Controle financeiro

Plano do Projeto

# Introdução

Este documento contém o planejamento geral do projeto Acadêmico que será desenvolvido como estudo na disciplina de Engenharia de Software II.

# Organização do projeto

Daniel Eduardo Costa de Almeida - responsável pelo desenvolvimento, suporte e instalação do *software*.

# Práticas e medidas do projeto

Modelo de ciclo de vida pode ser entendido como uma prescrição de um conjunto de elementos de processo adaptáveis às pessoas, aos problemas e ao projeto. Seu principal objetivo é ordenar o desenvolvimento de software. Após analisar os diversos tipos de modelos prescritivos de processo, optou-se pela escolha do modelo OpenUP, cuja referência pode ser encontrada em: <http://epf.eclipse.org/wikis/openup/>.

As métricas são metodologias de mensuração cujos principais objetivos são de estimar o tamanho de um *software* e auxiliar no gerenciamento dos projetos de desenvolvimento de sistemas. Após analisar as métricas disponíveis optou-se pelo método *User Case Points*.

Em relação ao desenvolvimento do *software* será adotada a automatização de testes. Com testes automatizados consegue-se entender melhor os problemas, já que o desenvolvedor, pela prática, valida sua hipótese considerando diferentes cenários. Além disso, reduz-se o stress e aumenta-se a satisfação, pois com um bom conjunto de testes, *bugs* são detectados mais cedo no ciclo de desenvolvimento e menos problemas chegam ao cliente, diminuindo com isso o custo.

O código do projeto pode ser encontrado em: <https://github.com/DanielEduardoA/ControleFinanceiro.git>

# Marcos e objetivos do projeto

|  |  |  |
| --- | --- | --- |
| **Iteração** | **Objetivos primários** | **Início ou fim agendado** |
| I1 | * Estabelecer a viabilidade do projeto (técnica e de requisitos) * Criar um caso de negócios para demonstrar objetivos de negócios quantificáveis * Capturar os requisitos essenciais do sistema * Identificar riscos críticos | 05/02/2018  13/03/2018 |
| I2 | * Criar uma *baseline* arquitetural executável * Refinar a avaliação de riscos * Definir atributos de qualidade * Capturar casos de uso para 80% dos requisitos funcionais * Planejar detalhadamente a fase de construção * Capturar estimativa de recurso, tempo, equipamento, pessoal e custo | 19/03/2018  24/04/2018 |
| I3 | * Completar os requisitos, análise e projeto * Construir o sistema funcional em cima da *baseline* arquitetural entregue pela fase de elaboração | 30/04/2018  29/05/2018 |
| I4 | * Corrigir defeitos * Preparar o ambiente de usuário para o novo *software* * Adaptar o *software* para operar no ambiente de usuário * Modificar o *software* devido a problemas imprevistos * Criar manual de usuário e outras documentações * Prover consultoria ao usuário * Conduzir revisões do projeto | 04/06/2018  19/06/2018 |

# Implantação

A implantação do sistema será realizada em uma única etapa, tendo em vista que o *software* não é integrado a outros aplicativos ou sistemas, sendo assim não será necessária uma integração, somente sua instalação e configuração.

Para que a implantação do sistema seja completada efetivamente e obtenha os resultados esperados, as diretrizes de implantação precisam ser seguidas rigorosamente e os treinamentos devem ser ministrados a todos os usuários. Para isso foi elaborada a seguinte matriz de atividades.

|  |
| --- |
| Atividades |
| **Planejamento** |
| Levantamento de recursos necessários de *hardware* |
| Levantamento de recursos necessários de *software* |
| Definição de estratégias para conversão e migração de dados |
| Programação dos treinamentos |
| Preparação dos testes de aceitação |
| **Execução** |
| Configuração de infraestrutura de *hardware* |
| Configuração de infraestrutura de *software* |
| Instalação do produto |
| **Avaliação** |
| Acompanhamento pós-implantação |

# Lições aprendidas

Não se aplica.