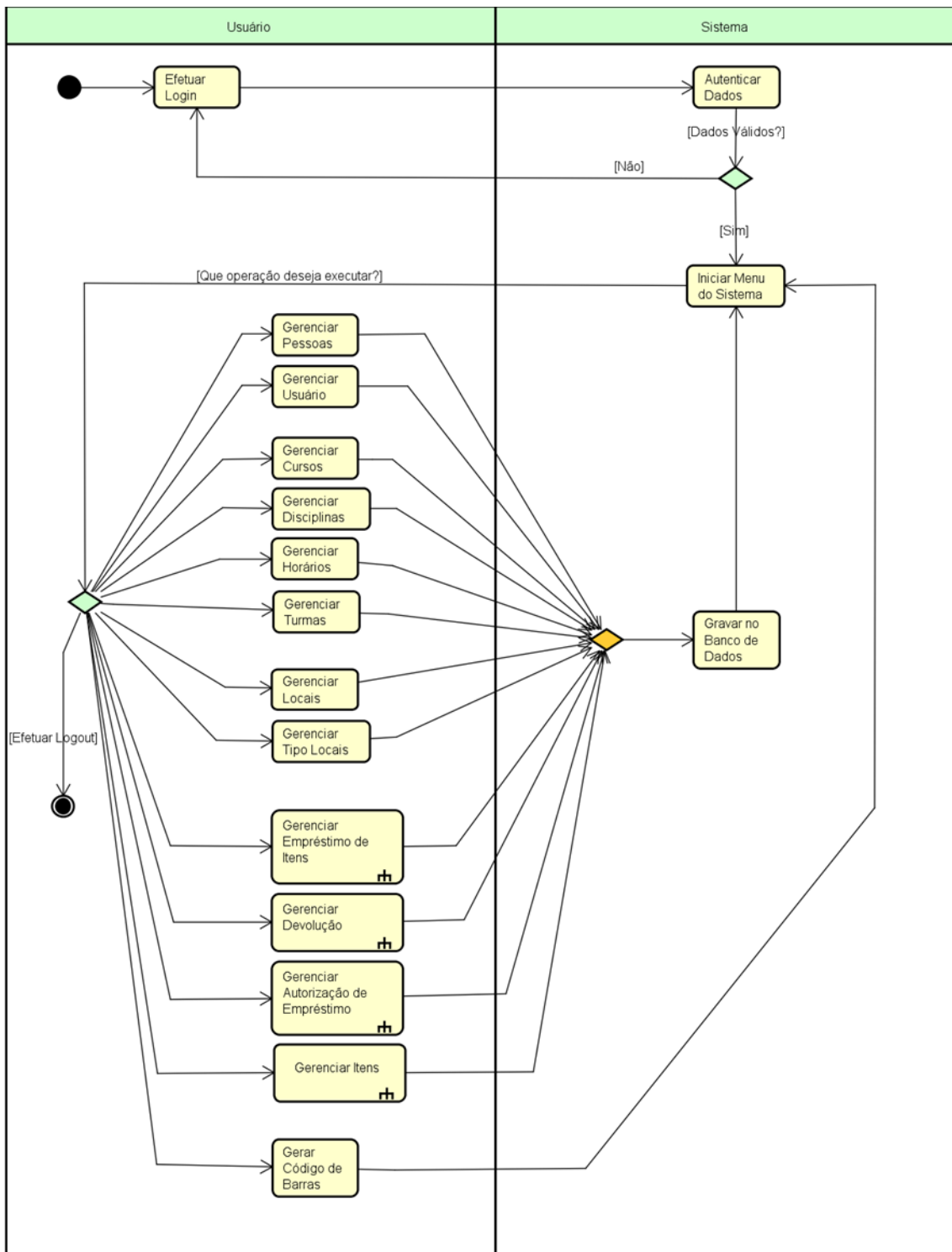


Figura 4. Diagrama de atividades sobre a gestão do sistema

Na seqüência (Figura 5), é apresentado o fluxo do subprocesso Registrar empréstimo, que incorpora o processo Gerenciar empréstimo de itens. Além da possibilidade da digitação das informações, poderá ser utilizado o leitor de código de barras para efetuar tal operação, que será realizada por meio da leitura dos códigos aderidos nos itens e nos crachás das pessoas vinculadas à instituição de ensino.

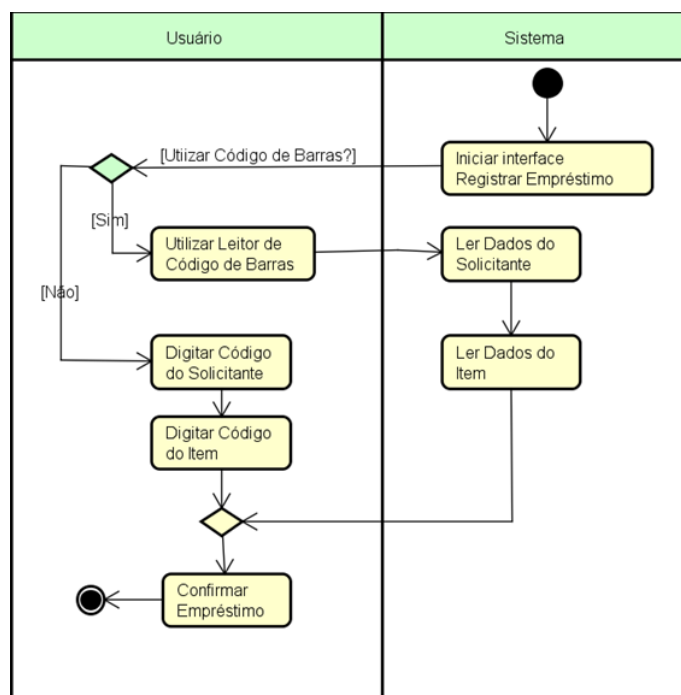


Figura 5. Diagrama de atividades registrar empréstimos de itens

Na Figura 6, é retratada a interação do funcionário/aluno com o usuário (administrador) do software a fim de realizar a solicitação de empréstimo de um determinado item.

