**Projeto Interdisciplinar das disciplinas: Engenharia de Software II / Desenvolvimento Web II / Banco de Dados Relacional**

**Nome aluno (a):** Felipe Avelino Pedaes

**Nome aluno (a):** Henrique Almeida Florentino

**Nome aluno (a):** Luiz Felipe Vieira Soares

Desenvolvimento

**1. Introdução**

**Propósito (ou Finalidade)**

“O projeto PawsSafety consiste em um ambiente virtual dedicado a apoiar a causa animal e promover a adoção responsável. A ideia surgiu após uma análise realizada pelo nosso grupo, onde percebemos a dificuldade de encontrar animais para adoção de maneira confiável.”

**Escopo**

“O projeto foi idealizado como uma plataforma para resolver esses problemas. Nela, os usuários podem se cadastrar, listar seus animais para adoção e adotar animais de outros usuários. Isso cria um ecossistema centrado que facilita as adoções e apoia indivíduos interessados em garantir a segurança dos animais durante todo o processo.”

**Benefícios Esperados**

A PawsSafety oferece uma plataforma centralizada e confiável para facilitar a adoção de animais, promovendo seu bem-estar e segurança. Isso cria uma comunidade engajada de pessoas interessadas em ajudar animais necessitados, promovendo a responsabilidade social entre empresas e indivíduos que apoiam a causa.

***Business Process Modeling Notation* ( BPMN )**

*Inclua o BPMN dos processos do sistema* (será preenchido no próximo semestre de DSM)

**Gráfico

Descrição gerada automaticamente com confiança baixa**

***Estórias de Usuários***

**Estória de Usuário 1:** *Como um usuário interessado em adotar um animal, desejo poder me cadastrar na plataforma PawsSafety para acessar as funcionalidades de adoção de animais..*

**Descrição:**

* O usuário deve ser capaz de acessar a página de cadastro na plataforma.
* Deve ser fornecido um formulário de cadastro com campos para inserção de informações pessoais, como nome, e-mail, senha, endereço, entre outros.
* Após o preenchimento do formulário, o usuário deve receber um e-mail de confirmação para validar seu cadastro.
* Após a confirmação do cadastro, o usuário deve ser redirecionado para a página inicial da plataforma logado em sua conta.

**Estória de Usuário 2:** *Como um usuário que deseja listar um animal para adoção, quero poder inserir os dados do animal na plataforma PawsSafety para que ele seja visualizado por outros usuários interessados em adotar.*

**Descrição:**

* O usuário logado deve ter acesso à opção de listar um animal para adoção.
* Deve ser fornecido um formulário para inserção dos dados do animal, incluindo nome, espécie, raça, idade, descrição e fotos.
* O usuário deve ter a opção de salvar os dados do animal para que sejam exibidos publicamente na plataforma.

**Estória de Usuário 3:** *Como um usuário interessado em adotar um animal, desejo poder visualizar os detalhes dos animais disponíveis na plataforma PawsSafety e entrar em contato com o anunciante para obter mais informações.*

**Descrição:**

* O usuário deve ser capaz de navegar pela seção de adoção e visualizar a lista de animais disponíveis.
* Cada animal deve ter uma página com informações detalhadas, como nome, foto, raça, idade, gênero e descrição.
* O usuário deve ter a opção de entrar em contato diretamente com o anunciante através de um sistema de e-mail integrado na plataforma.
* O sistema de chat deve permitir uma comunicação direta e segura entre o interessado e o anunciante, facilitando o processo de adoção.

***Modelo CANVAS***

Gráfico de mapa de árvore

Descrição gerada automaticamente com confiança média

**2. Especificação das Regras de Negócios (RN) e dos Requisitos Funcionais (RF) e Não Funcionais (RNF)**

**Regra de Negócio**

|  |
| --- |
| **RN 001 –** **Segurança dos Dados** |
| **Descrição:** Todos os dados armazenados no banco de dados da plataforma, como informações de login, devem ser criptografados para garantir a segurança e privacidade. |

|  |
| --- |
| **RN 002 – Listar para adoção** |
| **Descrição:** Os usuários podem listar animais para adoção ou procurar animais disponíveis para adoção |

|  |
| --- |
| **RN 003 – Utilizar comunicação** |
| **Descrição:** Os usuários podem se comunicar durante a negociação através da plataforma por envio de e-mails. |

