**Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - *Campus* Capivari**

**SISTEMA DE CONTROLE DE PORTARIA**

**CAPIVARI/SP**

**2015**

**Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - *Campus* Capivari**

**Documento de um Produto de Software**

**Versão 1.0**

Trabalho elaborado pelos alunos do Curso Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, *Campus* Capivari.

Alunos:

Beatriz Regina de Souza

Jônatas Bueno do Livramento

Michele Bragante Frois

Professor Orientador:

Ricardo Figueredo

**CAPIVARI/SP**

**2015**

Sumário

RESUMO iv

1 INTRODUÇÃO 5

1.1 JUSTIFICATIVA 5

1.2 OBJETIVO 5

2 LEVANTAMENTO DE DADOS 6

2.1 QUETIONÁRIO 6

3 REQUISITOS DO SISTEMA 8

3.1 REQUISITOS FUNCIONAIS 8

3.2 REQUISITOS NÃO- FUNCIONAIS 9

3.3 PLATAFORMA NECESSÁRIA 9

4 ANÁLISE DO PROJETO 10

4.1 DIAGRAMA DE CASO DE USO 10

4.1.1 ESPECIFICAÇÃO DO CASO DE USO 11

4.2 DIAGRAMA DE CLASSES 21

4.3 DIAGRAMA DE SEQUÊNCIA 25

4.4 DIAGRAMA DE ATIVIDADES 30

4.5 DIAGRAMA DE MÁQUINA DE ESTADOS 35

4.6 MODELO DE DADOS 37

4.6.1 MODELO DE ENTIDADE E RELACIONAMENTO 37

4.6.2 CRIAÇÃO FÍSICA DA BASE DE DADOS 38

4.6.3 DICIONÁRIO DE DADOS 45

4.6.4 SISTEMAS E COMPONENTES EXTERNOS UTILIZADOS 50

5 CRONOGRAMA 51

5.1 CRONOGRAMA PROPOSTO 51

5.2 CRONOGRAMA REALIZADO 52

6 ORÇAMENTO 54

6.1 SOFTWARES NECESSÁRIOS 54

7 PLANO DE TESTE 55

7.1 IDENTIFICAÇÃO DO PROJETO 55

7.2 REQUISITOS A TESTAR 55

7.2.1 TESTE DO BANCO DE DADOS 55

7.2.2 TESTE FUNCIONAL 56

7.2.3 TESTE DA INTERFACE DO USUÁRIO 56

7.2.4 TESTE DE SEGURANÇA E DE CONTROLE DE ACESSO 56

7.2.5 TESTE DE INSTALAÇÃO 57

7.3 ESTRATÉGIA DE TESTE 57

7.3.1 TESTE DE INTEGRIDADE DE DADOS E DO BANCO DE DADOS 57

7.3.2 TESTE DE FUNÇÃO 58

7.3.3 TESTE DA INTERFACE DO USUÁRIO 58

7.3.4 TESTE DE SEGURANÇA E CONTROLE DE ACESSO 58

7.3.5 TESTE DE INSTALAÇÃO 59

8 IMPLANTAÇÃO 60

9 CONCLUSÃO 61

10 GLOSSÁRIO 62

REFERÊNCIAS 63

Resumo

NOME DO TRABALHO

**Objetivo:**. **Material e Método:** . **Resultados:** . **Conclusão**:

**Descritores:**

# Introdução

O Sistema de Controle de Portaria – IFSP Capivari (SCP-IFSP) consiste em um sistema de apoio ao fluxo nas dependências do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – *Campus* Capivari. Esse apoio consiste no melhoramento de processos que são feitos manualmente e a criação de processos ainda não existentes.

Tais rotinas de controle, consistem em processos da entrada e saída de terceirizados, alunos, fornecedores, funcionários e visitantes em um único sistema.

## justificativa

Nota-se um grande crescimento no *Campus* Capivari, com passar do tempo as necessidades quanto ao estacionamento são de encontrar vagas, garantir a vaga dos funcionários, a quantidade reservada para visitantes no instituto, localização do dono do veículo e entre outros estão cada vez maiores. Outro ponto notório é o controle de saída de alunos menores de idade, com o Ensino Médio integrado totalmente ao *Campus*, o fluxo de adolescentes aumentou muito.

Evidentemente, faz-se necessário um sistema para organizar e otimizar esses processos buscando garantir maior segurança para os que frequentam o prédio.

## objetivo

Dentre todos os objetivos de valor de agregação acadêmica, este trabalho objetiva:

1. Aumentar a segurança do *campus*, evitando a saída de alunos menores, antes do horário previsto, sem autorização e sem acompanhamento de um responsável;
2. Organizar as vagas do estacionamento;
3. Otimizar a autenticação dos membros do *campus*, já na entrada do prédio;
4. Auxiliar o trabalho dos vigilantes;
5. Facilitar o arquivo de documentos.

