**Livre Facile**

**Plano de Projeto**

**Versão <2.3>**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| <21/03/2019> | <1.0> | <Nome da empresa, histórico de revisão, introdução> | <Douglas, Daniel, Rafael, Welliton> |
| <04/04/2019> | <1.1> | <Atualização do escopo e introdução, objetivos> | <Rafael> |
| <09/04/2019> | <1.2> | <Criação do modelo conceitual de banco de dados> | <Daniel, Douglas, Rafael |
| <10/04/2019> | <1.3> | <Atualização no esboço e introdução, Atualização do modelo conceitual do banco de dados> | <Douglas, Rafael> |
| <17/04/2019> | <1.4> | <Prototipação de telas> | <Welliton> |
| <24/04/2019> | <1.5> | <Implementação parcial do banco de dados, e finalização da prototipação de telas> | <Rafael, Douglas, Daniel> |
| <29/04/2019> | <1.5> | <Linkar banco de dados com o código java para cadastramento de clientes> | <Rafael, Douglas, Welliton> |
| <03/05/2019> | <1.6> | <Atualização de histórico de revisão, e documentação em geral, e implementação de manter cliente> | <Daniel, Douglas, Rafael, Welliton> |
| <16/05/2019> | <1.7> | <Funcionalidade Manter Funcionário foi adicionada> | <Rafael, Douglas> |
| <23/05/2019> | <1.8> | <Banco de dados Implementado e Funcional> | <Rafael, Douglas, Daniel> |
| <05/06/2019> | <1.9> | <As Funcionalidades Manter Cliente, Manter Livro e Empréstimo foram adicionadas > | <Rafael, Douglas, Daniel, Welliton> |
| <19/06/2019> | <2.0> | <JOB (Agendamento automático para execução), Realizar Devolução > | <Rafael, Douglas> |
| <20/06/2019> | <2.1> | < Funcionalidade Gerar Relatório > | <Daniel, Welliton> |
| <03/07/2019> | <2.2> | <Testes do sistema, Correção de Bugs> | <Rafael. Daniel, Douglas, Welliton> |
| <05/07/2019> | <2.3> | <Revisão e Documentação dos Testes, Criação de Logo, Correção de Bugs> | <Rafael, Daniel, Douglas, Welliton> |

**Índice Analítico**

1. Introdução 4

2. Escopo e Propósito do documento 4

3. Objetivos do Projeto 4

3.1 Objetivos 4

3.2 Funções Principais 4

3.3 Questões de Desempenho 4

3.4 Restrições Técnicas e Administrativas 4

4. Estimativas 4

4.1 Técnica 4

4.2 Estimativa 4

5. Riscos do Projeto 4

6. Cronograma 4

7. Recursos do Projeto 5

7.1 Pessoal 5

7.2 Hardware 5

7.3 Software 5

7.4 Recursos Especiais 5

8. Organização de Pessoal 5

8.1 Estrutura de Equipe 5

**Plano de Projeto**

1. **Introdução**

Este documento apresenta o plano de projeto para o desenvolvimento de um novo sistema que automatize o controle da biblioteca Livre Facile. O plano contém um conjunto de características que descrevem o projeto, onde ele se fará util, seus objetivos, um esboço de suas funcionalidades, apresenta questões sobre o desempenho do software, estimativa do seu tempo de produção demonstrando um cronograma completo a ser seguido incluindo todos detalhes técnicos de hardwares e de softwares necessários para sua conclusão.

1. **Escopo e Propósito do documento**

A biblioteca **Livre Facile** precisa de um software computacional que substitua os principais serviços que até então são realizados manualmente pelos seus funcionários. A atividade principal oferecida pela biblioteca é o controle dos empréstimos e reservas de exemplares para usuários, para esse controle é necessário guardar informações sobre os livros e usuários, essa manipulação de dados que atualmente é feita manualmente passará a ser feita através do software que será desenvolvido durante o projeto. O software facilitará para que os usuários consultem o acervo, verificando- se um determinado livro está disponível através do seu título, autor ou assunto. O sistema será desenvolvido para ambiente desktop, o usuário terá apenas acesso a realização de pesquisas sobre o acervo, as demais funções estarão restritas para utilização dos funcionários, que possibilitará o uso de funções como cadastrar e excluir usuários e livros, assim como atualizar os dados dos mesmos, realizar empréstimos, gerar multas, bloquear ou desbloquear usuário e gerar relatórios que contém estatísticas e informações dos livros mais procurados para poder se adquirir mais exemplares. Para cada livro será gerado um código único para sua identificação e localização dentro da biblioteca.