Modelo para descrever Regra de Negócio:

|  |
| --- |
| **RN 001 – Pedido Cancelado** |
| **Descrição:** Quando um pedido é cancelado este deve ser finalizado retornando a situação do Estoque |

|  |
| --- |
| **RN 002 – Entrega de Pedido** |
| **Descrição:** Enviar o pedido somente se o cliente possui endereço de entrega |

**Requisitos Funcionais**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **RF 001 – Cadastrar usuários:** | | Categoria: ( ) Oculto  (X) Evidente | | Prioridade: (X) Altíssima  ( ) Alta  ( ) Média  ( ) Baixa | | |
| **Descrição**: Os usuários devem ser capazes de criar uma conta com informações pessoais, como nome, e-mail e senha. Os usuários devem poder atualizar seu perfil com informações adicionais, como endereço e foto. | | | | | | |
| **Requisitos não funcionais** | | | | | | |
| Nome | Restrição | | Categoria | | Obrigatoriedade | Permanência |
| RNF 1.1 Controle de Dados | Os dados dos usuários, incluindo informações de pagamento, devem ser armazenados e transmitidos de forma segura. | | Segurança | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| RNF 1.2 Política de Privacidade | Deve haver uma política de privacidade clara e transparente, explicando como os dados dos usuários serão usados e protegidos. | | Privacidade | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| RNF 2.1 Recurso de Acessibilidade | O site deve ser acessível a pessoas com deficiência, seguindo padrões de acessibilidade web. | | Acessibilidade | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| RNF 2.2 Planejamento de Escalabilidade | A plataforma deve ser escalável para lidar com o crescimento de usuários e dados ao longo do tempo. | | Escalabilidade | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| RNF 2.3 Estratégia de Backup | Deve haver um plano de backup regular e um procedimento de recuperação de dados em caso de falha no sistema. | | Recuperação | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **RF 002 –** **Excluir animais listados:** | | Categoria: ( ) Oculto  (X) Evidente | | Prioridade: ( ) Altíssima  (X) Alta  ( ) Média  ( ) Baixa | | |
| **Descrição**: Os usuários devem ser capazes de realizar doações monetárias para organizações. Deve haver a opção para onde a doação será destinada, se aplicável. | | | | | | |
| **Requisitos não funcionais** | | | | | | |
| Nome | Restrição | | Categoria | | Obrigatoriedade | Permanência |
| RNF 1.1 Facilidade de uso | A opção de excluir animais listados deve ser intuitiva e facilmente acessível na interface do usuário. | | Usabilidade | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| RNF 1.2 Tempo de Resposta | O sistema deve realizar a exclusão do animal listado em no máximo 5 segundos. | | Desempenho | | (X) Desejável  ( ) Obrigatório | (X) Permanente  ( ) Transitório |
| RNF 1.3 Mensagens de Confirmação | O sistema deve fornecer uma mensagem de confirmação ao usuário antes de excluir o animal listado, evitando exclusões acidentais. | | Usabilidade | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **RF 003 – Adotar animais:** | | Categoria: ( ) Oculto  (X) Evidente | | Prioridade: (X) Altíssima  ( ) Alta  ( ) Média  ( ) Baixa | | |
| **Descrição**: Os usuários devem poder listar animais disponíveis para adoção, incluindo informações detalhadas sobre cada animal. Os usuários devem ser capazes de buscar animais para adoção com base em critérios como raça, idade e localização. | | | | | | |
| **Requisitos não funcionais** | | | | | | |
| Nome | Restrição | | Categoria | | Obrigatoriedade | Permanência |
| RNF 1.1 Regulamentos | O site deve cumprir todas as leis e regulamentos relacionados à proteção de dados, direitos dos animais e direitos dos usuários. | | Conformidade Legal | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| RNF 1.2 Política de Privacidade | Deve haver uma política de privacidade clara e transparente, explicando como os dados dos usuários serão usados e protegidos. | | Privacidade | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| RNF 1.3 Controle de Dados | Os dados dos usuários, incluindo informações de pagamento, devem ser armazenados e transmitidos de forma segura. | | Segurança | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **RF 004 – Utilizar bate-papo:** | | Categoria: ( ) Oculto  (X) Evidente | | Prioridade: ( ) Altíssima  (X) Alta  ( ) Média  ( ) Baixa | | |
| **Descrição**: Os usuários devem ter acesso a envio de e-mails para interagir com outros usuários da plataforma. | | | | | | |
| **Requisitos não funcionais** | | | | | | |
| Nome | Restrição | | Categoria | | Obrigatoriedade | Permanência |
| RNF 1.1 Controle de disponibilidade | A plataforma deve estar disponível 24 horas por dia, 7 dias por semana, com tempo de inatividade mínimo para manutenção programada. | | Disponibilidade | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| RNF 1.2 Políticas de desempenho | A plataforma deve ser responsiva e ter tempos de carregamento rápidos, mesmo com muitos acessos simultâneos. | | Desempenho | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| RNF 1.3 Planejamento de Escalabilidade | A plataforma deve ser escalável para lidar com o crescimento de usuários e dados ao longo do tempo. | | Escalabilidade | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |

