**Sistema de Inventário e Empréstimos de Computadores e Periféricos**

¹**Michael Pablo Gomes da Silva**

²**Daiane Mastrangelo Tomazeti**

Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas  
Instituto Federal de São Paulo – Campus Hortolândia (IFSP)

¹ michael.silva@aluno.ifsp.edu.br, ²daianetomazeti@ifsp.edu.br

***Abstract.*** *The routines for the assets management for the observed company are made manually, by spreadsheets, that are laborious and exhausting, no mention the risks that affects the analysis, using too many filters that might culminate in mistakes in the calculation of informatics assets price, the control of Desktops and Laptops lending for the user in Company X and the management of informatics peripherals for lending. Beginning with this scenario, a software will be build, based in a Web Platform with the use of the good practices of Software Engineering.*

***Resumo.*** As rotinas para administração dos ativos de informática da Empresa observada neste projeto são feitas manualmente, por meio de planilhas eletrônicas, que são laboriosas e desgastantes, sem contar os riscos que afetam a análise, utilizando muitos filtros que podem culminar em erros de cálculo de preço de ativos de informática, o controle de empréstimo de computadores ou computadores portáteis para os usuários da Empresa observada e também administração de periféricos de informática para empréstimo. A partir deste cenário, será construído um software, baseado na plataforma Web com a utilização das boas práticas de Engenharia de Software.

1. **Introdução**

Os desafios organizacionais dos dias atuais, onde o ambiente é volátil, incerto, complexo e ambíguo, obriga as empresas a serem mais rápidas em respostas do mercado e para isso elas precisam de organização, melhoria contínua e foco nas necessidades do cliente, interno ou externo, para obtenção de resultados condizentes com o cenário real.

O sistema de informação que pode ser definido tecnicamente como “um conjunto de componentes inter-relacionados que coletam (ou recuperam), processam, armazenam e distribuem informações destinadas a apoiar a tomada de decisões, a coordenação e ao controle de uma organização.” (LAUDON, 2010, p.12). Com esta finalidade, nos últimos anos, os sistemas de informação tiveram um papel importante no desenvolvimento de muitas empresas, seja os sistemas em nível operacional, tático ou gerencial.

Para este projeto, o sistema controlará o empréstimo de computadores e periféricos. E o resultado esperado beneficiará o controle não somente de computadores e periféricos como também o controle de custos por meio da quantidade destes equipamentos e o melhor aproveitamento do parque de máquinas da Empresa observada.

O desenvolvimento de sistemas demanda a compreensão de um problema ou processo a se modelar e acima de tudo ter a perspectiva do cliente. “Modelagem de sistema é o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema.” (SOMMERVILLE, 2011, p.82). A modelagem deste sistema pretende, portanto, seguir premissas dos processos de negócios da Empresa observada.

Com isso o objetivo deste projeto é controlar o parque de máquinas e periféricos de Tecnologia da Informação (TI) e facilitar o controle de máquinas e periféricos de TI e o empréstimo de computadores ao usuário.

Com a conclusão do sistema proposto, ele estará alinhado com o plano de negócios do Departamento de Tecnologia da Informação, que visa a melhoria contínua dos processos.

1. **Justificativa**

Devido à grande demanda de empréstimos de laptops por usuários e o controle ser realizado por uma planilha eletrônica e também pela grande rotatividade de máquinas entre os indivíduos da Empresa observada, vimos a necessidade de fazer um sistema de informação que ajude no controle do processo de empréstimo de computadores e periféricos.

A planilha eletrônica, é uma planilha usada pelo Departamento de Informática da Empresa observada para controlar o empréstimo de computadores. Nela, são encontrados campos de identificação do computador, nome do usuário e principalmente a data de empréstimo em conjunto com a data de devolução dos computadores.

A necessidade da utilização da planilha eletrônica surgiu, da demanda do controle de empréstimo de computadores reservas para usuários e informalmente para pessoas de divisões distintas dentro da Empresa observada. Assim, a planilha eletrônica serviria como controle para os dados como: data de empréstimo e devolução, responsável, usuário, razão do empréstimo. O Analista do departamento de informática, teria maior controle das máquinas reservas que o Departamento de Informática possui no momento. A segunda necessidade que surgiu após o controle desse empréstimo, foi a de controle das máquinas que estão em manutenção e quando elas voltariam, isso seria estimado pelo analista junto a equipe interna que daria manutenção a estas máquinas que por sua vez, alertaria o usuário sobre o retorno de seu computador da manutenção.