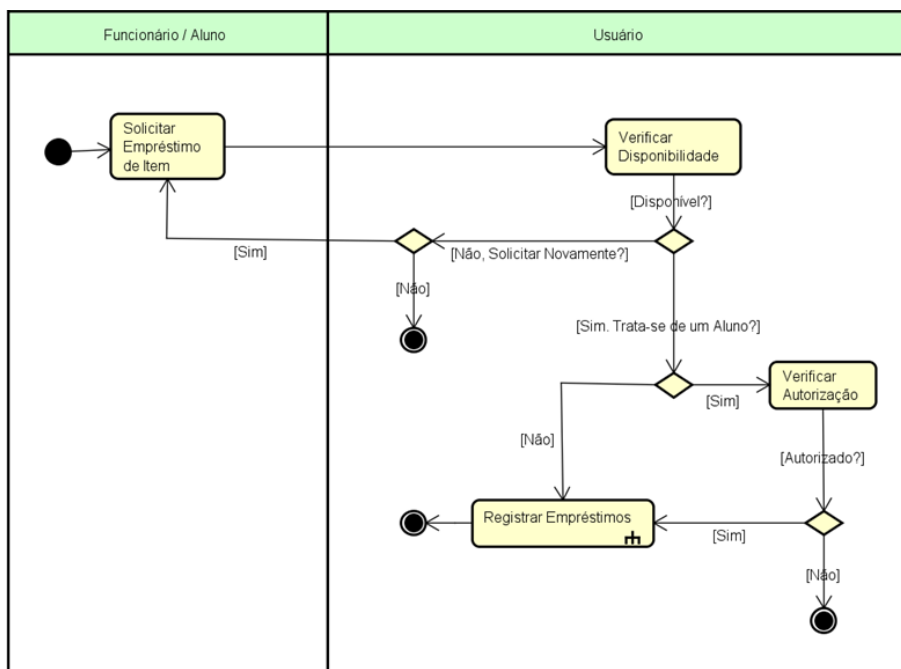


Figura 6. Diagrama de atividades solicitar empréstimo de itens

4.4. Detalhamento por funcionalidade

Nesta fase, é apresentado o Diagrama de Classes (Figura 7), que tem como finalidade facilitar a visualização das classes que constituirão o sistema, com seus respectivos atributos e métodos, além da relação entre as mesmas.

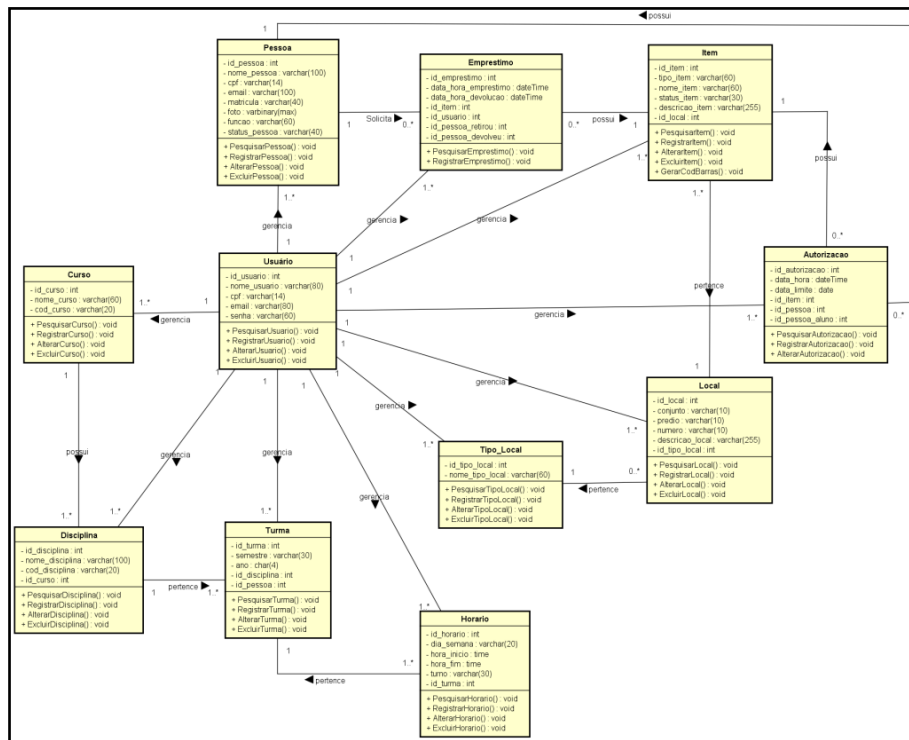


Figura 7. Diagrama de Classes

Expondo a estrutura do Banco de Dados deste projeto, apresenta-se, na Figura 8, o Diagrama Entidade Relacionamento (DER). Este, expõe uma visão lógica do banco de dados, viabilizando um conceito mais generalizado de como estão estruturados os dados de um sistema.

