

# Desenvolvimento de um sistema de informação para gerenciamento de ativos em laboratórios de informática

Gabriel Gregório Silveira Netto<sup>1</sup>, Gustavo Stangherlin Cantarelli<sup>1</sup>

1 Curso de Sistemas de informação - Universidade Franciscana (UFN)  
97010-491 - Santa Maria – RS – Brasil

{gabrielnetto@ufn.edu.br, gust.cant@gmail.com}

**Abstract.** *This work proposed the development of a web application to control information in the information technology laboratories of the Universidade Franciscana. The system contains computer inventory information, registered software, change history and features for file transfer and remote access download via Windows Terminal Service. The system was implemented using the PHP programming language, Falcon Admin Dashboard template, jQuery JavaScript library and SQL Server database system. To develop the work, the Feature Driven Development methodology was adopted. The work achieved the objectives set out in the planning, meeting the demands of the IT Support sector that will use the software.*

**Resumo.** *Este trabalho propôs o desenvolvimento de uma aplicação web para controle das informações dos laboratórios de informática da Universidade Franciscana. O sistema contém informações de inventário dos computadores, softwares cadastrados, histórico de alteração e funcionalidades para transferência de arquivos e download de acesso remoto via Terminal Service do Windows. Para a implementação do sistema foi utilizada a linguagem de programação PHP, template Falcon Admin Dashboard, biblioteca javascript jQuery e sistema de banco de dados SQL Server. Para desenvolver o trabalho foi adotado a metodologia Feature Driven Development. O trabalho atingiu os objetivos previstos no planejamento, atendendo as demandas do setor de Suporte de TI que utilizará o software.*

## 1. Introdução

A gestão de ativos de TI (tecnologia da informação) tem por finalidade cuidar de todos os componentes de tecnologia da organização, sejam eles virtuais ou físicos [Gomes 2019]. Por meio da gestão de ativos é possível inventariar equipamentos como servidores, switches, computadores ou softwares, mantendo informações como data de compra do ativo, descrição e onde está localizado fisicamente na organização.

A gestão de ativos de TI é de grande importância para a organização, pois através dela é possível antecipar problemas, controlar o suporte de ativos adquiridos, aumentar a vida útil e controlar o uso dos equipamentos [Schultz 2018].

Tratando-se de antecipação de problemas, com base no controle de ativos de TI, é possível verificar quais são as fraquezas da infraestrutura de tecnologia da informação e quais são os pontos que necessitam de correção ou reestruturação [Schultz 2018].

A manutenção de equipamentos tecnológicos realizada em momentos certos auxiliam na redução de custos para a organização e prolongam a vida útil dos ativos,

através da gestão de ativos é possível identificar qualquer problema que possa ocorrer com os ativos e corrigi-lo antes que possa prejudicar permanentemente o ativo [Schultz 2018].

### **1.1. Justificativa**

Tendo em vista a numerosa quantidade de laboratórios de informática existentes na Universidade Franciscana (UFN) e a grande variedade de softwares utilizados em cada um dos laboratórios surge a necessidade de gerir estes ativos de forma mais otimizada.

Até o momento deste trabalho, o controle destas informações ainda é realizado de forma manual, através das ferramentas do Microsoft Office, como planilhas eletrônicas e documentos de texto. Estas informações por vezes acabam duplicadas ou dispersas em diferentes arquivos.

### **1.2. Objetivo Geral**

O trabalho teve por objetivo desenvolver um sistema web para gerenciamento dos laboratórios de informática da instituição.

O sistema mantém o controle dos softwares que estão instalados em cada laboratório, a configuração de hardware, armazena informações de rede, mantém um histórico de alteração de hardware de cada máquina e um histórico de quando houver troca do conjunto das máquinas de um laboratório.

O sistema tem por finalidade obter melhor eficiência no controle das informações e das manutenções realizadas.

Ao final do desenvolvimento do projeto, o sistema foi integrado a um outro sistema existente que auxilia o setor de Infraestrutura e redes, e o setor de Suporte de TI que é responsável pela manutenção dos laboratórios da instituição.

### **1.3. Objetivos Específicos**

Os objetivos específicos para o presente trabalho são:

- Fazer uso da linguagem de programação PHP (*Hypertext Preprocessor*).
- Fazer uso do *framework Bootstrap* com o tema *Falcon*, para criação de interfaces responsivas.
- Utilizar o SGBD Microsoft SQL Server para gerir os dados da aplicação devido a segurança que ele proporciona.
- Utilizar a ferramenta para versionamento de código GitHub, para obter total controle das alterações realizadas no código.
- Utilizar as boas práticas da metodologia ágil FDD para padronização e desenvolvimento do projeto.
- Utilizar o a biblioteca *jQuery* para manipulação dos elementos HTML.
- O sistema será validado pelos funcionários do setor de Suporte de TI da UFN.

## **2. Referencial Teórico**

Nesta seção, serão abordados os conceitos necessários para a realização do presente trabalho, subdividido em tópicos. Iniciando pela revisão bibliográfica do tema escolhido para contextualizar a proposta do trabalho, logo após serão apresentados os conceitos sobre cada tecnologia que será utilizada para o desenvolvimento do trabalho.

### **2.1 Sistemas de informações Gerenciais (SIG)**

Para uma melhor contextualização será apresentado o conceito de sistemas de informações gerenciais com base em dois autores.

Para Kroenke (2012) um SIG consiste no desenvolvimento e uso de sistemas de informação que ajudam as empresas a alcançarem suas metas e objetivos. No seu entendimento, este conceito apresenta três elementos chave: desenvolvimento e uso, sistemas de informação e metas e objetivos de negócios.

Outra definição de SIG temos que:

“Os sistemas de informação gerencial são sistemas capazes de acessar, organizar, resumir e exibir informações para dar suporte à tomada repetitiva de decisões em áreas funcionais” [Turban, Mclean e Wetherbe 2004].

Num primeiro momento, estes sistemas, equipados para a gerência de médio escalão, apresentavam como principal característica a facilidade em produzir relatórios, como por exemplo, o controle diário do ponto dos funcionários, ou um relatório mensal de despesas e sua repercussão orçamentária. Na sua origem, os SIGs apresentavam somente uma evolução histórica do comportamento dos dados, após a ocorrência deles.

Hoje, os relatórios dos SIGs podem incluir relatórios sumários, para os mais diversos períodos [Turban, Mclean e Wetherbe, 2004].

### **2.2. Ativo**

Para melhor fundamentar a abordagem do tema proposto, será apresentada uma breve definição do que é um ativo com base em diferentes autores.

Na Contabilidade, são considerados ativos os bens e os direitos de uma entidade, expressos em moeda e à disposição da administração [Perez e Famá 2006].

Se olharmos através de uma ótica econômica e financeira, os ativos são recursos controlados pela empresa e capazes de gerar benefícios futuros (entradas de caixa ou redução de saídas de caixa). Portanto, pode ser considerado um ativo, todo e qualquer elemento com ou sem natureza física, que seja controlado pela empresa e que a ela proporcione a possibilidade de obtenção de fluxos de caixa” [Perez e Famá 2006].