# Levantamento de dados

## Quetionário

**1. Qual será o meio de identificação dos docentes e discentes?**

**R:** *Aluno carterinha, funcionários crachá.*

**2. A faixa etária do aluno será identificada por meio do crachá?**

**R:** *Sim, alunos menores terão a cor da carteirinha diferenciada.*

Carteirinha, autorização de saída. Maior não precisa de nada

**3. As pessoas responsáveis pelo processo de saída do aluno têm acesso aos computadores com acesso à internet?**

**R:** *Sim, ainda está em implantação.*

**4. Qual o número de vagas disponibilizadas para os servidores públicos?**

**R:** *São as vagas fixas de frente para o campus.*

**5. Será necessário utilizar fotos, se sim por que e como?**

**R:** *Não. Autenticação por telefone.*

**6. No caso de outros responsáveis não cadastrados como responsáveis, vierem buscar alunos menores de idade, qual será o procedimento?**

**R:** *Comunicar os responsáveis por telefone e verificar se a pessoa que veio buscar está autorizada.*

**7. O cadastro do corpo discente e docente será realizado por quem?**

**R:** *Administração.*

**8. Quais pessoas acessarão o sistema?**

**R:** *Administração, sócio pedagógico, secretaria e vigilantes.*

**9. Esqueceu a identificação rápida.**

**R:** *Aluno mostra RG e registra no livro, com ass e motivo. Professor mostra RG.*

**Alguns adentros:**

* + Aluno realiza cadastro na administração para estacionar, recebe um crachá temporário para identificar o carro = identificação do aluno pela carteirinha.
  + Funcionários = Cracha fixo para o carro, vaga fixa de frente para o campus. Cor do crachá verde.
  + Visitante = Crachá devolvido no fim da visita. Cor vermelho.
  + Tercerizado = Crachá fixo. Cor azul.
  + *5 grupos: tercerizados, alunos, fornecedores, funcionários e visitantes.*
  + Servidores chachá e identificação do carro.
  + Visita - identificação fornecendo os damos e motivos da visita. Dados anexo3, é fornecido crachá de visitante e informado o horario para visitar o setor desejado.
  + Fornecedor identifica, fornece os dados e motivação da visita, fora do horário fixo não entra para descarregar, mas se quiser visitar entra como visitante.
  + Fornecer comunicados aos vigilantes. Entradas aos, sábados e domingos e fora do horário do funcionamento.
  + Cadastro com foto na secretaria.
  + Contato: Gilberto coordenador adm. processos de estacionamento.

# rEQUISITOS DO SISTEMA

## rEQUISITOS FUNCIONAIS

Os requisitos funcionais especificam ações que o sistema é capaz de executar. Nesse sistema tem-se as seguintes funções:

* Manter usuário do sistema (CRUD, liberar permissões);
* Manter dados do Aluno (CRUD, importar dados de alunos do software NAMBEI);
* Manter dados do Professor (CRUD);
* Manter dados do Terceirizado (CRUD);
* Manter dados do Fornecedor (CRUD);
* Manter dados do Visitante (CRUD);
* Manter dados do Porteiro (CRUD);
* Manter dados do Funcionário do IFSP (CRUD);
* Solicitar vaga para estacionamento (sendo que os privilégios são dos servidores públicos);
* Solicitar autorização de saída para aluno menor de idade (processo qual o aluno solicita saída para a assistente estudantil, ela verifica com os pais através de um telefonema e/ou e-mail se o aluno pode sair. Se aprovado, ela informa no sistema e irá aparecer para o porteiro também, assim o aluno poderá ser liberado na portaria);
* Verificar a quantidade de carros estacionados no âmbito escolar, uma vez que, será delimitado no sistema a quantidade de vagas disponíveis;
* Registrar permanência de veículos (quando um veículo apresentar problemas e necessitar ficar no *Campus* fora do horário de funcionamento do IFSP);
* Registrar Visitas;
* Manter Registro de saída e entrada de funcionários terceirizados;
* Registrar fornecedores para carga/descarga;
* Gerenciar o fluxo do estacionamento em dias, baseado na agenda (horário) dos servidores públicos;
* Gerar relatórios mensais sobre o controle/gerenciamento da portaria.

## requisitos não- funcionais

Este item apresenta os requisitos não-funcionais do sistema de Controle de Acesso, que especificam restrições sobre os serviços ou funções providas do sistema.

* O sistema será implementado direcionado a plataforma Desktop.

## Plataforma Necessária

Para que o sistema funcione é preciso ter os requisitos a seguir:

* Servidor MySql;
* Computadores desktop;
* Rede Local.

# Análise do projeto

Para a criação do sistema ser consistente, foi criado variados diagramas do sistema, que serão exibidos e explicados ao longo do capitulo.