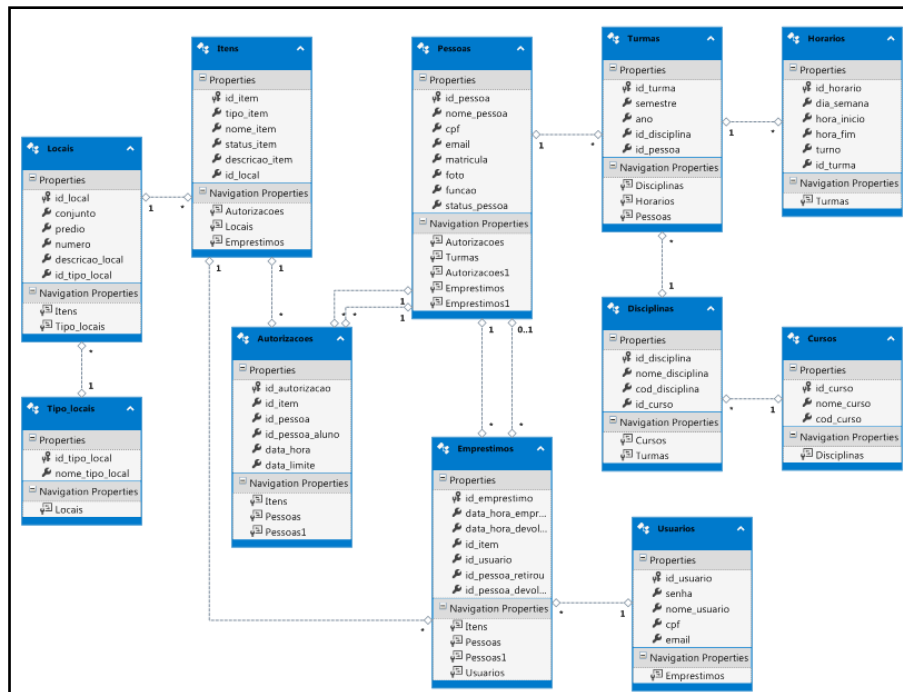


Figura 8. Diagrama Entidade Relacionamento

5. Construção por Funcionalidade

Baseando-se nos requisitos previamente definidos, realizou-se a implementação/codificação das funcionalidades do software (Apêndice D). Para tal fim, fez-se o uso das tecnologias conceituadas neste trabalho.

Isto posto, na Figura 9, pode-se analisar o método implementado dentro do *Controller* de Empréstimos para salvar os dados inseridos no registro do Empréstimo.

```
42 [HttpPost]
43 [ValidateAntiForgeryToken]
44 0 references
45 public ActionResult Create(Emprestimos emprestimos)
46 {
47     var statusI = db.Itens.First(i => i.id_item == emprestimos.id_item);
48     var statusP = db.Pessoas.First(p => p.id_pessoa == _idPessoaRetirou);
49
50     if (ModelState.IsValid)
51     {
52         //verifica se o itens esta disponivel
53         if (statusI.status_item == "Disponível")
54         {
55             //verifica se a pessoa esta autorizada para retirar um item
56             if (statusP.status_pessoa == "Autorizada")
57             {
58                 statusI.status_item = "Emprestado";
59                 emprestimos.data_hora_emprestimo = DateTime.Now;
60                 emprestimos.id_pessoa_retirou = _idPessoaRetirou;
61                 emprestimos.id_usuario = int.Parse(Session["id_usuario"].ToString());
62
63                 db.Emprestimos.Add(emprestimos);
64                 db.SaveChanges();
65                 return RedirectToAction("Index");
66             }
67             else
68             {
69                 MessageBox.Show(statusP.nome_pessoa + " não está autorizado(a) retirar itens");
70             }
71         }
72         else
73         {
74             MessageBox.Show("Item já emprestado");
75             return RedirectToAction("Create");
76         }
77     }
78
79     ViewBag.id_item = new SelectList(db.Itens, "id_item", "tipo_item", emprestimos.id_item);
80     ViewBag.id_pessoa_devolveu = "";
81     ViewBag.id_pessoa_devolveu = new SelectList(db.Pessoas, "id_pessoa", "nome_pessoa", emprestimos.id_pessoa_devolveu);
82     ViewBag.id_usuario = new SelectList(db.Usuarios, "id_usuario", "cpf", emprestimos.id_usuario);
83     return View(emprestimos);
84 }
```

Figura 9. Método Implementado na *Controller* de Empréstimos

Para possibilitar o empréstimo de um item, o mesmo deve estar cadastrado no sistema. A Figura 10 apresenta o método implementado dentro do *Controller* Itens, o qual salva os dados inseridos no registro de itens.

```
77 [HttpPost]
78 [ValidateAntiForgeryToken]
79 0 references
80 public ActionResult Create(Itens itens)
81 {
82     if (ModelState.IsValid)
83     {
84         db.Itens.Add(itens);
85         db.SaveChanges();
86         return RedirectToAction("Index");
87     }
88
89     ViewBag.id_local = new SelectList(db.Locais, "id_local", "descricao_local", itens.id_local);
90     return View(itens);
91 }
```

Figura 10. Método Implementado na *Controller* de Itens

6. Testes e Resultados

A concepção desta proposta resultou em um sistema para controle de itens pertencentes a uma instituição ensino. Para validar a conformidade do software desenvolvido, inicialmente, foram realizados testes utilizando dados fictícios. Logo, foi possível observar se o comportamento da aplicação correspondia aos requisitos preestabelecidos.