Modelo para descrever Requisito Funcional e os Não Funcionais associados aos Funcionais:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **RF 001 – Registrar Empréstimos** | | Categoria: ( ) Oculto  (X) Evidente | | Prioridade: (X) Altíssima  ( ) Alta  ( ) Média  ( ) Baixa | | |
| **Descrição**: O sistema deve registrar empréstimos de filmes, indicando o cliente e os filmes que foram emprestados, bem como a data do empréstimo e valor previsto para pagamento na devolução | | | | | | |
| **Requisitos não funcionais** | | | | | | |
| Nome | Restrição | | Categoria | | Obrigatoriedade | Permanência |
| RNF 1.1 Controle de Acesso | A função só pode ser acessada por usuário com perfil de operador ou superior | | Segurança | | ( ) Desejável  (X) Obrigatório | (X) Permanente  ( ) Transitório |
| RNF 1.2 Identificação de Filmes | As fitas devem ser identificadas por um código de barras | | Interface | | (X) Desejável  ( ) Obrigatório | ( ) Permanente  (X) Transitório |
| RNF 1.3 Janela Única | Todas as funções relacionadas a empréstimos devem ser efetuadas em uma única janela | | Interface | | (X) Desejável  ( ) Obrigatório | ( X) Permanente  ( ) Transitório |

Modelo para descrever os Requisitos não funcionais que NÃO estão relacionados aos Funcionais:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requisitos não funcionais** | | | | |
| Nome | Restrição | Categoria | Obrigatoriedade | Permanência |
|  |  |  | ( ) Desejável  ( ) Obrigatório | ( ) Permanente  ( ) Transitório |
|  |  |  | ( ) Desejável  ( ) Obrigatório | ( ) Permanente  ( ) Transitório |
|  |  |  | ( ) Desejável  ( ) Obrigatório | ( ) Permanente  ( ) Transitório |

**Matriz de Rastreabilidade entre Requisitos Funcionais (RF) x Regras de Negócio (RN):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RF x RN** | RF001 | RF002 | RF003 | RF004 |
| RN001 |  | X |  |  |
| RN002 | X |  | X | X |
| RN003 | X |  | X | X |

**Matriz de Rastreabilidade entre Requisitos Funcionais (RF) x Requisitos Funcionais (RF):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RF x RF** | RF001 | RF002 | RF003 | RF004 |
| RF001 |  | X | X | X |
| RF002 | X |  |  |  |
| RF003 |  |  |  | X |
| RF004 |  |  | X |  |

**3. Modelagem**

**Índices de Casos de Uso**

Faça uma breve descrição de cada use case que foi identificada para o sistema.

Para cada Use Case identificar:

Código do Use Case (Ex.: UC001, UC002, UC003, ...).

Função do Use Case.

**Matriz de Rastreabilidade entre Requisitos Funcionais e Use Cases:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **RF x UC** | RF01 | RF02 | RF03 |  | RF... |
| UC01 | X |  |  |  |  |
| UC02 |  | X | X |  |  |
| UC03 | X |  |  |  |  |
| UC04 |  | X |  |  |  |

**Identificação do Atores do Sistema**

Faça uma breve descrição de como este ator interage com o sistema.

Exemplo: Este ator é uma *[pessoa ou um sistema ou um dispositivo]* que atua no sistema para ....