## Diagrama de caso de uso

Através deste, é possível compreender o comportamento do sistema em nível de usuário. A figura a seguir, mostra uma visão ampla das funcionalidades que o Sistema de Controle de Portaria oferece.

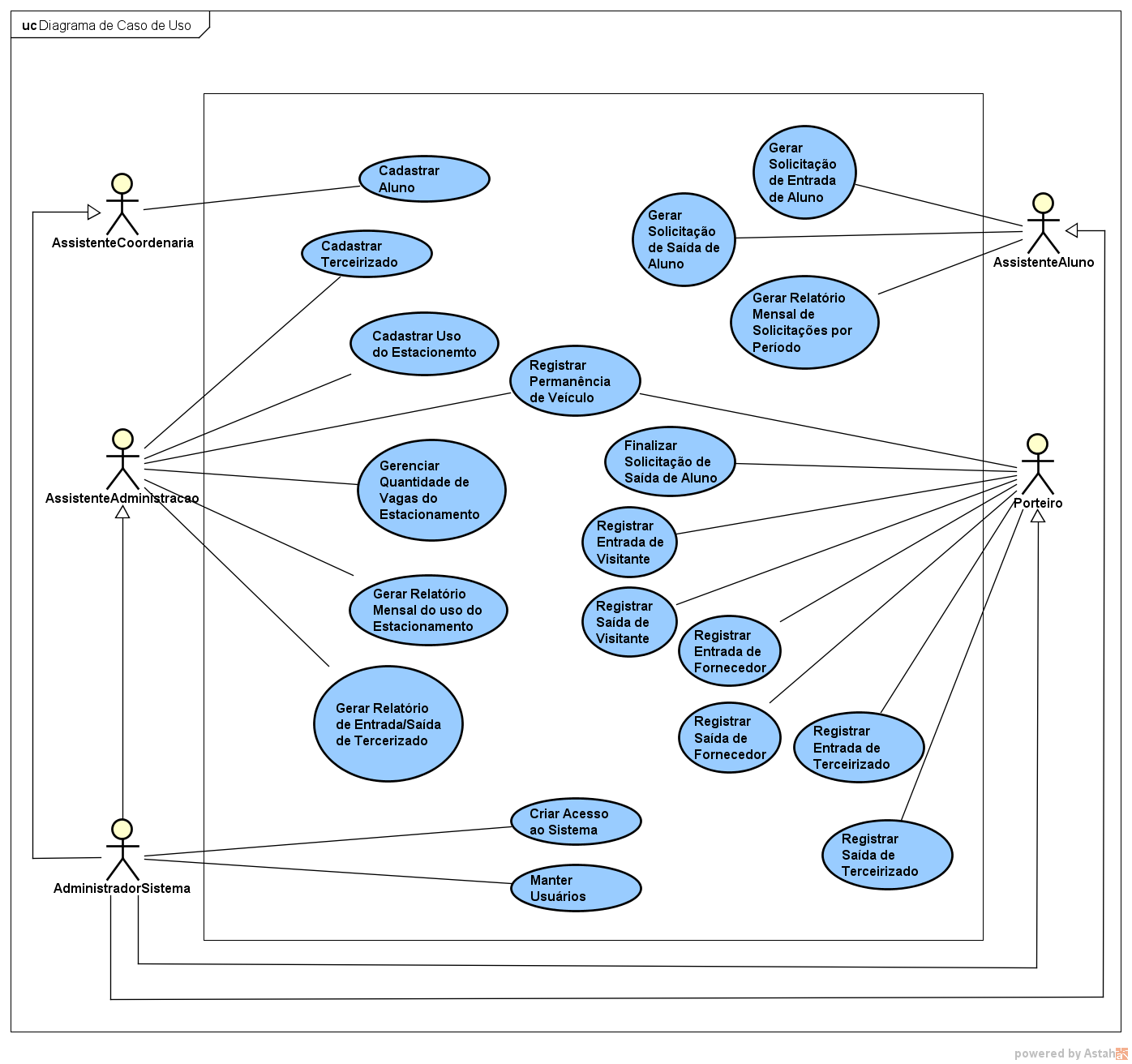


Figura 1 - Diagrama De Caso De Uso.

Um dos itens principais do Diagrama de Caso de Uso são os atores. Eles representam os papéis desempenhados pelos diversos usuários do sistema. A seguir, uma breve descrição de cada ator deste diagrama.

* **Assistente de Administração**

É o funcionário que trabalha de fato na área administrativa do *Campus* que terá acesso ao sistema. Ele é responsável pelas funcionalidades burocráticas do sistema, bem como cadastrar o uso do estacionamento, cadastrar funcionário terceirizado, gerar relatórios entre outros.

* **Administrador do Sistema**

Usuário máster do sistema. Responsável por incluir, visualizar, alterar e excluir usuários e/ou permissões de acesso e, responsável por manter o sistema em funcionamento.