Após a certificação dos testes fictícios, apresentou-se o software à gerente de RH da instituição. Junto a ela, foram elaborados testes de usuário, os quais resultaram em feedbacks positivos. Segundo ela, é de suma importância que o controle de itens seja feito com o auxílio de um sistema de informação. Inclusive, ressaltou os benefícios que

este software proporcionará à instituição no quesito gestão e efetividade, além da grande quantidade de informações que irá fornecer aos usuários. Novas ideias e sugestões também agregaram valor à aplicação.

Esta proposta foi apresentada nos eventos SEPE (Simpósio de Ensino, Pesquisa e Extensão) e SIRC (Simpósio de Informática) no ano de 2019 na instituição UFN (Universidade Franciscana). Estas oportunidades foram importantes para expor/disseminar a ideia do trabalho, esclarecer dúvidas sobre o tema proposto, além dos feedbacks que ajudaram no engrandecimento do trabalho.

Referente à aplicação, na Figura 11, é apresentada a interface inicial do software, onde são disponibilizadas diversas funções aos usuários, destacando os módulos “Itens”, “Empréstimos” e “Autorizações”.

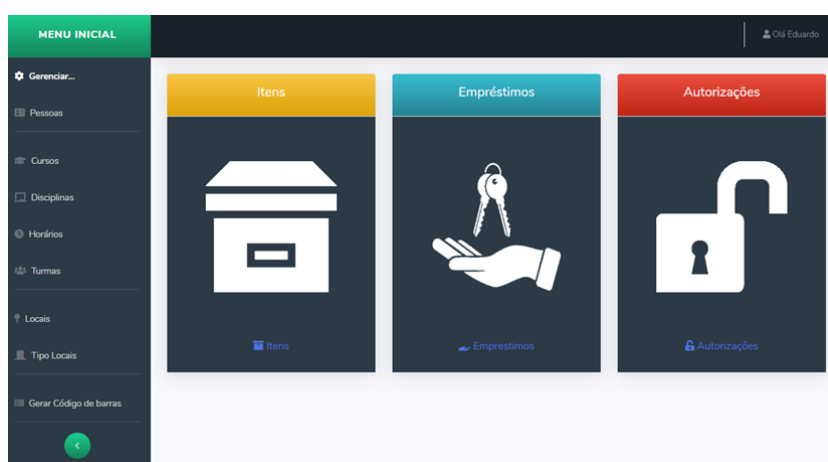


Figura 11. Interface menu inicial do software

A Figura 12, exibe a interface de gerenciamento de Itens, onde o usuário do sistema poderá realizar o cadastro, edição, exclusão e pesquisa de itens, além da possibilidade de gerar o código de barras dos mesmos.

A interface de gerenciamento de Itens mostra uma barra superior com 'Menu Inicial'. Abaixo, há uma seção 'Lista de Itens' com botões para 'Cadastrar Novo' e 'Gerar Código de Barras', além de filtros para 'Status' (Todos) e 'Tipo' (Todos), e um campo de 'Pesquisar'. A tabela principal contém os seguintes dados:

Ícone	Tipo	Nome	Status	Descrição	Local	Ações
	Chave	CHLABINF311P4C1	Emprestado	Chave do laboratório de informática.	Conjunto: 1 Prédio: 1 Número: 109 Tipo: Sala de Aula	Editar Excluir Cod. Barras
	Chave	CH_LAB108P4C1	Disponível	chave lab. de inf.	Conjunto: 1 Prédio: 4 Número: 311 Tipo: Laboratório de Informática	Editar Excluir Cod. Barras
	Controle Ar	CACLAB109P1C1	Emprestado	Controle ar-condicionado	Conjunto: 1 Prédio: 1 Número: 109 Tipo: Sala de Aula	Editar Excluir Cod. Barras
	Controle Ar	C_AR_LAB311P4C1	Disponível	Controle ar-condicionado do laboratório de informática	Conjunto: 1 Prédio: 1 Número: 109 Tipo: Sala de Aula	Editar Excluir Cod. Barras
	Chave	CH_SALA109P1C1	Disponível	Sala de aula	Conjunto: 1 Prédio: 1 Número: 109 Tipo: Sala de Aula	Editar Excluir Cod. Barras

Figura 12. Interface de gerenciamento de Itens

Na Figura 13, o usuário terá duas opções para realizar o processo de empréstimos de itens. A primeira delas consiste em digitar manualmente o “ID do Item”

e, respectivamente, a “Matrícula da Pessoa”. Já a segunda, considerada a mais eficaz, é possível utilizar leitores de código de barras para fazer a leitura dos códigos.

Menu Inicial

Cadastro Empréstimo

Item

Retirou

Nome Pessoa

Figura 13. Interface Cadastro de Empréstimo

7. Conclusão

O presente trabalho apresentou o desenvolvimento de um software web para controle de empréstimos e devoluções de itens pertencentes a uma instituição de ensino. Em decorrência da falta de um sistema para exercer determinada prática (controle de empréstimos de itens), após realizados testes, verificou-se que este software tende a proporcionar aos usuários mais agilidade e segurança na execução da atividade estipulada, além de assegurar uma melhor gestão dos itens.