A ABRAMAN (Associação brasileira de manutenção e gestão de ativos) define ativo como:

“Ativo (A) é algo que tenha valor real, ou potencial, para uma organização. A definição de valor pode variar entre diferentes tipos de organização e seus públicos de interesse. Pode ser tangível, ou intangível, financeiro, ou não financeiro. Para muitas organizações,

ativos físicos costumam referir-se a equipamentos, inventários, propriedades de posse da organização e contrapõem-se aos ativos intangíveis, não físicos, como aluguéis, marcas, ativos digitais, propriedades intelectuais, licenças de uso, reputação e acordos” [Lafraia 2013].

### 2.3. Gestão de ativos de TI

A definição para gestão de ativos para Silva e Pinto (2019) é:

“A gestão de ativos de rede compreende além do registro das informações dos dispositivos outros dados pertinentes ao seu ciclo de vida incluindo informações de aquisição e garantia. Essas informações permitem ao administrador ter uma visão estratégica do parque de computadores podendo assim prever possíveis atualizações e manutenções [Silva e Pinto 2019].

Neste contexto a gestão de ativos possui um papel importante para o administrador. Sendo este processo, portanto o primeiro a ser dado para a implantação de uma rede organizada, onde todos os dispositivos são registrados e sua utilização é acompanhada desde a aquisição até o descarte, reduzindo desperdícios e maximizando a utilização dos recursos existentes, evitando ao máximo o desperdício de recursos [Silva e Pinto 2019].

### 2.4. PHP

PHP é acrônimo recursivo para *Hypertext Preprocessor* é uma linguagem de *script* de código aberto amplamente utilizada, especialmente adequada para desenvolvimento web e pode ser incorporada em código HTML (*HyperText Markup Language*).

O PHP possui diversas vantagens por ser uma linguagem simples para iniciantes, porém oferece recursos avançados para programadores profissionais [PHP 2021].

### 2.5. jQuery

jQuery é uma biblioteca *JavaScript* pequena e com diversos recursos. Com *jQuery* torna-se mais fácil a manipulação de elementos HTML, animações, eventos e requisições *Ajax* por exemplo. Com a facilidade de utilização do *jQuery*, a biblioteca tornou-se amplamente utilizada por desenvolvedores [jQuery 2021].

### 2.6. Bootstrap

*Bootstrap* é um *framework front-end* de código aberto criado por Mark Otto e Jacob Thornton. O *framework* é uns dos mais populares pois proporciona uma grande facilidade para programadores desenvolverem sites responsivos sem a necessidade de utilizar códigos CSS (*Cascading Style Sheets*) [Logen 2021].

Além disso o *Bootstrap* possui uma gama de componentes e ícones disponíveis para utilização. Por ser uma ferramenta gratuita diversos desenvolvedores criam temas, tendo como base o *Bootstrap*, para diferentes finalidades: seja um painel administrativo ou um site de determinado produto por exemplo [Bootstrap 2021].

A utilização do *framework Bootstrap* será de grande importância para o desenvolvimento do trabalho, uma vez que possibilita a construção de interfaces

responsivas como mencionado anteriormente e com isso otimiza-se tempo no desenvolvimento [Logen 2021].

## 2.7. SQL Server

O SQL Server é um SGBD (Sistema Gerenciador de Banco de Dados) relacional, desenvolvido e mantido pela Microsoft.

O SGBD oferece uma alta confiabilidade e integridade nos dados, e um alto desempenho nas consultas. O SQL Server possui diversas versões e ferramentas para utilização [Microsoft 2021], no presente trabalho foi utilizado o SMSS (*SQL Server Management Studio*) na versão 2018.

## 2.8. GitHub

Gitub é uma plataforma onde é possível hospedar códigos em repositórios, estes repositórios podem ser públicos ou privados.

Nos repositórios públicos é possível que desenvolvedores visualizem códigos de outros desenvolvedores com a possibilidade de contribuir no projeto através de *pull requests* que são solicitações de alteração, ou resolver *issues* que são problemas já conhecidos que não possuem solução.

A plataforma mantém um histórico de alterações baseado em Git, que é uma ferramenta de versionamento de código [GitHub 2021].

## 2.9. FDD (*Feature-Driven Development*)

FDD é um processo ágil de desenvolvimento de software que inclui atividades como: modelagem, planejamento prévio e controle do projeto. Como contém características de processos ágeis, inclui: foco na programação, interação constante com o cliente e entrega frequente de versão do produto.

O FDD prevê práticas apenas para o desenvolvimento de software, sem se preocupar com outros fatores como a escolha de tecnologias e ferramentas, a definição de procedimentos de aquisição, dentre outros [Silva, Hoentsch e Silva 2009].

Embora não seja tão orientada à documentação, relatórios que controlam o estado e o progresso das atividades são previstos. Os artefatos principais são: plano de projeto, lista de funcionalidades e diagrama de sequência [Silva, Hoentsch e Silva 2009].

O plano de projeto é o documento principal de saída a ser aprovado pelo cliente, nele está definido o escopo, a lista de funcionalidades, os riscos, as métricas para controle do projeto, os critérios de aceitação, dentre outras informações pertencentes ao domínio da aplicação [Silva, Hoentsch e Silva 2009].

A processo ágil é constituída por cinco fases que são: desenvolver um modelo abrangente, construir lista de funcionalidades, planejar por funcionalidades, projetar por funcionalidades e construir por funcionalidades [Silva, Hoentsch e Silva 2009].

## 3. Trabalhos Correlatos

Nesta seção é apresentado os trabalhos correlatos que contribuíram para o desenvolvimento do trabalho. Os trabalhos foram escolhidos tendo como critério as tecnologias utilizadas e a semelhança com o tema proposto.

### **3.1. Sistema de autogestão para condomínio de pequeno porte**

O trabalho de Junior (2020) desenvolvido como trabalho final de graduação na Universidade Franciscana (UFN) propõe o desenvolvimento de um sistema web para autogestão de condomínios de pequeno porte para auxiliar o síndico a administrar sozinho as obrigações de um condomínio.

Segundo Junior (2020) ainda hoje alguns condomínios utilizam planilhas eletrônicas como alternativa de baixo custo para gerenciar o condomínio, controles manuais tornam-se ineficazes para gerir um grande volume de informação.

O sistema será desenvolvido utilizando a linguagem de programação PHP (*Hypertext Preprocessor*) em conjunto com o *Slim framework* para gerenciamento de rotas, e como sistema gerenciador de banco de dados será utilizado o MySQL.

### **3.2. Sistema Web para Gerenciamento de Laboratórios de informática**

O projeto realizado por Wagner (2007) da Universidade Tecnológica Federal do Paraná (UFFPR) tem por objetivo o desenvolvimento de um sistema web de fácil acesso aos alunos, professores e administradores da Rede Local de Ensino (RLE), para consultar horários, solicitar reservas e gerenciar os recursos fornecidos pelo sistema.

A escolha deste projeto se justifica pela necessidade de manter a integridade dos dados referente ao controle do número de patrimônio dos equipamentos. Outro problema estava relacionado aos horários de utilização do laboratório. Para um aluno saber se o laboratório estava disponível era necessário deslocar-se até o local e verificar o informativo que nem sempre estava atualizado, e para um professor efetuar uma reserva era necessário contatar pessoalmente algum administrador da Rede Local de Ensino.