* **Assistente de Alunos**

Funcionário responsável por realizar solicitação de entrada e saída dos alunos fora do horário de aula e verificar a segurança do aluno dentro do instituto.

* **Assistente de Coordenadoria**

Funcionário responsável pelo cadastro de alunos.

* **Porteiro**

Funcionário responsável pela portaria do *Campus*. Controla o fluxo de entrada e saída de pedestres e/ou de veículos.

### ESPECIFICAÇÃO DO CASO DE USO

A especificação de um caso de uso tem como objetivo, descrever em uma linguagem simples, informações como a função em linhas gerais do caso de uso, quais atores participam com ele, quais etapas devem ser executadas pelo ator e pelo sistema para que o caso de uso execute sua função, quais parâmetros devem ser fornecidos e quais restrições e validações o caso de uso deve ter. A seguir, as especificações de casos de uso do sistema.

**Documentação do caso de uso: Gerar Solicitação de Entrada de Aluno**

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Gerar Solicitação de Entrada de Aluno** |
| **Caso de Uso Geral** |  |
| **Ator (es) Principal (is)** | Assistente de Alunos |
| **Ator (es) Secundário (s)** |  |
| **Resumo** | Este caso de uso descreve as etapas percorridas para registro de entrada atrasada dos alunos. |
| **Pré-Condições** | Ser aluno regular no IFSP. |
| **Pós-Condições** |  |
| **Fluxo Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Pesquisar aluno |  |
|  | 1. Validar dados do aluno (Prontuário, Nome, RG) |
| 1. Gerar solicitação de entrada de aluno |  |
|  | 1. Registrar solicitação de entrada de aluno |
| **Restrições/Validações** | Apenas para casos quando se tem a necessidade de entrar para a sala de aula fora do horário regular. |

**Documentação do caso de uso: Gerar Solicitação de Saída de Aluno**

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Gerar Solicitação de Saída de Aluno** |
| **Caso de Uso Geral** |  |
| **Ator (es) Principal (is)** | Assistente de Alunos |
| **Ator (es) Secundário (s)** |  |
| **Resumo** | Este caso de uso descreve as etapas percorridas para registro de saída antecipada dos alunos. |
| **Pré-Condições** | Ser aluno regular no IFSP. |
| **Pós-Condições** |  |
| **Fluxo Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Pesquisar aluno |  |
|  | 1. Validar dados do aluno (Prontuário, Nome, RG) |
| 1. Gerar solicitação de saída de aluno |  |
|  | 1. Registrar solicitação de saída de aluno |
| **Restrições/Validações** | Apenas para casos quando se tem a necessidade de sair do Instituto fora do horário das aulas. |
| **Fluxo Alternativo – Solicitação de saída antecipada para aluno com idade inferior a 18 anos** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Pesquisar aluno |  |
|  | 1. Validar dados do aluno (Prontuário, Nome, RG) |
| 1. Entrar em contato com o responsável para pedir autorização |  |
| 1. Autorização concedida, gerar pedido de liberação de saída para aluno |  |
|  | 1. Registrar solicitação de saída do aluno |
| **Fluxo Alternativo – Autorização não concedida pelo responsável para saída antecipada de aluno com idade inferior a 18 anos** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Pesquisar aluno |  |
|  | 1. Validar dados do aluno (Prontuário, Nome, RG) |
| 1. Entrar em contato com o responsável para pedir autorização |  |
| 1. Autorização negada |  |
|  | 1. Cancelar solicitação de saída de aluno |

**Documentação do caso de uso: Gerar Relatório Mensal de Solicitações por Período**

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | **Gerar Relatório Mensal de Solicitações por Período** |
| **Caso de Uso Geral** |  |
| **Ator (es) Principal (is)** | Assistente de Alunos |
| **Ator (es) Secundário (s)** |  |
| **Resumo** | Este caso de uso descreve as etapas percorridas para a geração de relatório mensal de alunos que entraram ou saíram mais cedo ou mais tarde. |
| **Pré-Condições** | Existir histórico de solicitações de entrada atrasada ou saída antecipada de aluno |
| **Pós-Condições** | Relatório gerado com sucesso. |
| **Fluxo Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Informar mês, ano e período para gerar relatório mensal |  |
|  | 1. Validar campos preenchidos corretamente |
|  | 1. Gerar relatório |
|  | 1. Relatório gerado com sucesso. |
| **Restrições/Validações** | Apenas relatório mensal. |

**Documentação do caso de uso: Cadastrar Uso do Estacionamento**

|  |  |
| --- | --- |
| **Nome do caso de uso** | **Cadastrar Uso do Estacionamento** |
| **Caso de uso geral** |  |
| **Ator (es) Principal (is)** | Assistente em Administração |
| **Ator (es) Secundário (s)** | Todos |
| **Resumo** | Este caso de uso descreve as etapas para cadastro dos usuários que utilizam o estacionamento. |
| **Pré-Condições** | Estar cadastrado como membro do IFSP. |
| **Pós-Condições** | Vaga concedida |
| **Fluxo Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Inserir dados no sistema |  |
|  | 1. Validar campos preenchidos corretamente |
|  | 1. Gerar código de identificação do veículo |
| 1. Entregar credencial ao membro para uso do estacionamento |  |
| **Restrições/Validações** | Uso do estacionamento apenas em horário registrado no cadastro. |