As pesquisas realizadas durante a elaboração do trabalho viabilizaram amplo amadurecimento e novos princípios para concepção do tema proposto. Outro fator significativo remete à escolha das tecnologias a serem utilizadas, além da metodologia ágil FDD, a qual visa melhor qualidade na gestão e planejamento para o processo de desenvolvimento do software.

O sistema desenvolvido possibilita ao administrador, além de seu registro, o cadastro de Itens, Pessoas, Cursos, Disciplinas, Horários, Turmas, Locais e Tipo Locais, bem como a alteração, exclusão e consulta destes. Outros módulos, como o de Empréstimos/Devoluções e Autorizações, complementam o sistema.

Diante disso, observaram-se trabalhos/projetos futuros que poderão ser desenvolvidos para aprimorar ainda mais o controle/gestão por parte do administrador do sistema, como o desenvolvimento de um módulo para possibilitar saber quais locais são mais ou menos usados/frequentados, bem como a criação de um aplicativo multiplataforma, o qual possibilitará o desenvolvimento de um módulo capaz de enviar notificações via WhatsApp, informando ao colaborador/aluno a não devolução do item.

Referências

- Alfieri, A. M., e Schutz, L. H. (2012) “Sistema web para Reserva dos laboratórios de Informática do Unasp”, <https://docplayer.com.br/555348-Sistema-web-para-reserva-dos-laboratorios-de-informatica-do-unasp.html>, Março.
- Araújo, E. (2013) “Introdução à linguagem C#”. <https://www.devmedia.com.br/introducao-a-linguagem-csharp/27711>, Novembro.

- Barbiere, L. (2017) “O Que é Bootstrap e Para Que Serve?”, <https://www.ciawebsites.com.br/dicas-e-tutoriais/o-que-e-bootstrap/>, Maio.
- Controlguarita. (2017) “Control Guarita”, <http://controlguarita.com.br/>, Abril.
- Ensinc. (2017) “A Automação de Processos nas Instituições de Ensino Superior.”, <http://www.ensinc.com.br/site/automacao-de-processos/>, Dezembro.
- Leone, L. (2017) “Bootstrap: o que é, porque usar e como começar com o *framework*”, <https://becode.com.br/bootstrap-o-que-e-porque-usar-e-como-comecar/>, Maio.
- Marques, A. N. (2012) “Metodologias ágeis de desenvolvimento: processos e comparações”, <http://www.fatecsp.br/dti/tcc/tcc00064.pdf>, Março.
- Microsoft. (2019a) “Documentação do Entity Framework”, <https://docs.microsoft.com/pt-br/ef/>, Dezembro.
- Microsoft. (2019b) “Documentação do IDE do Visual Studio”, <https://docs.microsoft.com/pt-br/visualstudio/opbuildpdf/73e0edff/toc.pdf?branch=live&view=vs-2019>, Dezembro.
- Microsoft. (2019c) “Introdução ao SQL Server 2019”, <https://www.microsoft.com/pt-br/sql-server/sql-server-2019>, Dezembro.
- Microsoft. (2019d) “O que é o .NET?”, <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>, Dezembro.
- Microsoft. (2019e) “Visão geral sobre o ASP.NET MVC”, <https://docs.microsoft.com/pt-br/aspnet/core/mvc/overview?view=aspnetcore-3.1>, Dezembro.
- Microsoft. (2019f) “What is ASP.NET?”, <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet>, Novembro.
- Monteiro, E. A. (2013) “Utilização de Técnicas Ágeis em Projetos Exclusivos de Teste de Software”, <https://acervodigital.ufpr.br/bitstream/handle/1884/48832/R%20-%20E%20-%20EDIVALDO%20ANTONIO%20MONTEIRO.pdf?sequence=1&isAllowed=y>, Dezembro.
- Retamal, A. M. (2008) “Feature-Driven-Development”, <http://homes.dcc.ufba.br/~mauricio052/Engenharia%20de%20Software%20I/FDD/FDD-Processos.pdf>, Março.
- Santos, A. H, e Carvalho, N. R. (2019) “Frameworks e seus Benefícios no Desenvolvimento de Software”, http://revistapensar.com.br/tecnologia/pasta_upload/artigos/a95.pdf, Dezembro.
- Teixeira, R. (2011) “Engenharia de Software – FDD”, https://prezi.com/wzrm_icbitk4/engenharia-de-software-fdd/, Abril.
- Wcond. (2017) “Plataforma para Condomínios”, <https://www.wcond.com/o-que-e>, Maio.

Anexo A. Linguagem de Programação C#

C# é uma linguagem de programação originada pela Microsoft como parte da plataforma .NET. Por ter uma sintaxe parecida com o Java e C++, o C# pode ser indicado mesmo para programadores principiantes, que já conhecem a programação Orientada a Objetos. Com ela pode-se criar desde aplicações web, desktop, mobile, incluindo jogos eletrônicos [Araújo 2013].

Anexo B. .NET

O .NET é uma plataforma de desenvolvimento composta de ferramentas, bibliotecas e linguagens de programação designada a concepção de diversos modelos de aplicações [Microsoft 2019d].

Anexo C. *Feature Driven Development* (FDD)

O FDD (*Feature Driven Development*), que significa desenvolvimento guiado por funcionalidades, é uma metodologia ágil utilizada no gerenciamento e desenvolvimento de softwares. Ele estabelece as melhores práticas para o gerenciamento ágil de projetos com uma abordagem completa para Engenharia de Software orientada a objetos [Teixeira 2011].