O sistema foi desenvolvido em Java, utilizando o *Tomcat*, e como sistema gerenciador de banco de dados foi escolhido o PostgreSQL, por ser um software de código aberto e multiplataforma.

Após o desenvolvimento do projeto ficou notória a diferença na simplicidade de realizar determinadas tarefas da RLE que antes eram repetitivas e de difícil gerenciamento.

### **3.3. Software OCS Inventory: Inventário de ativos de TI**

*OCS Inventory* é um software que permite usuários inventariar automaticamente seus ativos de TI. Ele dispõe de uma versão gratuita e duas versões pagas. A versão gratuita de código aberto (*Open Source*), é mantida pela comunidade de desenvolvedores, e está disponível no site GitHub.

O sistema coleta informações sobre hardware e software das máquinas em rede, executando o programa cliente OCS (*OCS Inventory Agent*). Este *Agent* por sua vez fica encarregado de enviar as informações da máquina para um servidor remoto, utilizando os protocolos de comunicação HTTP ou HTTPS.

O servidor OCS recebe os inventários enviados pelos agentes em formato XML e estes dados são armazenados em um banco de dados MySQL. O OCS também possui a possibilidade de realizar instalação de aplicativos nos computadores de acordo com critérios de pesquisa.

### 3.4. Conclusão trabalhos correlatos

Os trabalhos relacionados possuem tecnologias e/ou finalidade semelhantes ao que foi desenvolvido neste projeto. Todos os projetos têm seu foco em automatizar tarefas que anteriormente eram registradas de forma manual, tendo assim uma menor eficiência que por vezes pode gerar inconsistência nos dados.

Os trabalhos estudados foram de grande valia e agregaram com conhecimento para o desenvolvimento do atual projeto, seja em tecnologia ou no tema proposto uma vez que no trabalho de Junior (2020) que utiliza tecnologias semelhantes àquelas que foram utilizadas neste projeto, e o trabalho de Wagner (2007) se assemelha bastante com o tema proposto neste trabalho, abordando a gerência de laboratórios em informática e mantendo o registro de dos equipamentos, também se tratando de ser um sistema web.

O software apresentado como correlato trata-se de um sistema para inventariar ativos de TI, por este motivo possui semelhança com o presente projeto, pois ambos mantêm informações de hardware e listam os softwares instalados nos dispositivos.

## 4. Metodologia

Para o desenvolvimento do trabalho, foram adotadas boas práticas do processo de desenvolvimento ágil FDD. Com a sua utilização é possível definir claramente cada fase que compõe o desenvolvimento do software como por exemplo o modelo abrangente, que proporciona uma visão geral do que será desenvolvido.

### 4.1. Desenvolver Modelo Abrangente

Esta fase inicial que abrange todo o projeto, ela é realizada por membros do domínio do negócio em questão e por desenvolvedores. É realizado um estudo dirigido sobre o escopo do sistema [Retamal 2008].

Na Figura 1 é apresentado um diagrama de domínio, obtendo uma visão geral do projeto.

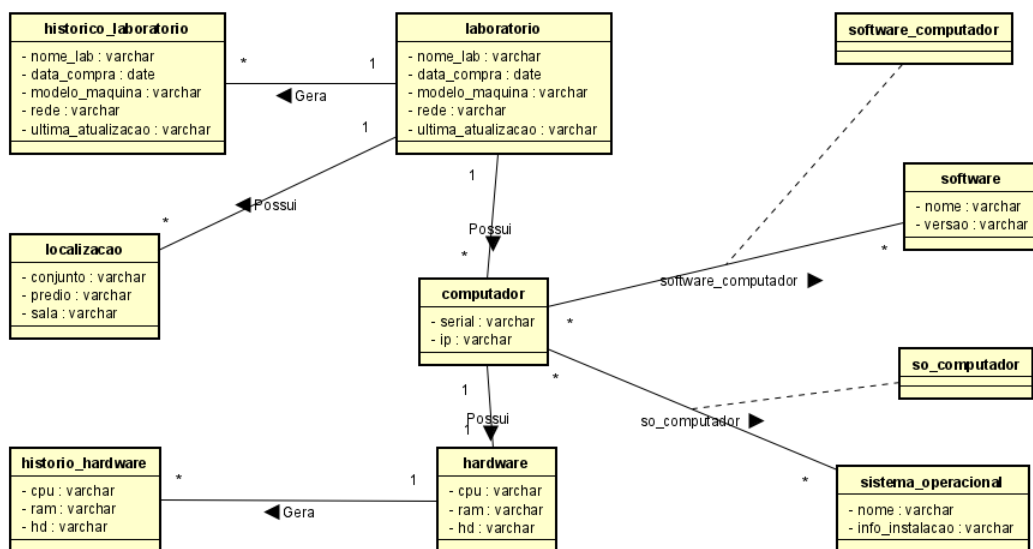


Figura 1. Diagrama de Domínio do Projeto

## 4.2. Desenvolver lista de funcionalidades

O segundo passo, conforme a FDD, é construir uma lista de funcionalidades. Esta etapa tem por objetivo identificar todas as funcionalidades do sistema.

As funcionalidades são compostas por requisitos funcionais (RF), e requisitos não funcionais (RNF) conforme listagem seguinte.

### Requisitos Funcionais

- RF1. Gerenciar Laboratórios
  - RF1.1 Gerenciar o cadastro, alteração e exclusão de laboratórios.
  - RF1.2 Vincular um ou mais laboratórios com um ou mais softwares.
  - RF1.3 Gerar histórico do laboratório após qualquer alteração nos dados.
  - RF1.4 Ao inserir um laboratório novo, sistema irá gerar automaticamente o número de máquinas inseridas.
  - RF1.5 Listagem dos laboratórios cadastrados
  - RF1.6 Download de *script* para acesso remoto das máquinas
  - RF1.7 Download de *script* para transferência de arquivos para diversas máquinas
- RF2. Gerenciar Softwares
  - RF2.1 Gerenciar o cadastro, alteração e exclusão de softwares.
  - RF2.2 Vincular um ou mais softwares com um ou mais laboratórios.
  - RF2.3 Listagem dos softwares cadastrados
- RF3. Gerenciar Localização
  - RF3.1 Gerenciamento o cadastro, alteração e exclusão da localização dos laboratórios.
- RF4. Gerar Histórico de Hardware
  - RF4.1 Gerar um histórico de hardware ao realizar alteração ou exclusão nos dados.

### Requisitos Não Funcionais

- RNF1. Fazer uso da linguagem de programação PHP;
- RNF2. Fazer uso do *framework Bootstrap*;
- RNF3. Fazer uso do *template Falcon – Admin Dashboard & WebApp*;
- RNF4. Fazer uso do Sistema Gerenciador de Banco de Dados SQL Server;
- RNF5. Fazer uso do GitHub para versionamento de código do projeto.