**Diagrama de Casos de Uso**

Inclua aqui o diagrama de use cases, construído a partir da ferramenta de modelagem. O diagrama deve incluir os use cases que irão implementar os requisitos do sistema e seus atores.

|  |  |
| --- | --- |
| **Caso de Uso – Listar animal para adoção** | |
| **ID** | UC 001 |
| **Descrição** | Este caso de uso permite o usuário listar animais para adoção |
| **Ator Primário** | Usuário do sistema |
| **Pré-condição** | Ter cadastro na plataforma |
| **Cenário Principal** | 1. O use case inicia quando o usuário efetua o login na plataforma 2. O sistema carrega a página inicial do site 3. O usuário seleciona a opção: “Listar animal para adoção” 4. O sistema retorna uma espécie de formulário para preencher os dados do animal 5. O usuário insere os dados do animal e clica em: “enviar” 6. O sistema recebe os dados 7. O sistema insere o animal na página de “Animais para adoção” 8. O usuário pode confirmar em seu perfil se o animal foi corretamente listado 9. O sistema retorna o cadastro do animal para o usuário; |
| **Pós-condição** | O sistema já possui o animal listado e permite que o mesmo seja visualizado por outros usuários |
| **Cenário Alternativo** | 6a – O usuário quer editar os dados do animal  7a. - O usuário clica em “Editar Animal” na aba “Meus Animais” no seu perfil  8a – O sistema permite a alteração exibindo novamente o formulário com os últimos dados do animal  9a. O usuário atualiza os dados e clica em “enviar”  10a – O sistema recebe a alteração e atualiza os dados do animal |

|  |  |
| --- | --- |
| **Caso de Uso – Excluir animais listados** | |
| **ID** | UC 002 |
| **Descrição** | Este caso de uso permite o usuário excluir animais listados |
| **Ator Primário** | Usuário do sistema |
| **Pré-condição** |  |
| **Cenário Principal** |  |
| **Pós-condição** |  |
| **Cenário Alternativo** |  |

|  |  |
| --- | --- |
| **Caso de Uso – Adotar um animal** | |
| **ID** | UC 003 |
| **Descrição** | Este caso de uso permite o usuário adotar animais |
| **Ator Primário** | Usuário do sistema |
| **Pré-condição** |  |
| **Cenário Principal** |  |
| **Pós-condição** |  |
| **Cenário Alternativo** |  |

Diagrama

Descrição gerada automaticamente

**Especificação dos Use Cases**

Exemplo:

|  |  |
| --- | --- |
| **Caso de Uso – Cadastrar Cliente** | |
| **ID** | UC 001 |
| **Descrição** | Este caso de uso tem por objetivo cadastrar o cliente ... |
| **Ator Primário** | Atendente |
| **Pré-condição** | Não possui |
| **Cenário Principal** | 1. O use case inicia quando o atendente seleciona a opção de cadastrar cliente 2. O sistema carrega o formulário para cadastro de cliente 3. O atendente informa o cpf do cliente a ser cadastrado 4. Inclua UC 002 – Buscar Cliente 5. O atendente informa os demais dados do cliente como: nome, RG, data de nascimento, ... 6. O atendente confirma o cadastro do cliente 7. O sistema recebe e valida os dados do cliente 8. O sistema confirma o cadastramento do cliente 9. O sistema encerra a operação |
| **Pós-condição** | Não possui |
| **Cenário Alternativo** | \*a – Em qualquer momento o atendente pode sair do sistema  4a – CPF já se encontra cadastrado  4a.1 O sistema mostra mensagem informando já existe cpf cadastrado  4a.2 O sistema retorna ao passo 3 do fluxo principal  7a – Data de Nascimento invalida  7a.1 O sistema mostra mensagem explicativa informando a inconsistência da data de nascimento informada  7a.2 O sistema retorna ao passo 5 do fluxo principal |
| **Inclusão** | UC 002 – Buscar Cliente |
| **Extensão** |  |

**Classes de Análise para os casos de uso**

Você deverá detalhar as classes de análise encontradas para o use case, caso uma use case utilize uma classe de outra descrita antes deve-se relacioná-la da seguinte forma:

*“Classe de [Tipo da Classe] [Nome da Classe] descrita na use case [Nome da use case]”*

#### Classe de Fronteira [Nome da Classe]

Descreva a responsabilidade da classe e, se for o caso, a qual interface esta relacionada. Se esta classe se relaciona com outros sistemas através de um protocolo descreva o mais detalhado possível este protocolo. Faça uma descrição para cada classe de fronteira.