Também, ressalta a produção de softwares de qualidade e funcionais. Propõe a realização de um trabalho significativo desde a primeira etapa do projeto, além disso, fornece informações de estado e progresso de forma simples e compreensíveis, agradando, assim, os gerentes e desenvolvedores.

A metodologia ágil FDD possui cinco fases, essas são definidas como: desenvolver um modelo abrangente, construir a lista de funcionalidades, planejar por funcionalidade, detalhar por funcionalidade e construir por funcionalidade.

- **Desenvolver um Modelo Abrangente:** O desenvolvimento do modelo abrangente é uma atividade introdutória da metodologia que abrange todo o projeto, geralmente realizada por membros do domínio do negócio e por desenvolvedores [Retamal 2008];
- **Construir a Lista de Funcionalidades:** Nessa fase o entendimento do escopo é ampliado através da identificação das funcionalidades do sistema a ser implementado [Monteiro 2013].
- **Planejar por Funcionalidade:** O objetivo desta fase é produzir o plano de desenvolvimento do sistema, levando em consideração a ordem na qual as funcionalidades serão implementadas, se há dependências entre elas, estimativa de tempo para desenvolvê-las e sua complexidade de implementação [Retamal 2008];
- **Detalhar por Funcionalidade:** Esta fase tem o intuito de reproduzir de forma mais aprofundada e detalhada os requisitos anteriormente determinados, além disso, demonstra um melhor esclarecimento do fluxo de execução do sistema e é concebido o mapeamento das classes e métodos que serão empregados no desenvolvimento das funcionalidades do sistema [Marques 2012].

- **Construir por Funcionalidade:** Esta fase do processo ocorre na iteração de construção, na qual o código referente a cada funcionalidade do produto é realmente implementado/codificado pelo desenvolvedor [Monteiro 2013].

Apêndice A. Planejamento Detalhado das Funcionalidades do Software

A Tabela 3 expõe o planejamento, o qual resultará no desdobramento das funcionalidades julgadas essenciais para este projeto.

Tabela 3. Planejamento por funcionalidades

Sequência	Funcionalidade	Tempo de Desenvolvimento Estimado (Hrs)
1	RF2. Gerenciar Itens	30
2	RF7. Gerenciar Horários	20
3	RF9. Gerenciar Cursos	10
4	RF8. Gerenciar Disciplinas	10
5	RF10. Gerenciar Turmas	10
6	RF6. Gerenciar Locais	10
7	RF11. Gerenciar Tipo Locais	10
8	RF12. Gerenciar Pessoas	30
9	RF13. Gerenciar Usuários	20
10	RF3. Gerenciar Empréstimos de Itens	80
11	RF4. Gerenciar Devoluções de Itens	50
12	RF5. Gerenciar Autorizações de Empréstimos de Itens	50
13	RF14. Gerar Relatórios	20
14	RF1. Efetuar Login	20

Apêndice B. Detalhamento dos Descritivos de Caso de Uso

O primeiro descritivo, apresentado na Tabela 4, esclarece os passos do caso de uso “Gerenciar empréstimos de itens”.

Tabela 4. Descritivo do caso de uso gerenciar empréstimos de itens

Identificação	UC15.
Caso de Uso	Gerenciar Empréstimos de Itens.
Descrição	Permite que o usuário pesquise e registre dados de um determinado empréstimo.
Ator Principal	Usuário.
Ator (es) Secundário(s)	N/A
Pré Condições	
PC1	O usuário deve estar logado no sistema.
Pós Condições	
PO1	Empréstimo pesquisado, registrado ou excluído.
Fluxo Principal	
FP1	O sistema exibe a interface de Gerenciar Empréstimos, o qual disponibiliza as seguintes operações: Pesquisar Empréstimos [FA1], Registrar Empréstimos [FA2].

(Continua)

(Conclusão)

FP2	Fim do caso de uso.
Fluxo Alternativo	
FA1	Pesquisar Itens: FA1. 1. O usuário filtra por solicitante(s) e/ou item(s) e/ou status e/ou realiza uma pesquisa mais objetiva. FA1. 2. O sistema exibe o resultado da Pesquisa.
FA2	Registrar empréstimo: FA2. 1. O usuário executa o [FA1] [FE1]. FA2. 2. O usuário seleciona o item o qual será emprestado. FA2. 3. O usuário seleciona a opção para iniciar o processo de empréstimo. FA2. 4. O sistema exibe a interface de realizar empréstimo. FA2. 5. O usuário efetua a leitura do código de barras existente no crachá do solicitante ou digita o código do mesmo. FA2. 6. O sistema exibe as informações referentes ao solicitante. FA2. 7. O usuário digita o código do item ou efetua a leitura do código de barras do mesmo. FA2. 8. O sistema exibe as informações referentes ao item. FA2. 9. O usuário realiza o empréstimo. FA2. 10. O sistema exibe mensagem de confirmação do empréstimo. FA2. 11. O sistema retorna ao [FP1].
Fluxo de Exceções	
FE1	Caso o item esteja reservado ou emprestado, é informado ao solicitante.
Regras de Negócio	N/A
Casos de Uso Incluídos	UC14, UC07

Seguindo a mesma lógica, na Tabela 5, é mostrado o descritivo referente ao caso de uso “Gerenciar itens”.