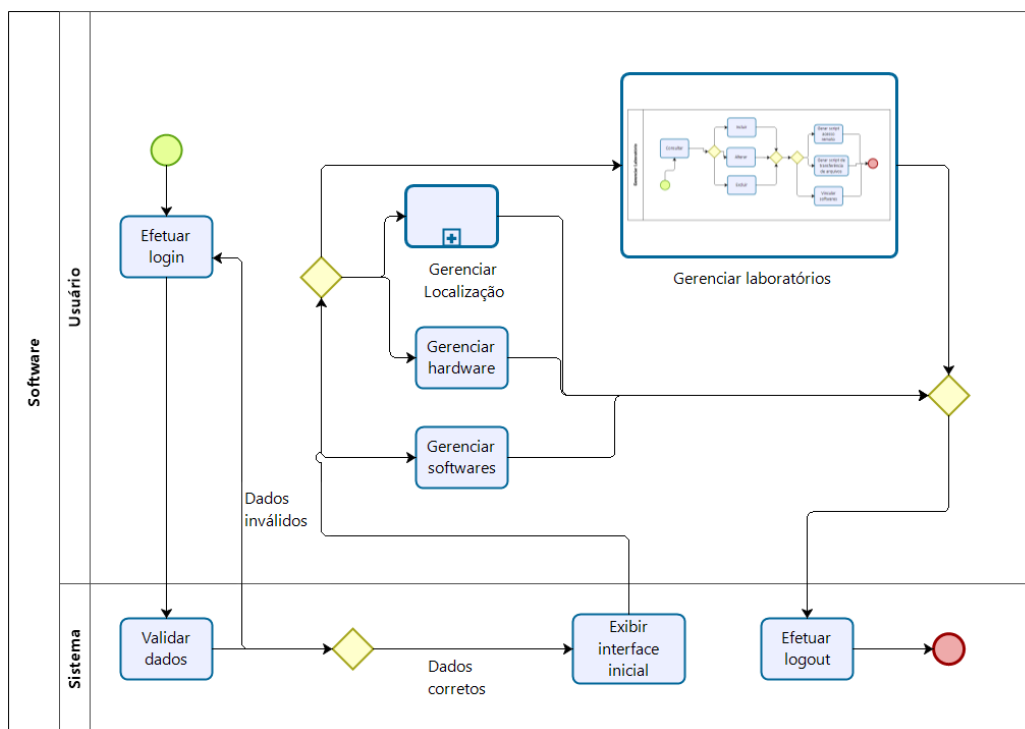
### 4.3. Planejamento por funcionalidades

Nesta etapa do processo, os gerentes de desenvolvimento e programadores planejam a ordem das funcionalidades que serão implementadas, baseados nas dependências que possuem [Retamal 2008].

Para estimar o tempo de projeto foi desenvolvida a Tabela 1 contendo informações sobre cada funcionalidade a ser implementada juntamente com o tempo previsto a para o desenvolvimento.

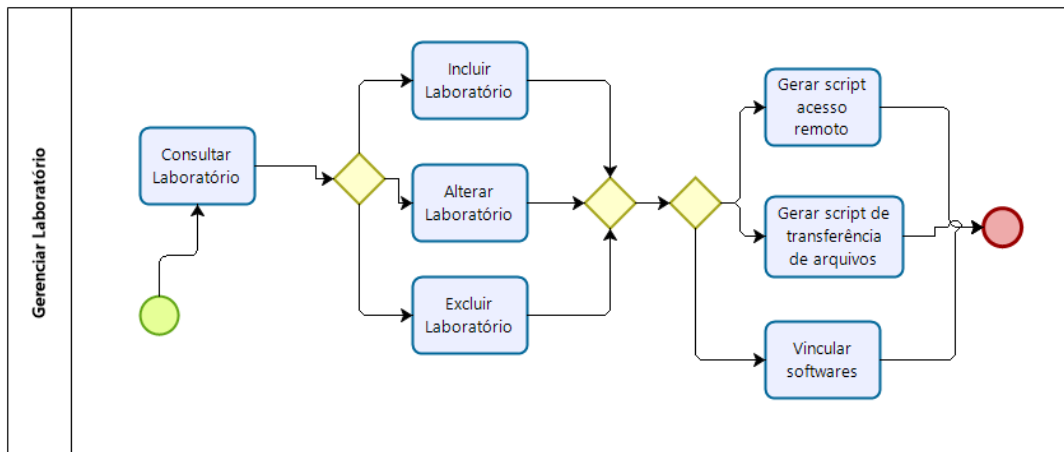
A Figura 2 apresenta o diagrama de Atividades do sistema contendo as atividades presentes no sistema.

Após o usuário realizar login no sistema, é exibido a interface contendo as opções de gerenciar softwares, localização, hardware e laboratórios.



**Figura 2. Diagrama de Atividades do projeto**

A Figura 3 apresenta o subprocesso Gerenciar Laboratório, esta é principal atividade do sistema, nela é possível visualizar os laboratórios cadastrados no sistema e suas informações.

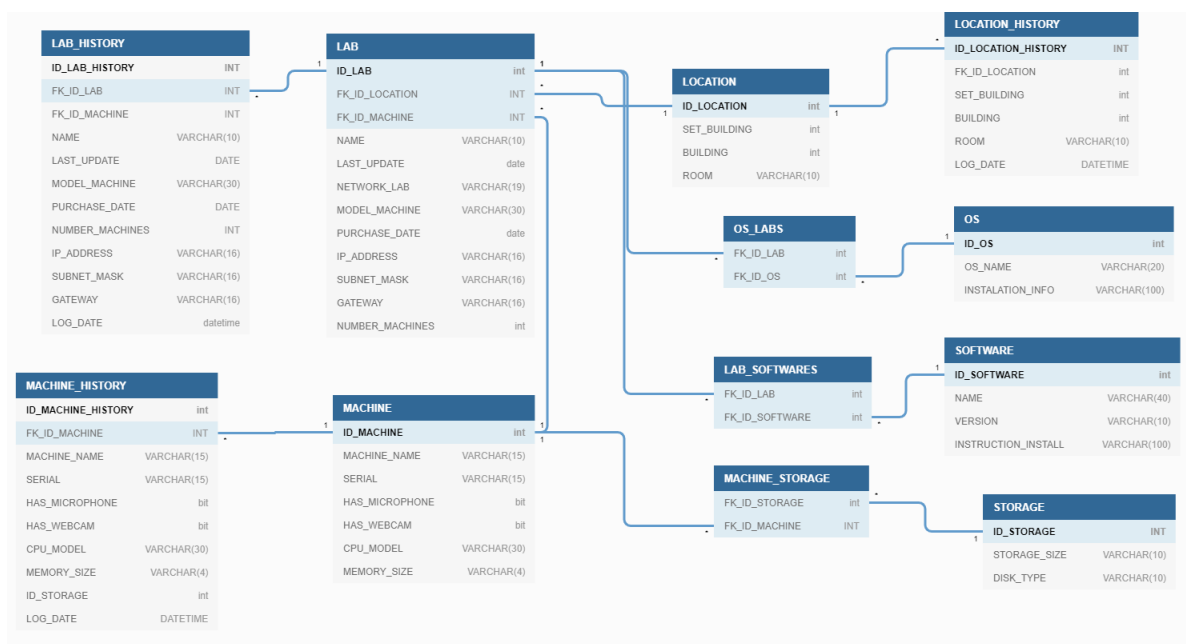


**Figura 3. Subprocesso Gerenciar Laboratório**

#### 4.4. Projetar por funcionalidade

Nesta etapa de projetar por funcionalidade, para cada funcionalidade prevista no sistema é definida uma atividade a ser realizada, também o modelo de interface do usuário é esboçado e os diagramas de sequência e classe são desenvolvidos [Silva, Hoentsch e Silva 2009].