O grande problema deste processo são os acordos informais feitos com os usuários que são registrados na planilha eletrônica, que torna o trabalho exaustivo e repetitivo. Outro gargalo observado nesse processo, são as máquinas reserva que não estão sendo utilizadas, por estarem em manutenção com a equipe de TI da Empresa observada ou pelo fabricante. As manutenções podem ser realizadas dentro de um dia pela equipe local, se não houver solução, a máquina é enviada para o fabricante que poderá demorar até um mês para realizar a manutenção.

Em alguns casos, o fabricante não fará a manutenção do computador através da garantia por mal-uso da máquina pelo Usuário, portanto, isso também deverá estar registrado na planilha eletrônica com uma observação e a compra de outro equipamento deverá ser feita para substituição dessa máquina. Outro fator relevante, é quando os usuários são roubados por trabalharem em diversas localidades como funcionários externos, nestes casos o Departamento de Informática deverá ser informado do roubo para baixa desta máquina no sistema.

Na grande parte do tempo, as situações citadas anteriormente, impedem a manutenção da utilização do inventário de computadores que é feita manualmente. Como muitos casos ocorrem, e os Analistas não reportam as informações na planilha de inventário, que é utilizado como base de dados, fica comprometido o controle das informações por não condizerem com a realidade.

Do ponto de vista dos usuários, será interessante ter algum sistema para solicitação e reserva de equipamentos. Do ponto de vista do Analista, é interessante ter um controle melhor do seu parque de máquinas para apresentar painéis com gráficos para a gerência informando dados sobre o status dos computadores.

Com os cenários apresentados, a razão para criação deste Sistema Web, seria dar maior controle aos Analistas de TI sob o parque de máquinas da Empresa estudada e atender as necessidades dos usuários e da gerência para tomar decisões com relação as máquinas com problemas constantes. Com isso, as informações para os interessados estarão disponibilizadas através deste Sistema Web.

1. **Objetivos**

**3.1. Objetivos Gerais**

O objetivo principal deste projeto será desenvolver uma ferramenta de controle do parque de maquinas e periféricos da Empresa estudada. Esta ferramenta ajudara os Analistas de TI da Empresa a controlar e mostrar o status das maquinas para gerentes demonstrando informações a nível gerencial e também para os usuários, que tem demandas de periféricos e laptops reservas.

**3.2. Objetivos Específicos**

O Objetivo específico será facilitar controle de maquinas e periféricos de TI e também, empréstimo de computadores ao usuário.

Para o desenvolvimento de um software e necessário ter o entendimento completo dos processos envolvidos. De acordo com (PRESSMAN, 2011), para alcançar o objetivo esperado do Software, é importante que o desenvolvimento seja tratado como um processo de engenharia visando a qualidade e a segurança das informações.

“Quando um processo tem suas atividades bem definidas, pode se criar métodos de Engenharia de *Software* que permitam avaliar o progresso dos processos e corrigir seus rumos quando ocorrerem problemas. Esses métodos de Engenharia subdividem os processos. Essas subdivisões são chamadas de fases. As subdivisões devem ser terminadas com marcos, isto é, pontos que representam estados significativos do projeto.” (LIMA et al., 2012).

O projeto se baseara em conceitos de engenharia de software, qualidade de software e boas práticas de desenvolvimento de software.

Este projeto de software se destaca de outros softwares relacionados a inventario porque o desenho do software foi feito para a Empresa estudada. Assim, vários conceitos aprendidos em sala de aula serão utilizados na construção deste sistema.

**4. Fundamentação Teórica**

**4.1 Engenharia de *Software***

Sommerville (2013, p.4) define engenharia de software como uma disciplina que se preocupa com todos os aspectos de produção de software. Em relação ao desenvolvimento de aplicações Web, espera-se que “leve em consideração algumas ou todas as lições aprendidas durante as muitas décadas de desenvolvimento de Software convencional” (BEDER, 2011, p. 16),

Engenharia de Software, e uma disciplina, porque carrega ferramentas de controle e documentação de um projeto de engenharia de software. Neste projeto, não será diferente, serão utilizadas boas práticas de desenvolvimento de software para a construção do software como qualidade de software, teste de software e gerenciamento de projetos.