Tabela 5. Descritivo do caso de uso gerenciar itens

Identificação	UC14
Caso de Uso	Gerenciar Itens.
Descrição	Permite que o usuário pesquise, registre, altere e exclua dados de um determinado item.
Ator Principal	Usuário
Ator (es) Secundário(s)	N/A
Pré Condições	
PC1	O usuário deve estar logado no sistema.
Pós Condições	
PO1	Item pesquisado, Registrado, alterado ou excluído.
Fluxo Principal	
FP1	O sistema exibe a interface de Gerenciar Itens, o qual disponibiliza

(Continua)

(Conclusão)

Fluxo Principal	
FP1	as seguintes operações: Pesquisar Itens [FA1], Registrar Itens [FA2], Alterar Itens [FA3] ou Excluir Itens [FA4].
FP2	Fim do caso de uso.
Fluxo Alternativo	
FA1	Pesquisar Itens: FA1. 1. O usuário filtra por tipo de item(s) e/ou realiza uma pesquisa mais objetiva. FA1. 2. O sistema exibe o resultado da Pesquisa.
FA1	Incluir Itens: FA1. 1. O usuário seleciona a opção para iniciar o processo de registro do item. FA1. 3. O sistema exibe a interface de registrar item. FA1. 4. O usuário preenche os campos. FA1. 5. O usuário realiza o registro do item. FA1. 6. O sistema gera o código de barras referente o item. FA1. 7. O sistema exibe mensagem de confirmação do registro do item e se deseja imprimir o código de barras. FA1. 8. O sistema retorna ao [FP1].
FA2	Alterar Itens: [FE1] FA2. 1. O usuário executa o [FA1]. FA2. 2. O usuário seleciona o item o qual será alterado. FA2. 3. O usuário seleciona a opção para iniciar o processo de alteração do item. FA2. 4. O sistema exibe a interface de alteração de itens. FA2. 5. O usuário altera os dados referentes ao item. FA2. 6. O usuário realiza a alteração do item. FA2. 7. O sistema exibe mensagem de confirmação da alteração do item. FA2. 8. O sistema retorna ao [FP1].
FA3	Excluir item: [FE1] FA3. 1. O usuário executa o [FA1]. FA3. 2. O usuário seleciona o item o qual será excluído. FA3. 3. O usuário seleciona a opção para iniciar o processo de exclusão do item. FA3. 4. O sistema exibe mensagem de confirmação da exclusão do item. FA3. 5. O sistema retorna ao [FP1].
Fluxo de Exceções	
FE1	Caso o item esteja emprestado ou reservado, não poderá ser alterado ou excluído.
Regras de Negócio	N/A
Casos de Uso Incluídos	UC09

Apêndice C. Interfaces do Software

Menu Inicial

Cadastro Item

Tipo Item

Chave
 Controle Ar-Condicionado

Nome

Status

Disponível
 Emprestado

Descrição

Local

Figura 14. Interface Cadastro de Item

Menu Inicial

Lista de Locais

Conjunto: Prédio: Número:

Conjunto	Prédio	Número	Descrição	Tipo Local		
1	1	109	Sala de aula	Sala de Aula	<input type="button" value="Editar"/>	<input type="button" value="Deletar"/>
1	4	311	Laboratório de informática	Laboratório de Informática	<input type="button" value="Editar"/>	<input type="button" value="Deletar"/>
1	3	108	Laboratório de Informática	Laboratório de Informática	<input type="button" value="Editar"/>	<input type="button" value="Deletar"/>
1	4	316	Laboratório de Informática	Laboratório de Informática	<input type="button" value="Editar"/>	<input type="button" value="Deletar"/>
1	3	203	Laboratório de informática...	Laboratório de Informática	<input type="button" value="Editar"/>	<input type="button" value="Deletar"/>
3	17	102	Salão de atos	Salão de Atos	<input type="button" value="Editar"/>	<input type="button" value="Deletar"/>

Figura 15. Interface Gerenciamento de Locais

Menu Inicial

Cadastro Local

Conjunto

Prédio

Número

Descrição

Tipo Local

Figura 16. Interface Cadastro de Local

Menu Inicial

Lista de Pessoas

[Cadastrar Nova](#)

Status: Todas | Matricula: [Pesquisar](#)

Nome	CPF	E-mail	Matricula	Foto	Função	Status	
Alexandre	842.356.899-66	xande@xande.com	888888888		Colaborador	Autorizada	Editar Excluir
Sylvio	879.354.849-65	syvio@syvio.com	777777777		Colaborador	Autorizada	Editar Excluir
Eduardo Rossato Alessio	040.966.380-80	dudu_alessio98@hotmail.com	2016010514		Aluno	Autorizada	Editar Excluir
Gustavo Cantarelli	040.966.440-56	gus@gus.com	5214785698		Colaborador	Autorizada	Editar Excluir

Figura 17. Interface Gerencimento de Pessoas

Menu Inicial

Cadastro Pessoa

Nome

CPF

E-mail

Matricula

Foto Nenhum arq... lecionado

Função

- Aluno
- Colaborador

Status

- Autorizada
- Não Autorizada

[Cadastrar](#)

[Voltar](#)