A Figura 4 apresenta o Diagrama Entidade Relacionamento (DER) do sistema, contendo a estrutura de tabelas do banco de dados. É possível visualizar na figura a estrutura de armazenamento dos dados das máquinas contidas em cada laboratório bem os respectivos softwares instalados, a configuração de hardware de cada máquina e sua devida localização.



**Figura 4. Diagrama de Entidade Relacionamento**

A Figura 5 apresenta o diagrama de classes do sistema, onde consta as classes do projeto juntamente com os atributos e métodos do sistema.

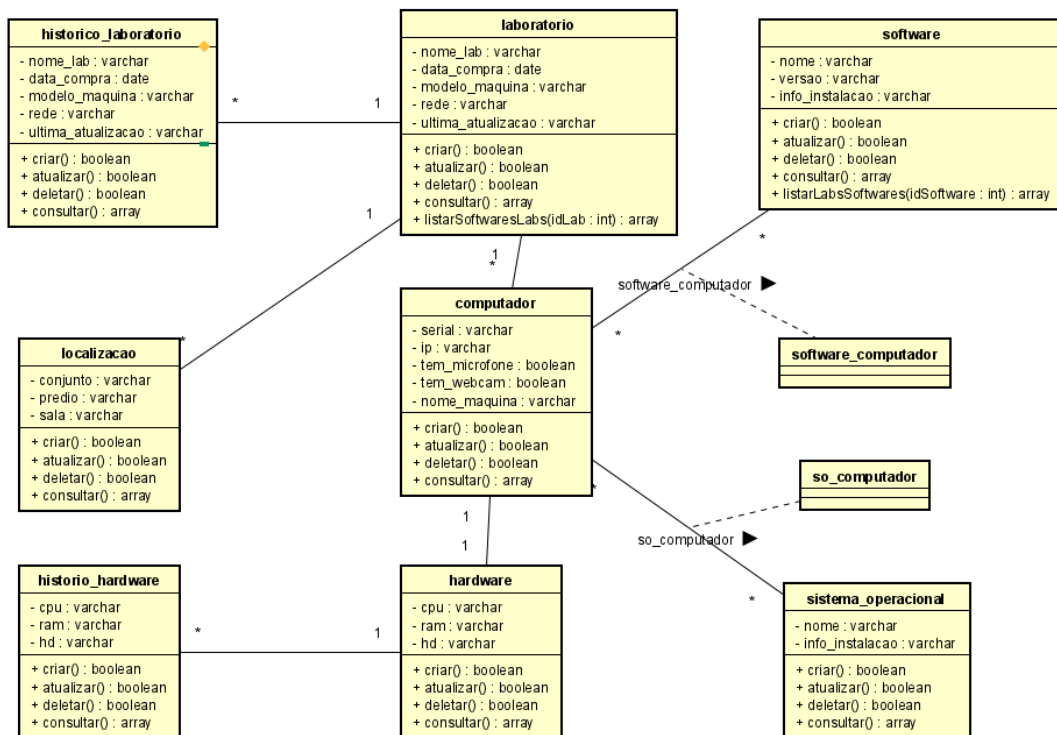


Figura 5. Diagrama de Classes

## 5. Implementação

Esta seção apresenta os trechos de códigos desenvolvidos durante a etapa 5 do FDD.

A Figura 6 apresenta a captura do evento de submissão de um formulário de cadastro de software utilizando *jQuery*, onde é utilizado um método para buscar os valores dos campos do formulário, após isso é realizada uma requisição HTTP do tipo POST para o servidor, passando os dados de criação do software, quando o servidor retornar a resposta é apresentado na interface o status da requisição.

```

$('#form-create').submit((event) => {

    event.preventDefault();

    let valuesForm = GetValuesForm($('#form-create :input'));

    $.post("/labsManagement/controllers/software/createSoftware.php", {
        "softwareName": valuesForm.softwareName,
        "version": valuesForm.version,
        "instructionInstall": valuesForm.instructionInstall
    })
    .done(function (data) {
        if (data == 1) {
            toastr.success("Software Cadastrado!");
            $('#form-create').trigger("reset");
        } else if (data == 2) {
            toastr.warning("Software já cadastrado!");
        } else {
            toastr.error("Erro ao Cadastrar!");
        }
    })
    .catch((err) => {
        console.log(err);
        toastr.error("Erro ao Cadastrar!");
    });
});

```

**Figura 6. Captura do evento de submissão de um formulário de cadastro de software**

A Figura 7 apresenta a classe de *Location*, contendo seus atributos, método construtor e o método para realizar a inserção no banco de dados e retornando o identificador que foi gerado.

```

class Location
{
    var $setBuilding;
    var $building;
    var $room;

    function Location($setBuilding, $building, $room)
    {
        $this->setBuilding = $setBuilding;
        $this->building = $building;
        $this->room = $room;
    }

    public static function Create($location)
    {
        $query = "
            INSERT INTO [dbo].[LOCATION]([SET_BUILDING],[BUILDING],[ROOM])
            VALUES ($location->setBuilding,$location->building,'$location->room')";

        try {

            $pdo = Database::getConnection();
            $smt = $pdo->prepare($query);

            if ($smt->execute()) {

                return $pdo->lastInsertId();
            };
        } catch (Exception $e) {

            return false;
        }
    }
}

```

**Figura 7. Classe *Location* junto com o método de inserir nova localização**

## 6. Resultados Obtidos

Como resultado do presente trabalho, um sistema web foi desenvolvido com foco no gerenciamento de ativos em laboratórios de informática. O sistema apresenta interfaces simples e intuitivas para utilização dos usuários.

O software foi disponibilizado para o setor de Suporte de TI da instituição para realização de testes, nos quais obtiveram-se resultados satisfatórios, atendendo as demandas pretendidas para o setor.

A Figura 8 apresenta a interface que contém todas as informações de um laboratório como localização, configuração de *hardware*, quantidade de softwares instalados entre outros.

Informações Gerais

### Laboratório 12

📍 Conjunto 4, Prédio 14, Sala 521  
 📅 Data de compra: 19/10/2021  
 🖼️ Última imagem: 09/11/2021  
 📜 Histórico de Atualizações  
 🌐 Ip inicial: 10.103.0.50 Máscara de rede: 255.255.255.0 Gateway: 10.103.0.1

### Configuração de Hardware

🖨️ HP Omni pro 200  
 🧠 Intel i5 2.40GHz  
 📀 4 GB  
 💾 HD 500 GB

🐧 Linux Ubuntu 12.04  
 📶 Sim  
 📷 Sim

### Software Instalados

Número de softwares instalados: 4 + Incluir novo(s) software(s)

Nome Software	Versão
Figma	
Kaspersky	2021
Krita	11
Microsoft Office	2019

### Computadores

32 computadores

Nome máquina	IP	Máscara de rede	Gateway
LAB12DT01	10.103.0.50	255.255.255.0	10.103.0.1
LAB12DT02	10.103.0.51	255.255.255.0	10.103.0.1

**Figura 8. Interface apresentando informações gerais de um laboratório.**

Na Figura 9 é apresentada a interface contendo a tabela de softwares cadastrados (ao fundo), e em destaque na Figura é apresentado um modal com informações de um software, contendo informações adicionais e quais laboratórios este software está instalado.