**4.2 Aplicação WEB**

O desenvolvimento de aplicações Web vem evoluindo constantemente e está se integrando rapidamente nas estratégias de negócio das organizações (2008 *apud* BEDER, 2011, p. 19)

Aplicações web são aplicações que são acessadas através de um navegador substituindo sistemas instalados nos desktops. As aplicações web são acessadas através de um sistema cliente-servidor, onde o servidor terá toda a aplicação junto com toda a base de dados e com um navegador. Isso ocorre devido ao protocolo HTTP para comunicação entre cliente e servidor, onde os clientes enviam requisições aos servidores e os servidores enviam respostas ao usuário.

O ponto forte da aplicação web e a acessibilidade, onde os usuários do sistema poderão acessa-lo de qualquer lugar, desde que este individuo tenha conexão com a rede empresa e também tenha login e senha para acesso a este sistema junto a algum navegador.

Além disso há mais algumas vantagens em aplicações Web:

* Nos dias atuais navegadores permitem sites com interfaces ricas e funcionais;
* Linguagens de programação para web são simples;
* O *deployment* da web site ocorre uma vez, assim ao invés de aplicações desktop que os *deployments* são maquina por máquina e as implementações ocorrem imediatamente.

Visando os pontos fortes e os atributos de uma aplicação web, eles conduziram a escolha deste formato de aplicação para esta situação específica da Empresa estudada.

**4.3 Modelagem de Sistemas Utilizando UML**

UML *(Unified Modeling Language -* Linguagem de Modelagem Unificada*)* foi desenvolvida para consolidar o desenvolvimento de métodos para o modelo de desenvolvimento Objeto Orientado. O principal proposito desta linguagem de modelagem, e permitir que o desenvolvedor possa utiliza-lo de maneira simples e direta para modelagem da situação apresentada para ele. Assim, através de diagramas pode-se explicar o desenvolvimento de sistemas através de representação do mundo real.

Booch; Rumbaugh; Jacobson (2000), a UML é uma linguagem muito significativa, compreendendo todas as visões imprescindíveis ao desenvolvimento e implantação de sistemas de informações corporativos distribuídos a aplicações fundamentadas em Web e até mesmo, sistemas complexos de tempo real**.**

A utilização do UML será importante neste projeto para entender e formalizar a necessidade do cliente em relação ao contexto do mundo real em que ele se encontra e ter uma documentação demonstrando os primeiros caminhos para o desenvolvimento do sistema.

**4.4 Diagrama de Entidade Relacionamento (DER)**

Diagrama entidade relacionamento é um modelo diagramático que descreve o modelo de dados de um sistema com alto nível de abstração. Ele é a principal representação gráfica do Modelo de Entidades e Relacionamentos (SILVA, 2012).

No diagrama Entidade Relacionamento nota-se que há 3 tabelas, e duas tabelas de Log. A tabela de *usuario* contém o *idUsuario* que é o ID da Empresa estudada, *nome*, *sobrenome* e *centro\_de\_custo\_de\_pessoas* virão da base de dados já consolidadas da Empresa estudada. *IT-Partner* e uma auto referência da tabela usuário porque um *IT-Partner* vem da tabela usuário, um IT-Partner, “herda” todas as características da tabela usuário. A tabela *Departamento* tem vínculo com a tabela usuário de um para um. Já a tabela computadores está vinculada de muitos para um, com o atributo de *numero\_ativo, centro\_de\_custo\_ativo, modelo\_computador, leasing\_vencimento* e *backup,* no caso de maquinas backup o atributo *backup* terá uma flag *true.* As tabelas *log\_computador* e *log\_computador\_old,* respectivamente é uma *trigger* que toda vez que uma linha na tabela *log\_computador* e mudada um log e armazenada e na tabela *log\_computador* e salvo o dado antigo que foi modificado.

A screenshot of a social media post

Description automatically generated

Figura 1 – Diagrama entidade relacionamento

**5. Metodologia de Desenvolvimento**

A metodologia incremental foi escolhida para desenvolver esse projeto. Nela, terá entregas definitivas e ao longo do projeto, serão adicionadas novas entregas para o crescimento e melhoria do projeto.