* 1. **Objetivos**

O projeto tem como objetivo facilitar a vida dos usuários e funcionários da biblioteca, tornando o atendimento mais rápido, eficiente e com um melhor controle sobre empréstimos e também do acervo, disponibilizando um software simples e objetivo que realiza cadastros, empréstimos, buscas, reservas, etc.

* 1. **Funções Principais**

As principais funções a serem implementadas no projeto são:

- Cadastro, remoção e atualização de clientes.

- Cadastro, remoção e atualização de funcionários.

- Cadastro de livros para o controle do acervo.

- Pesquisa de acervo.

- Controle de reservas e empréstimos.

- Gerar relatório.

* 1. **Questões de Desempenho**

O software a ser desenvolvido tem como objetivo ser de fácil e intuitiva utilização, visando um desempenho sólido e funcional, utilizando recursos de linguagem de programação simples, que resultarão em um software conciso e que desempenhe com excelência o que foi proposto.

* 1. **Restrições Técnicas e Administrativas**

O desenvolvimento do software será realizado em linguagem Java, utilizando o ambiente de desenvolvimento integrado NetBeans. Para implementação e gerenciamento do banco de dados será utilizado o PostgreSQL, para o projeto conceitual e para a modelagem do banco de dados usaremos o BrModelo.

O tempo de desenvolvimento por se tratar de um projeto universitário ficará restrito ao semestre letivo, portanto, entre 3-4 meses. Seu custo real será o tempo que cada integrante da equipe se dedicará ao desenvolvimento e manutenção do projeto.

1. **Estimativas**
   1. **Técnica**

As estimativas documentadas constituem a base para a elaboração de um plano do projeto de software. As estimativas de tamanho de projetos de software devem ser utilizadas para a derivação das estimativas de custo, esforço e cronograma, a técnica de estimativas escolhida a ser utilizada é a Use-Case Point.

* 1. **Estimativa**

Fator de Complexidade Técnica (TCF):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fator** | **Descrição** | **Peso** | **Valor** | **P\*V** |
| T1 | Sistema Distribuído | 2 | 2 | 4 |
| T2 | Performance | 1 | 3 | 3 |
| T3 | Eficiência do usuário final | 1 | 4 | 4 |
| T4 | Facilidade de uso | 0.5 | 4,5 | 2,25 |
| T5 | Facilidade de manutenção | 1 | 3 | 3 |
| T6 | Concorrência | 1 | 1 | 1 |
| T7 | Características especiais de segurança | 1 | 4 | 4 |
| T8 | Provê acesso direto para terceiros | 0,5 | 4 | 2 |
| T9 | Necessidade de treinamento especializado | 1 | 2 | 2 |
| T10 | Processamento interno complexo | 1 | 3 | 3 |
| T11 | Reusabilidade | 1 | 4 | 2 |
| T12 | Facilidade de instalação | 0,5 | 4 | 2 |
| T13 | Portabilidade | 2 | 3 | 6 |
|  |  |  |  | 38,25 |

**TCF = 0,6 + (0,01\*38,25) = 0,982**

Fator de Complexidade Ambiental (ECF):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fator** | **Descrição** | **Peso** | **Valor** | **P\*V** |
| E1 | Familiaridade com UML | 1,5 | 4 | 6 |
| E2 | Experiência na aplicação | 0,5 | 2 | 1 |
| E3 | Experiência em Orientação a Objetos | 1 | 4 | 4 |
| E4 | Capacidade de liderança do analista | 0,5 | 2 | 1 |
| E5 | Motivação | 1 | 2 | 2 |
| E6 | Requisitos estáveis | 2 | 4 | 8 |
| E7 | Funcionários período parcial | -1 | 0 | 0 |
| E8 | Dificuldade na linguagem de programação | 2 | 4 | 8 |
|  |  |  |  | 30 |

**ECF = 1,4 + (-0,03 \* 30) = 0,5**

Pontos de Caso de Uso Não Ajustados (UUCP):

- Unadjusted Use-Case Weight (UUCW)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo** | **Descrição** | **Peso** | **Nº** | **Total** |
| Simples | Uma interface simples que afeta apenas uma entidade no banco de dados; seu fluxo principal afeta 3 passos ou menos. Sua implementação envolve menos de 5 classes | 5 | 4 | 20 |
| Médio | Mais interfaces e afeta 2 ou mais entidades de banco de dados; entre 4 a 7 passos; Sua implementação envolve de 5 a 10 classes | 10 | 4 | 40 |
| Complexo | Envolve uma interface ou processamento complexos e afeta 3 ou mais entidades de banco; Sua implementação envolve mais de 10 classes | 15 | 2 | 30 |