**Documentação do caso de uso: Gerenciar Quantidade de Vagas do Estacionamento**

|  |  |
| --- | --- |
| **Nome do caso de uso** | **Gerenciar Quantidade de Vagas do Estacionamento** |
| **Caso de uso geral** |  |
| **Ator (es) Principal (is)** | Assistente em Administração |
| **Ator (es) Secundário (s)** | Todos |
| **Resumo** | Este caso de uso descreve as etapas percorridas para determinar a quantidade de vagas delimitadas ao estacionamento. |
| **Pré-Condições** | Ter vagas no estacionamento |
| **Pós-Condições** | Não há |
| **Fluxo Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
|  |  |
| 1. Inserir a quantidade de vagas do estacionamento |  |
|  | 1. Salvar quantidade de vagas. |
| **Restrições/Validações** | Não ter estacionamento |

**Documentação do caso de uso: Gerar Relatório Mensal do uso do Estacionamento**

|  |  |
| --- | --- |
| **Nome do caso de uso** | **Gerar Relatório Mensal do uso do Estacionamento** |
| **Caso de uso geral** |  |
| **Ator (es) Principal (is)** | Assistente em Administração |
| **Ator (es) Secundário (s)** |  |
| **Resumo** | Este caso de uso descreve as etapas percorridas para gerar relatório mensal do uso do estacionamento. |
| **Pré-Condições** | Existir usuários cadastrados para uso do estacionamento. |
| **Pós-Condições** |  |
| **Fluxo Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Informar mês, ano e período para gerar relatório |  |
|  | 1. Validar campos preenchidos corretamente |
|  | 1. Gerar relatório |
|  | 1. Relatório gerado com sucesso. |
| **Restrições/Validações** | Apenas relatório mensal. |

**Documentação do caso de uso: Gerar Relatório Mensal de Entrada/Saída de Terceirizado**

|  |  |
| --- | --- |
| **Nome do caso de uso** | **Gerar Relatório Mensal de Entrada/Saída de Terceirizado** |
| **Caso de uso geral** |  |
| **Ator (es) Principal (is)** | Assistente em Administração |
| **Ator (es) Secundário (s)** |  |
| **Resumo** | Este caso de uso descreve as etapas percorridas para gerar relatório mensal do registro de entrada e saída de terceirizado. |
| **Pré-Condições** | Existir funcionário cadastrado para registrar entrada e saída do mesmo. |
| **Pós-Condições** | Existir histórico de registro do funcionário. |
| **Fluxo Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Informar funcionário e mês para gerar relatório |  |
|  | 1. Validar campos preenchidos corretamente |
|  | 1. Gerar relatório |
|  | 1. Relatório gerado com sucesso. |
| **Restrições/Validações** | Apenas relatório mensal. |

**Documentação do caso de uso: Registrar Permanência de Veículo**

|  |  |  |
| --- | --- | --- |
| **Nome do caso de uso** | | **Registrar Permanência de Veículo** |
| **Caso de uso geral** | |  |
| **Ator (es) Principal (is)** | | Assistente em Administração |
| **Ator (es) Secundário (s)** | | Porteiro |
| **Resumo** | | Este caso de uso descreve as etapas percorridas para registrar permanência de veículo no estacionamento do *Campus*. |
| **Pré-Condições** | | Ter acesso ao estacionamento do IF (possuir credencial) |
| **Pós-Condições** | |  |
| **Fluxo Principal** | | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. Pesquisar membro do IFSP. | |  |
|  | | 1. Validar campos corretamente |
| 1. Entrar em contato com o diretor do *Campus* para pedir autorização de permanência do veículo | |  |
| 1. Autorização concedida, registrar solicitação de permanência do veículo | |  |
|  | | 1. Gerar solicitação de permanência do veículo 2. Salvar em formato PDF |
| 1. Imprimir solicitação para assinatura do diretor do *Campus*. | |  |
| **Restrições/Validações** | | Quando a necessidade de o veículo permanecer no estacionamento fora do horário previsto. |
| **Fluxo Alternativo – Autorização não concedida pelo diretor do *Campus*** | | |
| **Ações do Ator** | **Ações do Sistema** | |
|  | 1. Cancelar solicitação de permanência de veículo no estacionamento | |