A metodologia incremental é um modelo evolutivo, nele são definidos os elementos e como eles irão se relacionar dentro de cada entrega. Em consequência disso, o contato com o cliente é constante durante as mesmas, após o fim de cada entrega uma funcionalidade é testada e validada. Após testes e validações, um *feedback* é gerado e novas demandas são trabalhadas na próxima sequência do projeto.

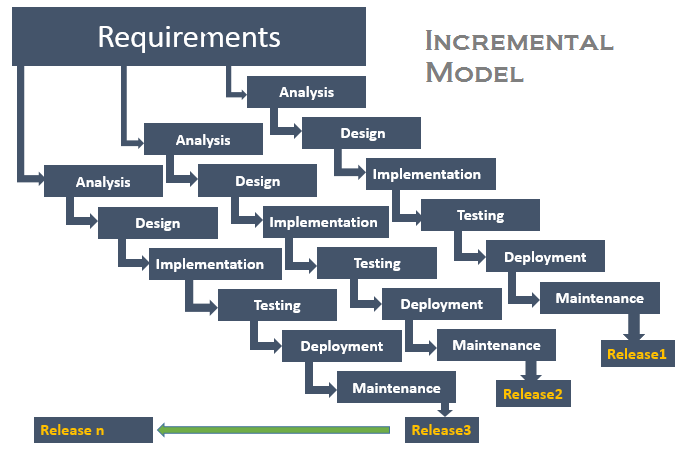


Figura 2 – Fluxo do Modelo Incremental

**5.1 Observação Natural**

A escolha da observação natural foi devido ao fato de que vários processos do usuário já serem dominar o processo que ocorre dentro da Empresa estudada. Observando seu trabalho, foi possível modelar um fluxo de trabalho junto com o mesmo.

Foi observado que o funcionário utiliza uma planilha para realizar *Crosschecks* [[1]](#footnote-2)sobre os computadores que ele controla. Esta planilha toma muito tempo do funcionário para ficar pronta pela atualização manual e a demonstração de dados ser lenta através de tabelas dinâmicas.

Com essa observação foi possível discutir e sugerir possíveis soluções de software para o funcionário da Empresa estudada, visando a otimização de tempo do mesmo.

**6. Analise e Especificação de Requisitos**

Analise e Especificação de Requisitos e fundamental para o desenvolvimento de software para as partes interessadas principalmente se tratando de custos.

Na figura 3, e possível observar que a Análise de Requisitos representa somente 5% do custo de desenvolvimento. Porem em 55% dos erros introduzidos estão ainda nesta fase. E se o erro ainda e observado nesta vaze o custo pode chegar a 1%.

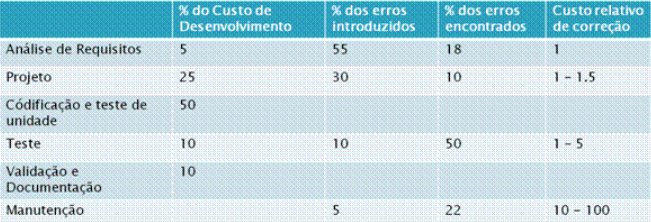


Figura 3 – Cenário de Custos de Software.

**6.1 Requisitos Funcionais**

Requisitos funcionais apoiam as descrições realizadas no sistema conforme o que o cliente pediu junto com o desenho do processo.

|  |
| --- |
| RF01 – Efetuar *Login*  O *“User ID”* do Administrador, deverá conter o mesmo padrão de *“User ID”* da Empresa estudada. |
| RF02 – Consultar Dashboards  Os *Dashboards* deverão ser gráficos de barras e quando *Drilldown* [[2]](#footnote-3)nos dados forem feitos, os *Dashboards* além de apresentarem os gráficos de barras, deverá ser apresentada tabelas com os dados pertinentes aquele gráfico. |
| RF03 – Cadastrar Usuários  Funcionalidade que é utilizada pelos usuários dos sistemas quando o usuário for novo. |
| RF04 – Editar Usuários  Esta função e executada quando usuários são desligados ou o centro de custo deles e mudado. |
| RF05 – Cadastrar Computador  Usuário cadastra computadores que estão vinculados ao centro do custo do usuário e o nome dele |
| RF06 - Gerar Relatório de maquinas  Gerar relatório de maquinas e usuários feita pelo usuário do sistema. |