Software

Lista de todos os softwares

+ Novo Software

Registros por página: 15

Nome	Ações
Adobe Illustrator	...
Adobe Photoshop	...
Adobe Premiere	...
Arduino IDE	...
Corel draw	...
Microsoft Office	...
NetBeans	8.2 ...
Solidworks	2019 ...

**Adobe Illustrator 2021**

Informações Adicionais

Licença do software está disponível no servidor de arquivos

Laboratórios que possuem o software instalado

- LABORATÓRIO 23 Conjunto 3,Prédio 17, Sala 23A
- LABORATÓRIO 23 Conjunto 3,Prédio 17, Sala 23A

Fechar

**Figura 9. Interface apresentando os softwares cadastrados em um determinado laboratório.**

## 7. Conclusão

O presente trabalho possibilitou o desenvolvimento de um sistema web para gerenciamento de laboratórios de informática da Universidade Franciscana, visando a melhoria no controle dos softwares instalados, manutenção dos laboratórios e inventário dos computadores.

Com base nos requisitos funcionais desenvolvidos ao longo do trabalho, o software cumpre com os objetivos, entregando uma aplicação capaz de cadastrar os laboratórios da instituição, manter o controle dos softwares instalados e visualizar o histórico de alteração de cada laboratório entre outras funcionalidades descritas nos requisitos funcionais.

Quanto à metodologia, foram utilizadas boas práticas do FDD, por se tratar de um processo ágil e que se adequa à proposta do trabalho desenvolvido.

Os trabalhos correlatos foram de grande valia para o desenvolvimento do projeto, uma vez que serviram como apoio para escolher a ferramenta adequada e obter um maior entendimento sobre o tema.

E, por fim, para desenvolvimento de trabalhos futuros, sugere-se a implementação de um sistema de agendamento, possibilitando aos professores da instituição o agendamento de horários para a utilização dos laboratórios.

## Referências

- Bootstrap (2021). “Build fast, responsive sites with Bootstrap”. Disponível em: <https://getbootstrap.com/>, Acesso: março.
- GitHub (2021). “GitHub”. Disponível em: <https://github.com/>. Acesso: abril.
- Gomes, Pedro César Tebaldi (2019).” O que é a gestão de ativos de TI e como ela pode te ajudar?”. Disponível em: <https://cutt.ly/HbEaCAv>, Acesso: maio
- Junior, Luiz Augusto Marques Paz (2020). “Sistema de Autogestão para Condomínio de Pequeno Porte”. Trabalho Final de Graduação I – Universidade Franciscana (UFN). Santa Maria – RS
- jQuery (2021). “jQuery write less, do more”. Disponível em: <https://jquery.com/>, Acesso: outubro
- Kroenke, David (2012.) “Sistemas de informação gerenciais”. São Paulo: Saraiva.
- Lafraia, João Ricardo (2013). “GESTÃO DE ATIVOS E PAS 55”. Disponível em: [https://abramanoficial.org.br/page/gestao\\_de\\_ativos#:~:text=Ativo%20\(A\)%20%C3%A9%20algo%20que,%20financeiro%20ou%20n%C3%A3o%20financeiro](https://abramanoficial.org.br/page/gestao_de_ativos#:~:text=Ativo%20(A)%20%C3%A9%20algo%20que,%20financeiro%20ou%20n%C3%A3o%20financeiro), Acesso: abril.
- Logen, Andrei. (2021). “O Que é Bootstrap? Guia para Iniciantes”. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-bootstrap/>, Acesso: março
- Microsoft (2021). “Introdução ao SQL Server 2019”. Disponível em: <https://www.microsoft.com/pt-br/sql-server/sql-server-2019>, Acesso: abril.

- OCS Inventory (2021a). "Newbie documentation - What you should know". Disponível em: <http://wiki.ocsinventory-ng.org/02.Newbie-documentation/OCS-Inventory-NG-Basics/> Acesso: março.
- OCS Inventory (2021b). "About OCS Inventory". Disponível em: <https://ocsinventory-ng.org/?lang=en>. Acesso: março
- Perez, Marcelo Monteiro; Famá Rubens. (2006). "Ativos intangíveis e o desempenho empresarial". Revista Contabilidade & Finanças. VOL. 17, NUM. 40. Disponível em: [http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1519-70772006000100002](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1519-70772006000100002). Acesso em abril.
- Pinto, Jefferson de Souza; Silva, Sérgio Francisco (2019). "Análise da importância da gestão de ativos de TI no ambiente micro e pequenas empresas". Disponível em: <http://periodicos.faex.edu.br/index.php/e-Locucacao/article/view/181/149>, Acesso: abril.
- PHP. (2021). "Manual do PHP". Disponível em: <https://www.php.net/manual/en/intro-what-is.php>, Acesso: março.
- Retamal, Adail Muniz (2008). "Feature-Driven Development". Disponível em: <https://silo.tips/download/feature-driven-development>, Acesso: abril.
- Silva, F. G; Hoentsch, S. C. P; Silva, L. (2009). "Uma análise das metodologias ágeis fdd e scrum sob a perspectiva do modelo de qualidade mps.br". Disponível em: <https://www.scienciaplena.org.br/sp/article/view/678/348>, Acesso: abril.
- Schultz, Felix (2018). "Inventário de TI: uma visão geral sobre aplicação e uso nas empresas". Disponível em: <https://milvus.com.br/inventario-de-ti-uma-visao-geral-sobre-aplicacao-e-uso-nas-empresas/>, Acesso: maio
- Turban, Efrain; Mclean, Ephraim; Wetherbe, James. (2004) "Tecnologia da informação para gestão". 3. ed. Porto Alegre: Bookman.
- Wagner, Franklin Felipe (2007). "Sistema Web para Gerenciamento de Laboratórios de informática". Disponível em: <https://pessoal.dainf.ct.utfpr.edu.br/criskochem/SistemaWebRLE.pdf>, Acesso: abril.



## 8. APÊNDICES

### APÊNDICE A – Planejamento por funcionalidade

**Tabela 1. Planejamento por funcionalidade**

<b>Ordem</b>	<b>Funcionalidade</b>	<b>Tempo Previsto</b>
<b>RF01</b>	<b>Gerenciar Laboratórios</b>	<b>100 Horas</b>
RF1.1	Gerenciar o cadastro, alteração e exclusão de laboratórios.	15 Horas
RF1.2	Vincular um ou mais laboratórios com um ou mais softwares.	15 Horas
RF1.3	Gerar histórico do laboratório após qualquer alteração nos dados.	15 Horas
RF1.4	Ao inserir um laboratório novo, sistema irá gerar automaticamente o número de máquinas inseridas.	25 Horas
RF1.5	Listagem dos laboratórios cadastrados	10 Horas
RF1.6	Download de arquivos para acesso remoto às máquinas	10 Horas
RF1.7	Download de script para transferência de arquivos para diversas máquinas	10 Horas
<b>RF02</b>	<b>Gerenciar Softwares</b>	<b>30 Horas</b>
RF2.1	Gerenciar o cadastro, alteração e exclusão de softwares.	15 Horas
RF2.2	Vincular um ou mais softwares com um ou mais laboratórios.	10 Horas
RF2.3	Listagem dos softwares cadastrados	5 Horas
<b>RF03</b>	<b>Gerenciar Localização</b>	<b>10 Horas</b>
RF3.1	Gerenciamento o cadastro, alteração e exclusão da localização dos laboratórios	10 Horas
<b>RF04</b>	<b>Gerenciar Hardware</b>	<b>10 Horas</b>
RF4.1	Gerar um histórico de hardware ao realizar alteração ou exclusão nos dados.	10 Horas
<b>Total</b>		<b>150 Horas</b>

## APÊNDICE B – Modelagem de caso de uso

Na Figura 10 é apresentado um diagrama de caso de uso que tem por objetivo apresentar as funcionalidades que o usuário pode realizar no sistema.