**Documentação do caso de uso: Registrar Entrada de Visitante**

|  |  |
| --- | --- |
| **Nome do caso de uso** | **Registrar Entrada de Visitante** |
| **Caso de uso geral** |  |
| **Ator (es) Principal (is)** | Porteiro |
| **Ator (es) Secundário (s)** |  |
| **Resumo** | Este caso de uso descreve as etapas percorridas para registro de visitante ao chegar na portaria do IFSP. |
| **Pré-Condições** | Identificar-se na portaria corretamente. |
| **Pós-Condições** | Pessoa ou departamento responsável por receber visitante estar presente no *Campus*. |
| **Fluxo Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Pedir identificação ao visitante |  |
| 1. Inserir dados obrigatórios no sistema |  |
|  | 1. Validar campos preenchidos corretamente |
|  | 1. Liberar vaga ou não para visitante |
| 1. Entregar crachá para identificação (uso interno) |  |
| 1. Avisar pessoa/departamento responsável por receber o visitante |  |
| 1. Informar ao visitante os horários de funcionamento do setor de atendimento ao público |  |
| **Restrições/Validações** | Visita apenas acontecerá se estiver em horário regular para atendimento ao público. |

**Documentação do caso de uso: Registrar Saída de Visitante**

|  |  |
| --- | --- |
| **Nome do caso de uso** | **Registrar Saída de Visitante** |
| **Caso de uso geral** |  |
| **Ator (es) Principal (is)** | Porteiro |
| **Ator (es) Secundário (s)** |  |
| **Resumo** | Este caso de uso descreve as etapas percorridas para registrar saída do visitante do IFSP. |
| **Pré-Condições** | Identificar-se na portaria corretamente. |
| **Pós-Condições** | Devolver crachá na portaria. |
| **Fluxo Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Pesquisar registro de entrada do visitante |  |
|  | 1. Finalizar visita |
| 1. Abrir portão para visitante sair do prédio. |  |
| **Restrições/Validações** | Não há. |

**Documentação do caso de uso: Registrar Entrada de Fornecedor**

|  |  |
| --- | --- |
| **Nome do caso de uso** | **Registrar Entrada de Fornecedor** |
| **Caso de uso geral** |  |
| **Ator (es) Principal (is)** | Porteiro |
| **Ator (es) Secundário (s)** |  |
| **Resumo** | Este caso de uso descreve as etapas percorridas para registro de fornecedor ao chegar na portaria do IFSP. |
| **Pré-Condições** | Identificar-se na portaria corretamente. |
| **Pós-Condições** | Pessoa ou departamento responsável por receber fornecedor estar presente no *Campus*. |
| **Fluxo Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Pedir identificação ao fornecedor |  |
| 1. Inserir dados obrigatórios no sistema |  |
|  | 1. Validar campos preenchidos corretamente |
|  | 1. Liberar vaga ou não para fornecedor |
| 1. Entregar crachá para identificação (uso interno) |  |
| 1. Avisar pessoa/departamento responsável por receber o fornecedor |  |
| 1. Informar ao fornecedor os horários de funcionamento do setor de atendimento ao público |  |
| **Restrições/Validações** | Entrega apenas acontecerá se estiver em horário regular para atendimento ao público. |

**Documentação do caso de uso: Registrar Saída de Fornecedor**

|  |  |
| --- | --- |
| **Nome do caso de uso** | **Registrar Saída de Fornecedor** |
| **Caso de uso geral** |  |
| **Ator (es) Principal (is)** | Porteiro |
| **Ator (es) Secundário (s)** |  |
| **Resumo** | Este caso de uso descreve as etapas percorridas para registrar saída do fornecedor do IFSP. |
| **Pré-Condições** | Identificar-se na portaria corretamente. |
| **Pós-Condições** | Devolver crachá na portaria. |
| **Fluxo Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Pesquisar registro de entrada do fornecedor |  |
|  | 1. Finalizar visita |
| 1. Abrir portão para o fornecedor sair do prédio. |  |
| **Restrições/Validações** | Não há. |

**Documentação do caso de uso: Finalizar Solicitação de Saída de Aluno**

|  |  |  |
| --- | --- | --- |
| **Nome do caso de uso** | | **Finalizar Solicitação de Saída de Aluno** |
| **Caso de uso geral** | |  |
| **Ator (es) Principal (is)** | | Porteiro |
| **Ator (es) Secundário (s)** | |  |
| **Resumo** | | Este caso de uso descreva as etapas percorridas para finalizar a solicitação de saída antecipada de aluno. |
| **Pré-Condições** | | Solicitação de saída antecipada de aluno com idade inferior a 18 anos realizada com sucesso. |
| **Pós-Condições** | | Aluno dirigir-se a portaria do *Campus* para sua saída. |
| **Fluxo Principal** | | |
| **Ações do Ator** | | **Ações do Sistema** |
| 1. Identificar aluno | |  |
| 1. Verificar se existe solicitação gerada em nome do aluno | |  |
| 1. Registrar/confirmar saída de aluno | |  |
|  | | 1. Finalizar solicitação de saída de aluno |
| **Restrições/Validações** | | Para finalizar uma solicitação a mesma deve existir |
| **Fluxo Alternativo – Não Confirmar Solicitação de Saída** | | |
| **Ações do Ator** | **Ações do Sistema** | |
| 1. Não confirmar a existência da solicitação de saída de aluno |  | |

## Diagrama de classes

Esses diagramas têm como principal objetivo a visualização das classes que compõem o sistema, com seus respectivos atributos e métodos, bem como em demonstrar como as classes dos diagramas se relacionam, complementam e transmitem informações entre si. As figuras a seguir, mostram os diagramas de classes que compõem este documento de software.