Tabela 1 - Análise dos Requisitos Funcionais do Sistema

**6.2 Requisitos não funcionais**

Requisitos não funcionais são requisitos que mesmo que não sejam formalizados, mas há expectativas sobre as funcionalidades que ficam implícitas durante o funcionamento do sistema.

|  |
| --- |
| * O *software* deve ser aplicação WEB; * Os *Dashboards* devem ser visualmente de simples consulta como na Planilha; * O cadastro de maquinas deve seguir o padrão da Empresa observado. * O relatório gerado deverá ser em .CSV ou .XLSX |

*Quadro 1 – Requisitos não funcionais.*

**7. Casos de Uso**

O caso de uso e uma ferramenta que ajuda a descrever características e funções desejadas do sistema pelo usuário. A partir daqui, será possível arquitetar o sistema para refinamento e expansão para vários outros modelos que representam outras visões de sistema.

Aqui no quadro 2 será representado o fluxo de inserção de um novo computador no banco de dados. Este mesmo fluxo também poderá ser utilizado para representar cadastros de usuários.

|  |
| --- |
| **Editar Tabela** – Inserção de novo Computador na base de dados.  **Ator:** Administrador  **Objetivo:** Inserir um novo computador a base de dados.  **Pré-condições**  O administrador deve estar logadono sistema.  Acessar a tela do cadastro de computadores.  **Fluxo Principal**   1. O administrador irá a tela de inserção de novos computadores; 2. Clica no ícone de inserção de nova máquina e preenche o formulário com dados sobre o usuário, o nome do computador e centro de custo; 3. O Administrador Clica em “Submeter”; 4. O sistema grava a inserção no banco de dados; 5. O sistema devolve ao administrador um Feedback de Computador foi feita com sucesso.   **Fluxo Alternativo**   1. Caso o administrador desista de submeter o cadastro, ele poderá cancelar a operação clicando no botão “Cancelar”.   **Pós Condições**   1. Uma tela de avisa sinalizara que o cadastro foi feito com sucesso. |

Quadro 2 – Fluxo de um computador no Banco de Dados

No diagrama de Caso de Uso podemos ver os fluxos que estarão contidos no sistema. Pode-se notar nas primeiras elipses na figura 2, que o cadastro de computadores vem com funções de Criação, Leitura, Atualização e Deleção de registros na interface e banco de dados. E os *Dashboards* [[3]](#footnote-4) são apenas visualização de dados já gerados pela primeira ramificação de cadastro de computadores e usuários das máquinas. E o ator Gerente, conseguira extrair esses dados em um relatório completo do sistema.