Figura 18. Interface Cadastro de Pessoa

Apêndice D. Trechos de Código

```
2 namespace ControleItens.Models
3 {
4     using System;
5     using System.Collections.Generic;
6     using System.ComponentModel.DataAnnotations;
7
8     18 referencias
9     public partial class Empréstimos
10    {
11        [Required(ErrorMessage = "Este campo é obrigatório!")]
12        0 referencias
13        public int id_emprestimo { get; set; }
14
15        [Required(ErrorMessage = "Este campo é obrigatório!")]
16        [Display(Name = "Data Empréstimo")]
17        1 referencias
18        public System.DateTime data_hora_emprestimo { get; set; }
19
20        [Display(Name = "Data Devolução")]
21        0 referencias
22        public Nullable<System.DateTime> data_hora_devolucao { get; set; }
23
24        [Required(ErrorMessage = "Este campo é obrigatório!")]
25        [Display(Name = "Item")]
26        7 referencias
27        public int id_item { get; set; }
28
29        [Required(ErrorMessage = "Este campo é obrigatório!")]
30        [Display(Name = "Usuário")]
31        6 referencias
32        public int id_usuario { get; set; }
33
34        [Required(ErrorMessage = "Este campo é obrigatório!")]
35        [Display(Name = "Retirou")]
36        5 referencias
37        public int id_pessoa_retirou { get; set; }
38
39        [Display(Name = "Devolveu")]
40        5 referencias
41        public Nullable<int> id_pessoa_devolveu { get; set; }
42
43        1 referencias
44        public virtual Itens Itens { get; set; }
45
46        1 referencias
47        public virtual Pessoas Pessoas { get; set; }
48
49        1 referencias
50        public virtual Pessoas Pessoas1 { get; set; }
51
52        1 referencias
53        public virtual Usuarios Usuarios { get; set; }
54    }
55 }
```

Figura 19. Classe Empréstimos Implementada no *Model* de Empréstimos


```
159 public ActionResult GerarCodigoDeBarra(int? codigoDeBarra)
160 {
161     List<Models.Itens> listaItens = new List<Itens>();
162
163     if (codigoDeBarra != null)
164     {
165         //consulta com where
166         listaItens = (from itens in db.Itens
167                     where itens.id_item == codigoDeBarra
168                     select itens)
169                     .ToList();
170     }
171     else
172     {
173         //consulta completa
174         listaItens = (from itens in db.Itens
175                     select itens)
176                     .ToList();
177     }
178
179     byte[] arquivo = GerarPdf(listaItens);
180
181     MemoryStream memoryStream = new MemoryStream();
182
183     memoryStream.Write(arquivo, 0, arquivo.Length);
184     memoryStream.Position = 0;
185
186     return File(memoryStream, "application/pdf", "codigo.pdf");
187 }
```

Figura 20. Método GerarCodigoDeBarras Implementado na *Controller* de Itens

```

13 <div class="form-horizontal">
14 <hr />
15 @Html.ValidationSummary(true, "", new { @class = "text-danger" })
16 <div class="form-group">
17     @Html.LabelFor(model => model.tipo_item, htmlAttributes: new { @class = "control-label col-md-2" })<br />
18     <div class="col-md-10">
19         @Html.RadioButtonFor(model => model.tipo_item, "Chave", new { Checked = "checked" }) Chave <br />
20         @Html.RadioButtonFor(model => model.tipo_item, "Controle Ar") Controle Ar-Condicionado
21         @Html.ValidationMessageFor(model => model.tipo_item, "", new { @class = "text-danger" })
22     </div>
23 </div>
24
25 <div class="form-group">
26     @Html.LabelFor(model => model.nome_item, htmlAttributes: new { @class = "control-label col-md-2" })
27     <div class="col-md-10">
28         @Html.EditorFor(model => model.nome_item, new { htmlAttributes = new { @class = "form-control" } })
29         @Html.ValidationMessageFor(model => model.nome_item, "", new { @class = "text-danger" })
30     </div>
31 </div>
32
33 <div class="form-group">
34     @Html.LabelFor(model => model.status_item, htmlAttributes: new { @class = "control-label col-md-2" })<br />
35     <div class="col-md-10">
36         @Html.RadioButtonFor(model => model.status_item, "Disponível", new { Checked = "checked" }) Disponível<br />
37         @Html.RadioButtonFor(model => model.status_item, "Emprestado") Emprestado
38         @Html.ValidationMessageFor(model => model.status_item, "", new { @class = "text-danger" })
39     </div>
40 </div>
41
42 <div class="form-group">
43     @Html.LabelFor(model => model.descricao_item, htmlAttributes: new { @class = "control-label col-md-2" })
44     <div class="col-md-10">
45         @Html.TextAreaFor(model => model.descricao_item, new { htmlAttributes = new { @class = "form-control" } })
46         @Html.ValidationMessageFor(model => model.descricao_item, "", new { @class = "text-danger" })
47     </div>
48 </div>
49
50 <div class="form-group">
51     @Html.LabelFor(model => model.id_local, "Local", htmlAttributes: new { @class = "control-label col-md-2" })
52     <div class="col-md-10">
53         @Html.DropDownList("id_local", null, "Selecione", htmlAttributes: new { @class = "form-control" })
54         @Html.ValidationMessageFor(model => model.id_local, "", new { @class = "text-danger" })
55     </div>
56 </div>
57
58 <div class="form-group">
59     <div class="col-md-offset-2 col-md-10">
60         <input type="submit" value="Cadastrar" class="btn btn-info" />
61     </div>
62 </div>
63 </div>

```

Figura 21. Formulário de Registro de Itens Implementado na *View* de Itens