**ESPECIFICAÇÃO E PROJETO DE DESENVOLVIMENTO DE SOFTWARE**

1. **DADOS DE IDENTIFICAÇÃO** 
   1. **Título do Projeto**

Sistema para controlar setor interno de uma imobiliária, através do uso de tarefas.

* 1. **Responsável pelo Projeto**

*Igor Camargo Moiano*

*Igor.moiano@hotmail.com*

1. **OBJETIVOS**

O sistema tem como principal objetivo ser capaz de manter o controle de fluxo de atividades individual de colaboradores, também disponibilizando acesso restrito a funcionalidades para determinados níveis de cada membro. As tarefas serão armazenadas de forma que possam ser atribuídas a determinados membros. Sendo assim mantendo um histórico das atividades cotidianas realizadas.

1. **ESPECIFICAÇÃO INICIAL DOS REQUISITOS**
   1. **Método de Levantamento de Requisitos**

Os requisitos serão coletados através da perspectiva de um(a) gerente de uma imobiliária, pois este cargo tende a definir tarefas e realizar tarefas, sendo assim serão realizadas perguntas quanto a tipos de atividades o(a) entrevistado(a) tende a realizar e a definir para que seus subordinados tenham de realizar, juntamente informações necessárias para a realização da tarefa.

* 1. **Lista de funcionalidades**

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de ordem** | **Funcionalidade** | **Necessidades** | **Benefícios** |
| 1 | Cadastrar  gerente | Seu cadastro será realizado por um gerente já pré-cadastrado, e todas as outras demais funcionalidades responsáveis pelo gerenciamento do gerente fica responsável pelos de mesmo cargo. Utilizara a mesma interface de cadastro de um contribuinte, alterando apenas o tipo do membro na equipe. | Dessa forma o sistema fica sobre total controle dos gerentes, garantindo segurança quanto as informações do sistema. |
| 2 | Cadastrar contribuinte | Somente um gerente será responsável pelo cadastro de um contribuinte, assim como todo o seu gerenciamento, quando a alteração, e exclusão. Utilizara a mesma interface de cadastro de um gerente, alterando apenas o tipo do membro na equipe. | Os contribuintes ficam responsáveis apenas pela realização de suas tarefas, sem ter que se envolver com o gerenciamento dos mesmo de seu cargo, e garante que os dados não serão alterados por outros membros da equipe que não sejam gerentes. |
| 3 | Controlar tarefas | Saber quais tarefas devem ser realizadas. | Agilidade na definição de tarefas, diminuindo a necessidade da pessoa física definir as tarefas. Identifica o que cada usuário esta fazendo e qual atividade deve realizar, tornando o processo de resolução das atividades de forma organizada. |
| 4 | Controlar membros equipe | Fornece informações de cada contribuinte ou gerente do sistema. | Mostrando suas devidas informações cadastradas no sistema, disponibilizando alteração de dados e exclusão para membros do tipo gerente. |
| 5 | Gerar histórico geral de todas as tarefas | Quando um gerente precisar se ausentar por algum motivo, com o histórico será possível verificar as tarefas realizadas. | Sendo assim um novo gerente que assumir o cargo não ficara dependente da pessoa física que assumia a função anterior a ele explicar suas tarefas e o que deve ser realizado. |

1. **ESTUDO DE VIABILIDADE**

O sistema pode ser implementado utilizando a linguagem de programação Php e JavaScript, pois grande parte dos sistemas imobiliários como Jetimob em Santa Maria tende a utilizar Php para desenvolvimento Web. Devido o contato próximo a um(a) gerente a coleta de requisitos será o mais próximo possível de situação real, tornando a ferramenta algo que será utilizado no contexto de uma determinada imobiliária..

1. **PLANO DE ATIVIDADES E CRONOGRAMA**
   1. **Estudo de tecnologia**

Foi definida uma media em dias para retomar os conceitos de php, php orientado a objetos e aprender JavaScript, que ocorrera junto com o decorrer do projeto, visando terminar a revisão e o aprendizado antes do desenvolvimento da ferramenta final.

* 1. **Pré-proposta**

A pré-proposta está sendo desenvolvida baseada nas necessidades de um setor imobiliário, este que possui problemas quanto sua organização de suas tarefas internas.

* 1. **Projeto de Software**

Será feito o levantamento de requisitos necessários para o desenvolvimento do sistema, juntamente com seus devidos diagramas e definição da estrutura do banco de dados. Visando ao termino desta etapa ser possível apresentar imagens que represente um protótipo inicial do sistema. O protótipo será desenvolvido primeiramente utilizando ferramentas para desenhar a versão inicial, para que após uma estruturação visual do sistema, definir se a interface condiz com as necessidades do cliente.