#### Classe de Entidade [Nome da Classe]

Descreva a responsabilidade da classe e quais as informações que são pertinentes a esta classe. Faça uma descrição para cada classe de entidade.

#### Classe de Controle [Nome da Classe]

Descreva a responsabilidade da classe, a seqüência de controle (se necessário faça um diagrama de atividades) e os comportamentos relacionados a negócio. Faça uma descrição para cada classe de controle.

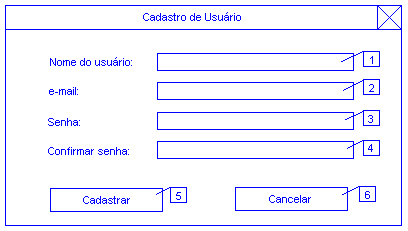
#### Diagrama de Classes de Análise

Coloque o diagrama de relacionamento entre as classes de análise para esta Use Case.

**Interfaces ( Lay-out de Tela)**

Uma interface é uma descrição lógica e conceitual de como uma ou mais use cases são providas pela interface do usuário, se for o caso, incluindo a interação requerida entre o(s) ator(es) e o sistema. Em geral janelas representam as interfaces necessárias para entender do ponto de vista macro os requisitos da interface do usuário.

Faça o desenho das interfaces gráficas referenciando os campos com etiquetas como no exemplo abaixo.



Descreva os campos da interface gráfica

1. Campo para a entrada e visualização do nome do usuário
2. Campo para a entrada e visualização do email do usuário
3. etc.

**Diagrama de Seqüência para os casos de uso**

Colocar os modelos de seqüência para cada caso de uso (curso normal e alternativo). Os modelos devem ser apresentados de forma legível.

1. **Projeto**

Texto geral sobre a plataforma adotada para o projeto. Sistema Operacional. Servidores de banco de dados, aplicações (tomcat), web. Linguagens de programação. Frameworks, padrões, componentes, e outras decisões de projeto.

**Diagrama de Classes de Projeto**

Apresentar os diagramas de classe de projeto que foram desenvolvidos a partir, principalmente, dos diagramas de classes de análise e dos diagramas de seqüência mostrando todos os relacionamentos entre as classes e as operações mais importantes (não é necessário que todas as operações ou métodos e atributos sejam mostrados, não se deve poluir o diagrama)

**Modelo de Arquitetura**

Apresentar um diagrama que mostre a arquitetura do sistema. Esse diagrama deve ter os componentes maiores do sistema, podendo ser apresentados com pacotes ou de componentes da UML. Esse diagrama é útil para verificar se a arquitetura MVC foi adequadamente seguida.

Exemplo:

Modelo de Arquitetura do caso de uso Cadastrar Carro

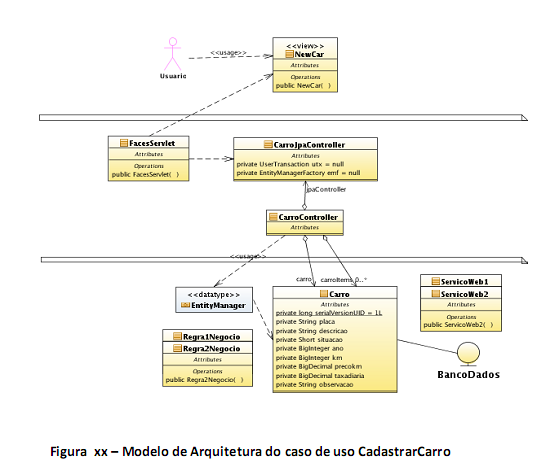
(Explicar o modelo, suas classes e a arquitetura adotada.)

A Figura xx apresenta o modelo de arquitetura do caso de uso Cadastrar Carro. Conforme

mostra a Figura a arquitetura segue o padrão MVC, com apoio do framework JSF (Java Server Face).

Observação: O modelo pode ser melhorado incluindo também os principais pacotes com os

frameworks (suas principais classes) e padrões de projeto.



**Modelo físico do banco de dados**

Apresentar o modelo MER usado para construção do banco de dados, construído a partir do modelo de classes do sistema. Apresentar os scripts SQL usados para construir o banco de dados.

**Diagrama de Implantação**

Diagrama de implantação da UML para a visualização de como cada nó de hardware se

comunica com os outros nós.