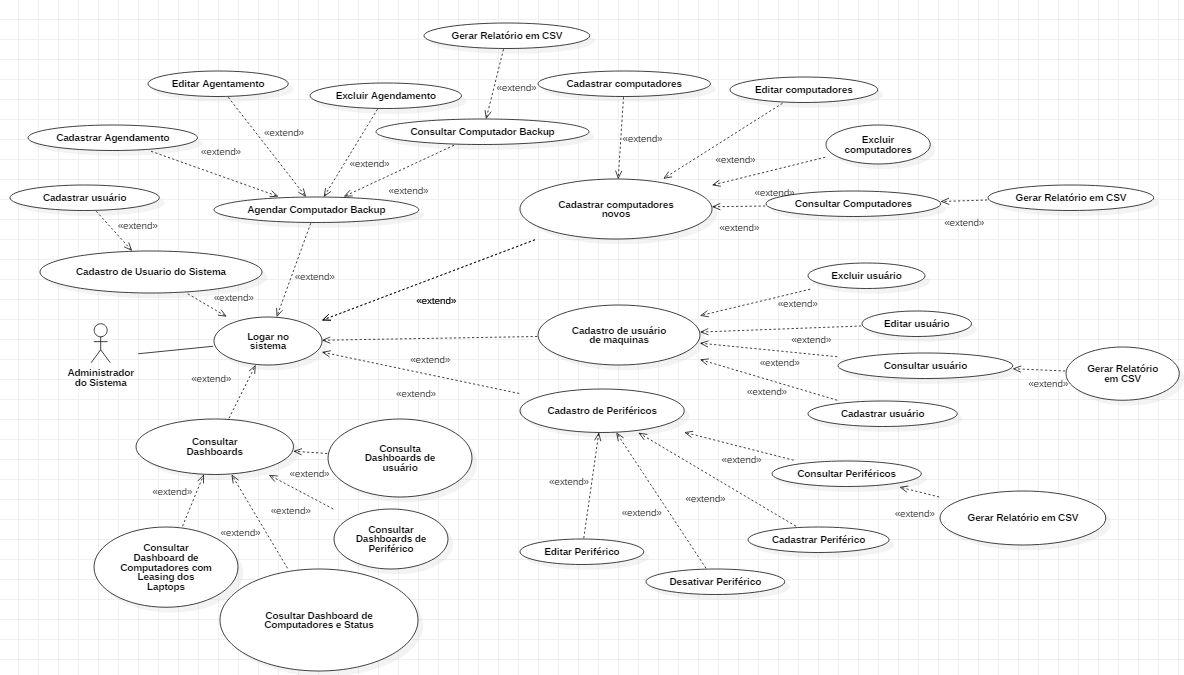


Figura 4 – Caso de uso visão do Usuário.

**8. Ambiente de Desenvolvimento**

**8.1 Navegador**

O navegador e um software que permite um indivíduo a acessar endereços (URL – *Uniform Resource Locator*). (JATINDRA KUMAR DAS, 2015, p. 324). Por tanto, o indivíduo poderá “surfar” na WWW (*World Wide Web*).

**8.2 PHP**

O PHP é uma linguagem de programação totalmente voltada a internet, possibilitando o desenvolvimento de sites dinâmicos. Sigla que hoje é um acrônimo recursivo para *PHP: Hypertext Preprocessor,* conquistou muito espaço nos últimos anos devido à facilidade de utilização e a grande quantidade de recursos que possui. (NIEDERAUER, 2004).

Esta linguagem e de simples uso em comparação com linguagens como o C ou Java, páginas PHP tem HTML incorporados ao código.

O código e executado através de um servidor, diferentemente de linguagens como por exemplo, o JavaScript. Com isso, e possível gerar código HTML e enviar para o cliente e também e possível gerar mais uma camada de proteção, fazendo os arquivos PHP e HTML inacessíveis ao usuário.

**8.3 HTML**

The HTML e uma linguagem de marcação do World Wide Web e o seu uso foi primeiramente utilizado para artigo científicos. Em linhas gerais, apesar desta linguagem ter sido construída para construção de artigo científicos, ela atualmente foi adotada para construção de uma miríade de aplicações baseadas nesta linguagem.

**8.4 CSS**

CSS (*Cascade Style Sheet)* ajuda na formatação das *tags* em HTML para entregar layouts visualmente interessantes para o desenvolvedor. Com CSS, será possível a edição de gráficos, menus e também determinar o Layout da página Web.

**8.5 JavaScript**

O JavaScript permite que o programador a dinamizar conteúdos estáticos nas páginas em HTML para interagir com o visitante da página. Como o JavaScript e uma linguagem leve, relativamente orientada a objetos e também funciona em muitos Navegadores. Com isso pode-se abrir muitas possibilidades para Design de páginas devido sua flexibilidade durante a programação.

**8.6 Servidor WEB**

Um Servidor Web pode ser definido tanto do de uma maneira física ou não física, tratando-se da aplicação Web que entrega o conteúdo Web que pode ser acessado através da internet. A função principal do Web Server e entregar páginas Web aos clientes utilizando do HTTT (*Hypertext Transfer Protocol)*.

**8.7 MySQL**

MySQL e um gerenciador de banco de dados relacional que opera como um gratuito que utiliza SQL (*Structured Query Language)* que e utilizada para inserção, retomada de dados manipulação de dados.

**8.8 StarUML**

Para o desenho de diagramas do *software* desenvolvido aqui foi utilizado o StarUML, na versão 2.8.0. O StarUML é um *software* gratuito de modelagem de diagramas baseada na UML com fácil interatividade e manuseio.

**8.9 GIT**

O GIT (*Global Information Tracker),* é um controlador de versão amplamente utilizado em projetos de sistemas de informação para controle e versionamento do sistema.

Alguns dos benefícios do GIT para projetos de sistemas de informação, seria a integridade e confiança, porque o GIT utiliza função de *Hash* criptografada chamada *Secure Hash Function (SHA1),* para identificar objetos no banco de dados.Apesar de não ser absoluto, em prática, foi considerado e comprovado que é solido o suficiente para garantir integridade nos repositórios do GIT. Além disto, o GIT considera o desenvolvimento em “*branches*”, que beneficia o desenvolvimento paralelo de um mesmo projeto.