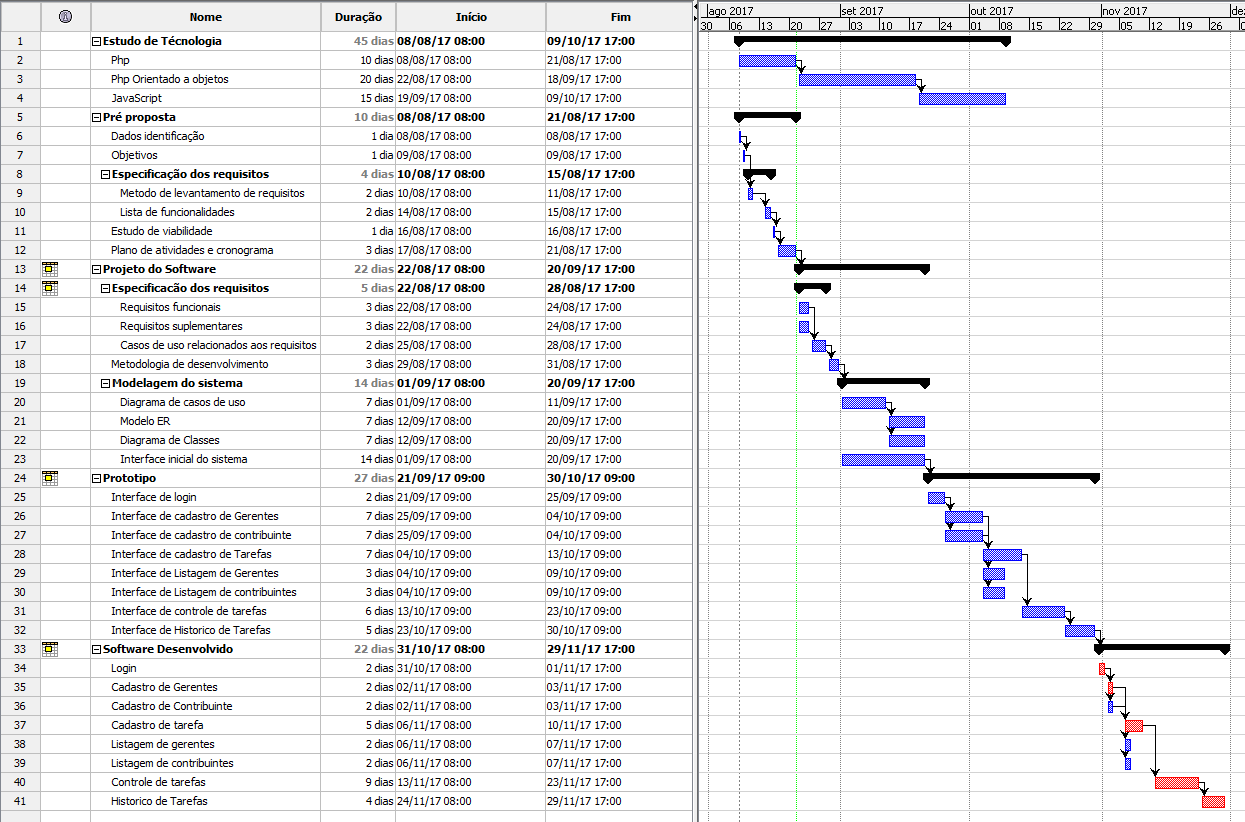
* 1. **Protótipo**

Após a aprovação da interface pelo cliente, será desenvolvido um protótipo mais próximo da ferramenta final, já sendo implementado utilizando html, css e php. Visando verificar se todos os requisitos estão sendo atendidos.

* 1. **Software desenvolvido**

Todo o desenvolvimento back-end do sistema estará separado em etapas semelhantes ao protótipo, visando a implementação dos CRUDs como prioridade para que possa ser possível verificar se as demais funcionalidades que precisam das informações básicas estarão funcionando. Como a logica do negocio esta baseada nas tarefas, foi definido um tempo maior para esta etapa, pois recebera muitas alterações, pelo fato de que o desenvolvimento do controle das tarefas será feito utilizado JavaScript para criar formulários já pré-definidos, baseados no tipo de tarefa selecionada pelo gerente.