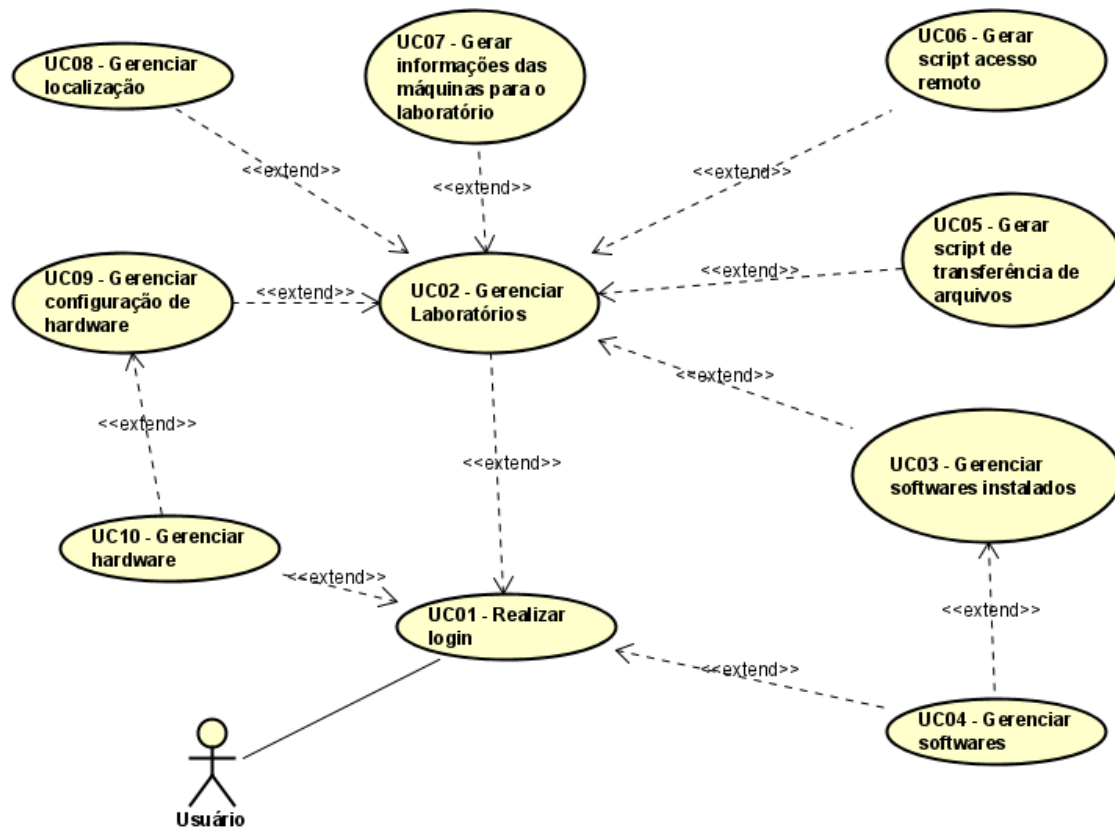


Figura 10. Diagrama de Caso de Uso do projeto

A Tabela 2 exibe o Descritivo de Caso de Uso Gerenciar Laboratórios sendo este o principal caso de uso do sistema.

Tabela 2. Descritivo de caso de uso Gerenciar Laboratórios

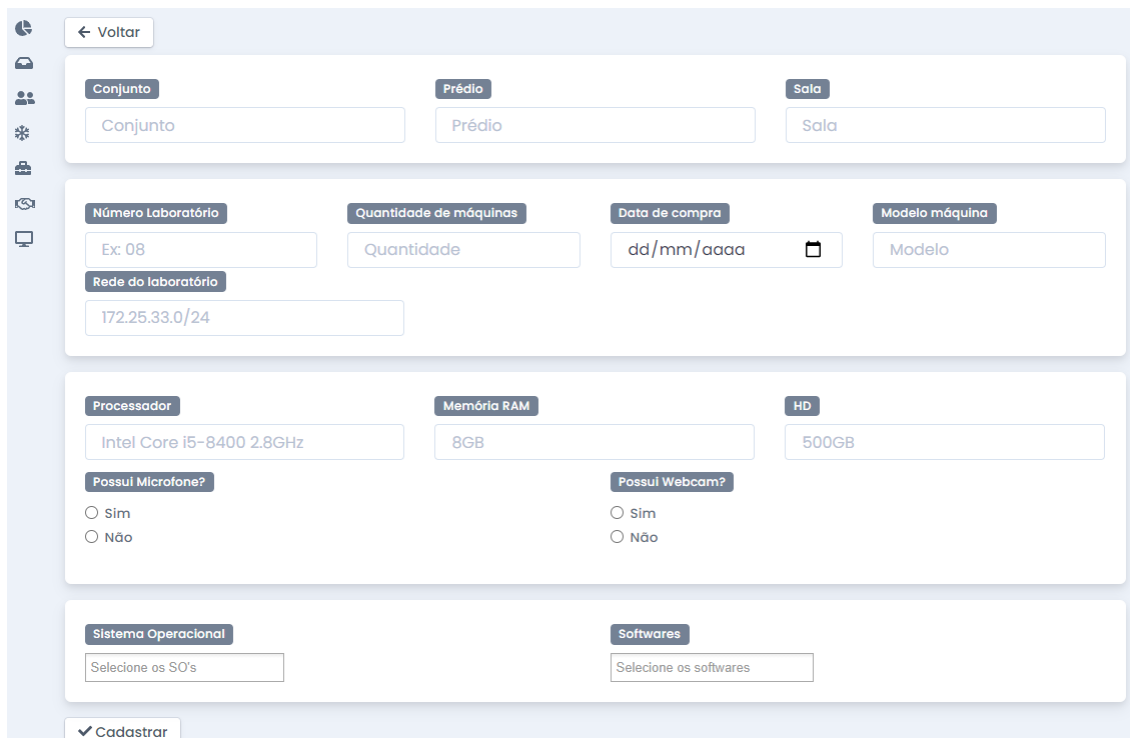
<b>Identificação</b>	UC02
<b>Caso de Uso</b>	Gerenciar Laboratórios
<b>Descrição</b>	Caso de uso responsável pela inclusão, alteração, exclusão, consulta dos laboratórios cadastrados no sistema, download de arquivos e vincular softwares instalados em cada laboratório.
<b>Ator Principal</b>	Usuário
<b>Ator(es) Secundário(s)</b>	N/A
<b>Pré Condições</b>	