**9. Implementação do Software**

**9.2 Login**

**9.1 Controle de Computadores**

**9.2 Cadastro de Usuário de Computadores**

**9.3 Cadastro de Periféricos**

**9.4 Empréstimo de Computadores**

**9.5 Dashboard**

**10. Considerações Finais**

Este projeto ainda está em fase de desenvolvimento, por tanto, utilizando-se do SCRUM para explicar onde estamos no desenvolvimento do projeto, foram feitos a entrevistas, analise de requisitos, plano de atividades do projeto, planejamento do escopo, cronograma, planejamento de qualidade e definição das atividades.

Depois de planejar todas as atividades citadas a cima, o projeto poderá partir para execução do plano de atividades do projeto, neste caso serão os *backlogs* do projeto que deverão seguir o plano de atividades e o cronograma de projetos.

Ao final de cada *Sprint,* então será verificado junto ao plano de qualidade que está atrelado aos requisitos funcionais e não funcionais que serão testados posteriormente pelo cliente e validado.

Ao final do projeto, para continuidade do ciclo de vida do projeto, o cliente será acompanhado para fase de suporte e correção de *bugs* com o sistema e propostas de melhoria para o Sistema de Inventário e Empréstimos de Computadores e Periféricos.

Ao final do ciclo de vida do projeto, será apresentado mostrando os conceitos utilizados aqui junto ao sistema desenvolvido.

**Referências:**

BEAIRD, J.; GEORGE, J. **The Principles of Beautiful Web Design**. [S.l.]: Site Point Pty. Ltd, 2010.

BRANDON, D. M. **Software Engineering for Modern Web Applications**. Hershey: Information Science Reference, 2008.

DE FATIMA COELHO, F. Impacto de requisitos não funcionais em projetos de software. **DevMedia**, 2016. Disponivel em: <https://www.devmedia.com.br/impacto-de-requisitos-nao-funcionais-em-projetos-de-software/36858>. Acesso em: 05 Dezembro 2017.

LAUDON, K. C.; LAUDON, J. P. **Management Infomation Systems**. Harlow: Person Education Limited, 2014.

LOELIGER, J. **Version Control with Git**. 1st. ed. Beijing, Cambridge, Farnham, Köln, Sebastopol, Taipei, Tokyo: O'Reilly, 2009.

MEDEIROS, H. Introdução aos Processos de Software e o Modelo Incremental e Evolucionário. **DevMedia**, 2013. Disponivel em: <https://www.devmedia.com.br/introducao-aos-processos-de-software-e-o-modelo-incremental-e-evolucionario/29839>.

MENDES, A. Gestão de Projetos de Software. **DevMedia**, 2009. Disponivel em: <https://www.devmedia.com.br/gestao-de-projetos-de-software/9143>. Acesso em: 05 Dezembro 2017.

PINTO, D. S. &. M. **The Web Application Hacker's Handbook**. Indianapolis: John Wiley & Sons, Inc., 2011.

PRESSMAN, R. S. . P. D. **Software Engineering A Practioncer's Approach**. New York: McGraw-Hill, 2010.

RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. **The Unified Modeling Language Reference Manual**. Reading, Harlow, Menlo Park, Berkeley, Don Mills, Sydney, Amsterdam, Tokyo, Mexico City: Addison-Wesley Longman Inc., 1999.

SHEMA, M. **Hacking Web Apps Detecting and Preventing Web Application Security Problems**. Waltham: Elsevier, 2012.

SHKLAR, L.; ROSEN, R. **Web Application Architecture Principles, protocols and practices**. Chichester: John Wiley & Sons Ltd, 2003.

TAHAGHOGHI, S. M. M.".; WILLIAMS, H. E. **Learning MySQL**. Sebastopol: O'Reilly Media, 2007.

1. *Crosschecks: Comparação cruzada entre informações.*  [↑](#footnote-ref-2)
2. *Drilldown*: Processo de visualização em seus níveis mais altos para os mais baixos. [↑](#footnote-ref-3)
3. *Dashboards: Painéis de visualização de Gráficos.*  [↑](#footnote-ref-4)