* 1. **Cronograma**



1. **ESPECIFICAÇÃO DOS REQUISITOS**
   1. **Requisitos Funcionais**

*Especificar todos os requisitos funcionais do projeto a ser desenvolvido e os requisitos não funcionais diretamente ligados a cada requisito funcional*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F1 <Manter Funcionário>** | | | **Oculto ( )** | |
| O sistema devera fazer a inclusão, exclusão, alteração e listar os funcionários cadastrados. Mantendo dados como nome, telefone, endereço, empregado, verificar o que mais precisa | | | | |
| **Requisitos não-funcionais associados** | | | | |
| **Nome** | **Restrição** | **Categoria** | | **Desejável** |
| NF1.1 Controle de acesso | A função só poderá ser acessada por um gerente ou um administrador do sistema. | Segurança | | ( ) |
| NF1.2 Exclusão de funcionários. | Esta funcionalidade não remove totalmente o funcionário do banco, pois é necessário manter o registro sendo assim penas definido a variável empregado para false. | Usabilidade | | ( ) |
| NF1.3 Inserção de funcionário | Durante a inserção de um novo funcionário o mesmo é definido com a variável empregado true | usabilidade | | ( ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F2 <Manter Tarefa>** | | | **Oculto ( )** | |
| O sistema devera fazer a inclusão, exclusão, alteração e listar as tarefas cadastradas. Mantendo dados como descrição, funcionário(s) responsável(is), data limite de conclusão, local onde deve ser realizado, verificar o que mais precisa | | | | |
| **Requisitos não-funcionais associados** | | | | |
| **Nome** | **Restrição** | **Categoria** | | **Desejável** |
| NF1.1 Controle de acesso | A função só poderá ser acessada por um gerente ou um administrador do sistema. | Segurança | | ( ) |
| NF1.2 Exclusão de tarefas. | Esta funcionalidade não remove totalmente a tarefa do banco, pois é necessário manter o registro. | Usabilidade | | ( ) |

*ATENÇÃO: repetir esta tabela para cada requisito funcional existente.*

* 1. **Requisitos Suplementares**

*Corresponde aos requisitos não funcionais pertinentes ao sistema como um todo. Os requisitos suplementares são todo tipo de restrição tecnológica ou lógica que se aplica ao sistema como um todo e não apenas a funções individuais.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Restrição** | **Categoria** | **Desejável** |
| S1 <Nome Requisito não-funcional geral> |  |  | ( ) |
| S2 <Nome Requisito não-funcional geral> |  |  | ( ) |
| S3 <Nome Requisito não-funcional geral> |  |  | ( ) |

* 1. **Casos de uso relacionados aos requisitos funcionais**

*A tabela a seguir mostra a relação entre os casos de uso do sistema e os requisitos que os especificam.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome caso de uso** | **Atores** | **Descrição** | **Referências Cruzadas*\**** |
| *<Caso de uso 1>* |  |  |  |
|  |  |  |  |

*\*Cada caso de uso será associado a um conjunto de requisitos funcionais do sistema. Indicar o identificador de cada requisito funcional associado ao caso de uso.\*Cada caso de uso será associado a um conjunto de requisitos funcionais do sistema. Indicar o identificador de cada requisito funcional associado ao caso de uso.*

1. **METODOLOGIA DE DESENVOLVIMENTO**

*Descrever a metodologia que será utilizada para o desenvolvimento do sistema. [para acompanhar as atividades, para implementar, para troca de informações com o cliente, etc.]*

1. **MODELAGEM DO SISTEMA** 
   1. **Diagrama de casos de uso**

*Elaborar o diagrama de casos de uso (notação UML) que represente o sistema.*

* 1. **Descrição dos casos de uso**

*Descrever textualmente os principais casos de uso do sistema. Sistemas pequenos devem apresentar todos os casos de uso, já sistemas maiores podem apresentar apenas os casos de uso das principais funcionalidades, de forma a não deixar o texto tão repetitivo.*

* + 1. **Caso de uso <nome 1>**

|  |  |
| --- | --- |
| Descrição: | |
| Ator(es) envolvido(s): | |
| Pré-condição: | |
| **Fluxo Normal:** | |
| Ator | Sistema |
|  |  |
|  |  |
|  |  |
|  |  |
| **Fluxo Alternativo:** | |
|  | |
| **Fluxo de exceção:** | |
|  | |

*ATENÇÃO: repetir esta tabela para cada caso de uso existente.*

* 1. **Modelo ER**

*Além do diagrama E-R, fazer* uma descrição das entidades no formato de uma tabela (no máximo um parágrafo para cada entidade).

* 1. **Diagrama de Classes**

*Apresentar o diagrama de classes do projeto do sistema (conversar com o orientador).*

* 1. **Interface do Sistema (Prototipação de baixa, média ou alta fidelidade)**

*Apresentar um esboço das interfaces do sistema (rascunho do design ou projeto das páginas HTML que irão compor a aplicação).*