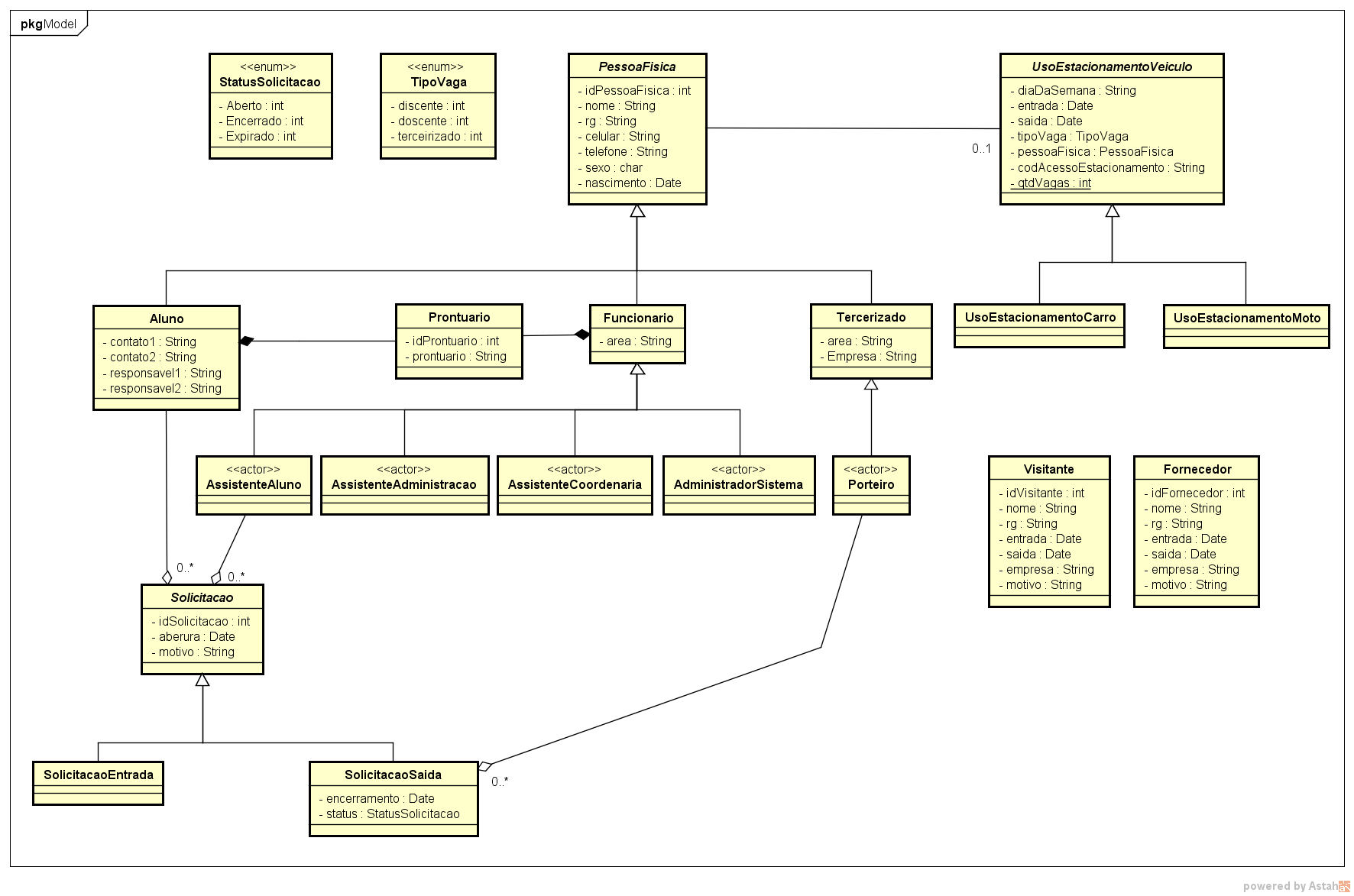


Figura 2 - DC - Representa a Camada de Modelo do Sistema.