<b>PC1</b>	Usuário logado no sistema [UC01]
<b>Pós Condições</b>	
<b>PO1</b>	Laboratório criado, alterado, excluído ou consultado com sucesso
<b>PO2</b>	Download de scripts concluído
<b>PO3</b>	Softwares vinculados com sucesso
<b>Fluxo Principal</b>	
<b>FP1</b>	O sistema lista os laboratórios cadastrados e exibe as opções
<b>FP2</b>	O usuário seleciona uma das opções: <ul style="list-style-type: none"> <li>• Cadastrar Laboratório [FA1],</li> <li>• Alterar Laboratório [FA2],</li> <li>• Excluir Laboratório [FA3],</li> <li>• Download script para acesso remoto [FA4],</li> <li>• Download script transferência de arquivos [FA5],</li> <li>• vincular softwares com laboratório [FA6],</li> </ul>
<b>Fluxo Alternativo</b>	
<b>FA1</b>	Cadastrar Laboratório <ul style="list-style-type: none"> <li>• FA1.1 – O usuário preenche os dados para cadastrar um novo laboratório</li> <li>• FA1.2 – O sistema valida os dados preenchidos</li> <li>• FA1.3 – O sistema grava os dados</li> <li>• FA1.4 - Gerar informações das máquinas para o laboratório [UC07]</li> <li>• FA1.5 – O sistema retorna ao [FP1].</li> </ul>
<b>FA2</b>	Alterar Laboratório <ul style="list-style-type: none"> <li>• FA2.1 – O usuário seleciona um laboratório que deseja realizar alguma alteração</li> <li>• FA2.2 – O usuário preenche as novas informações</li> <li>• FA2.3 – O usuário salva os dados alterados.</li> <li>• FA2.4 – O sistema grava os dados</li> <li>• FA2.5 – O sistema retorna ao [FP1]</li> </ul>
<b>FA3</b>	Excluir Laboratório <ul style="list-style-type: none"> <li>• FA3.1 – O usuário seleciona o laboratório que deseja excluir</li> <li>• FA3.2 – O sistema solicita a confirmação da exclusão</li> <li>• FA3.3 – O usuário confirma a exclusão</li> <li>• FA3.4 – O sistema exclui o laboratório</li> <li>• FA3.5 – O sistema retorna ao [FP1].</li> </ul>
<b>FA4</b>	Download script acesso remoto

	<ul style="list-style-type: none"> <li>• FA4.1 – O usuário seleciona uma ou mais máquinas de um laboratório no qual deseja acessar remotamente.</li> <li>• FA4.2 – O sistema gera o arquivo para download</li> <li>• FA4.3 – O sistema retorna ao [FP1]</li> </ul>
<b>FA5</b>	<p>Download script transferência de arquivos</p> <ul style="list-style-type: none"> <li>• FA5.1 – O usuário preenche os campos necessários.</li> <li>• FA5.2 – O sistema gera o arquivo para download</li> <li>• FA5.3 – O sistema retorna ao [FP1]</li> </ul>
<b>FA6</b>	<p>Vincular softwares</p> <ul style="list-style-type: none"> <li>• FA6.1 – O usuário seleciona os softwares que o laboratório possui instalado</li> <li>• FA6.2 – O usuário confirma os softwares vinculados</li> <li>• FA6.3 – O sistema registra os softwares</li> <li>• FA6.4 – O sistema retorna ao [FP1]</li> </ul>
<b>Fluxo de Exceções</b>	
<b>FE1</b>	N/A
<b>Regras de Negócio</b>	N/A
<b>Casos de Uso Estendidos</b>	UC03, UC05, UC06, UC07
<b>Casos de Uso Incluídos</b>	N/A
<b>Observações</b>	N/A

## APÊNDICE C – Protótipos de Interface

A Figura 11 apresenta a interface de cadastro de laboratório, onde serão inseridos todos os dados referentes ao laboratório que deseja incluir no sistema.



Protótipo de interface de cadastro de laboratório. A interface é organizada em seções horizontais com campos de entrada e botões de ação.

- Seção 1:** Campos para "Conjunto", "Prédio" e "Sala".
- Seção 2:** Campos para "Número Laboratório" (ex: 08), "Quantidade de máquinas", "Data de compra" (formato dd/mm/aaaa) e "Modelo máquina".
- Seção 3:** Campo para "Rede do laboratório" (ex: 172.25.33.0/24).
- Seção 4:** Campos para "Processador" (ex: Intel Core i5-8400 2.8GHz), "Memória RAM" (ex: 8GB) e "HD" (ex: 500GB).
- Seção 5:** Campos de seleção para "Possui Microfone?" e "Possui Webcam?", com opções "Sim" e "Não".
- Seção 6:** Campos para "Sistema Operacional" (Selecione os SO's) e "Softwares" (Selecione os softwares).
- Botão:** "Cadastrar" (com ícone de checkmark).

Figura 11. Protótipo de interface do cadastro de laboratório

A Figura 12, apresenta a interface contendo os laboratórios cadastrados no sistema, apresentando a localização e o modelo de máquina, com a possibilidade de incluir um novo laboratório, editar, excluir ou visualizar todas as informações do laboratório.



Protótipo de interface de gerenciamento de laboratórios. A interface apresenta um menu lateral, um cabeçalho de seção e uma tabela de dados.

**Menu Lateral:** Home, Notificações, Visitantes da rede, Climatização, Ferramentas, Controle de empréstimo, Laboratórios.

**Cabeçalho de Seção:** Laboratórios de informática, Gerenciamento dos laboratórios de informática da Universidade Franciscana.

**Botão de Ação:** + Novo Laboratório

**Tabela de Laboratórios:**

Laboratório	Localização	Modelo máquina	Quantidade	
01	Prédio 16, Sala 611	DELL OPTIPLEX 5260	24	...
02	Prédio 14, Sala 716B	HP OMNI PRO 110	21	...
03	Prédio 13, Sala 207	DELL OPTIPLEX 3030	40	...

**Menu de Ação (para o registro 01):** Informações, Editar, Excluir.

**Controles de Página:** Registros por página: 15, Buscar: [campo de texto], < 1 >

Figura 12. Protótipo de interface do gerenciamento de laboratórios

A Figura 13 apresenta a interface contendo todas as informações do laboratório que o usuário seleciona na Figura 12.

The interface displays the following information for 'Laboratório 01' (Conjunto 3, Prédio 16, Sala 611):

- Configuração de Hardware:**
  - Processador: Intel Core i5 4590S
  - Memória RAM: 8GB
  - Armazenamento: SSD 240GB
  - Sistema Operacional: Windows 10 Education
- Softwares Instalados:**
  - Adobe Premiere 2021
  - Corel Draw 2018
  - Solidworks 2019
  - Office 2019
  - Adobe Photoshop 2021
  - Adobe InDesign 2021
- Computadores:** A table listing five machines with checkboxes for selection and a remote access button.

	Nome máquina	IP
<input checked="" type="checkbox"/>	LAB01DT01SMA01	172.25.33.02
<input checked="" type="checkbox"/>	LAB01DT02SMA01	172.25.33.03
<input type="checkbox"/>	LAB01DT03SMA01	172.25.33.04
<input type="checkbox"/>	LAB01DT04SMA01	172.25.33.05

Figura 13. Protótipo de interface que mostra as informações do laboratório