- Unadjusted Actors Weight (UAW)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo** | **Descrição** | **Peso** | **Nº** | **Total** |
| Simples | O ator representa outro sistema com uma API definida. | 1 | 0 | 0 |
| Médio | O ator representa outro sistema interagindo através de um protocolo, tipo TCP/IP. | 2 | 0 | 0 |
| Complexo | O ator é uma pessoa interagindo via interface. | 3 | 3 | 9 |

**UUCP = UUCW + UAW = 90 + 9 = 99**

Fator de Produtividade:

**O valor estimado é 4.**

Cálculo Final:

**UCP = TCF \* ECF \* UUCP \* PF**

**UCP = 0,982\* 0,5 \* 99 \* 4**

**UCP = 194,436**

**A estimativa é que seriam necessárias aproximadamente 194 horas para desenvolver o projeto.**

1. **Riscos do Projeto**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID do Risco | Descrição do Risco | Probabilidade | Impacto | Plano de Redução do Risco | Plano de Contingência |
| R01 | Greve UTFPR | Baixo | Alto | Participar das reuniões do DCE e ficar atento as notícias sobre a universidade. | Continuar o projeto informalmente tirando dúvidas via e-mail |
| R02 | Não alcance dos requisitos | Alto | Baixo | Fazer uma boa e minuciosa analise de requisitos no primeiro momento | Mudança do plano de projeto |
| R03 | Alta complexidade técnica de implementação | Médio | Médio | Solicitar consultoria em especialistas na área | Contratar especialista |
| R04 | Problemas de saúde na equipe | Baixo | Alto | Bom ambiente de trabalho dentro da empresa.  Fazer rotação da equipe para que todos estejam inteirados sobre o projeto | Substituir funcionário por outro que tem domínio do mesmo assunto. |
| R05 | Falha técnica de equipamentos usados para o projeto | Baixo | Médio | Fazer backups do projeto em vários servidores | Encontrar máquina reserva para dar continuidade no projeto |
| R06 | Falha em algum software essencial no desenvolvimento | Baixo | Alto | Fazer backups do projeto na nuvem | Corrigir o problema ou reinstalar o software |
| R07 | Falha de Comunicação no Banco de Dados | Baixo | Alto | Realizar todos os testes possíveis antes de finalizar o Software | Encontrar a falha de comunicação e corrigi-las. |

1. **Cronograma**

|  |  |  |  |
| --- | --- | --- | --- |
| Etapa/Atividade | Data Inicial | Data Final | Responsável |
| Sprint 01 | 04/04/2019 | 17/04/2019 | Rafael – Scrum Master |
| Sprint 02 | 18/04/2019 | 01/05/2019 | Rafael – Scrum Master |
| Sprint 03 | 02/05/2019 | 23/05/2019 | Rafael – Scrum Master |
| Sprint 04 | 24/05/2019 | 06/06/2019 | Rafael – Scrum Master |
| Sprint 05 | 07/06/2019 | 19/06/2019 | Rafael – Scrum Master |
| Sprint 06 | 20/06/2019 | 05/07/2019 | Rafael – Scrum Master |

1. **Recursos do Projeto**
   1. **Pessoal**

Serão necessários os seguintes recursos para o desenvolvimento do projeto:

- Analista de requisitos. – 6 hrs

- Programador Java. – 40 hrs

- Gerenciador de banco de dados. – 30 hrs

- Gerente de projeto. – 30 hrs

- Equipe para teste. – 20 hrs

* 1. **Hardware**

Os recursos de hardware necessários são 4 notebooks para realização das tarefas pertinentes ao projeto. Recursos os quais já estão disponíveis.

* 1. **Software**

Os softwares adotados para realização do projeto são:

- NetBeans para programação.

- PostgreSQL para implementação e gerenciamento do banco de dados.

- BrModelo para modelagem do banco de dados.

- ProjectLibre para gerenciamento e estimativas do projeto.

- Visual Paradigm para o diagrama de classes, diagrama de sequência e diagrama de casos de uso.

Todos os softwares são de licença gratuita.

* 1. **Recursos Especiais**

Alguns recursos especiais necessários para o sólido andamento do projeto são:

- Consultoria e mentoria da professora Dra. Viviane.

- Consultoria em banco de dados e Java pelo prof. Dr. Ives.

- Cursos online conforme necessitado em certas fases do projeto.

1. **Organização de Pessoal**
   1. **Estrutura de Equipe**

A equipe é composta por 4 integrantes, sendo eles:

Daniel Taborda Afonso – Analista e Programador.

Douglas Ramos Marçal – Analista e Programador.

Rafael Anderson Dalmolin – Analista, Programador e Scrum Master.

Welliton Leal – Analista e Programador.