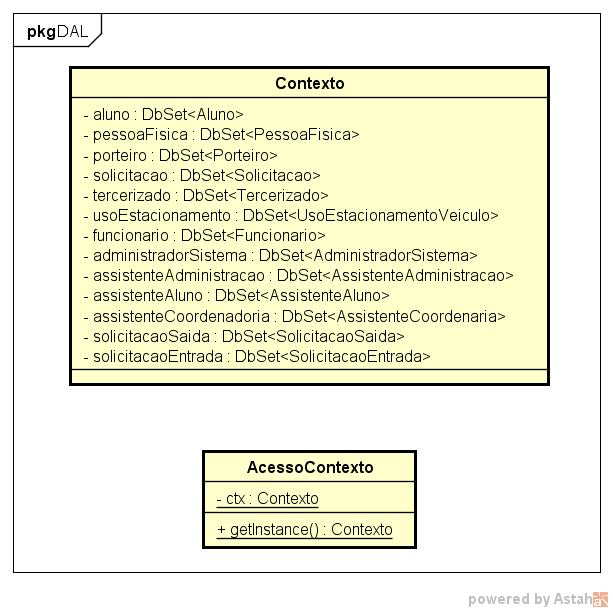


Figura 3 - DC - Representa a Camada DAL do Sistema.

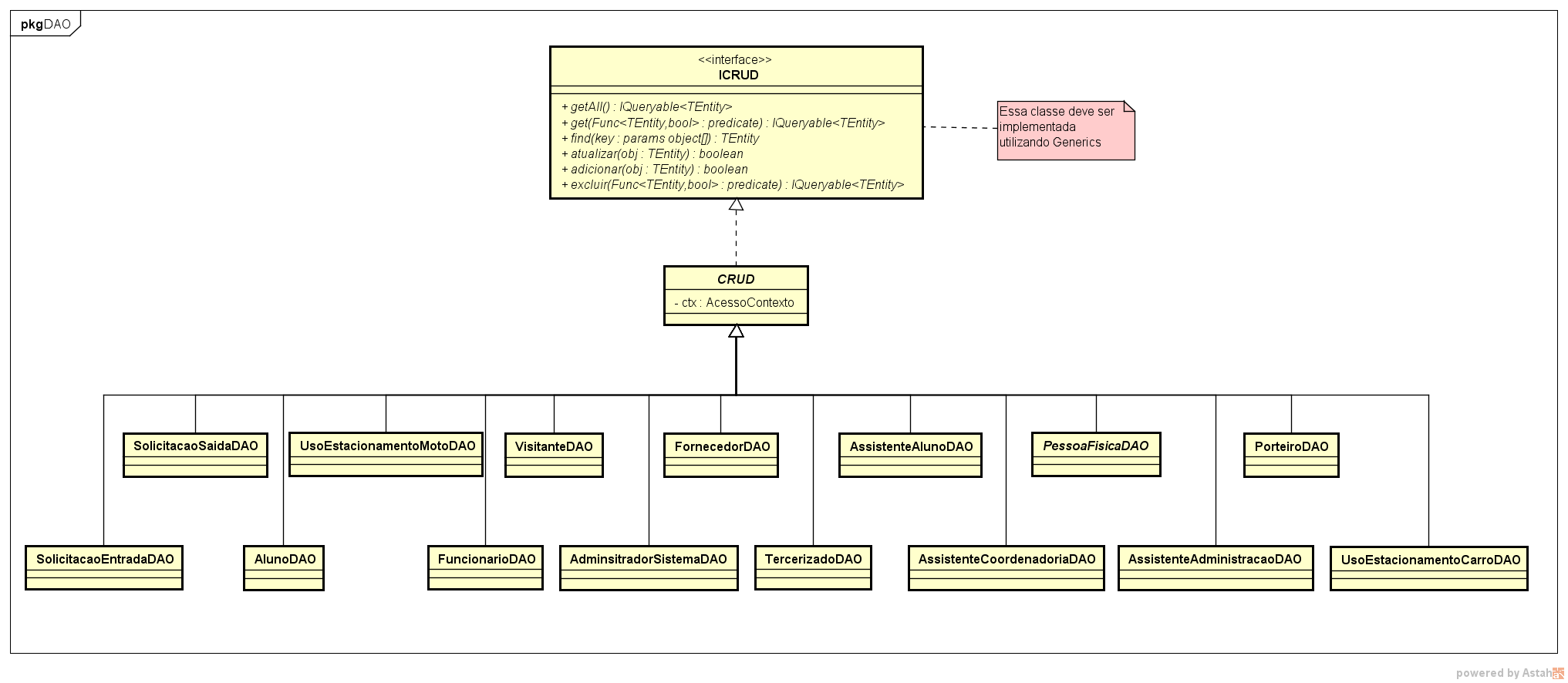


Figura 4 - DC - Representa a Camada DAO do Sistema.

Cada diagrama de classe representa uma camada existente em nosso projeto, com seus respectivos atributos e métodos que se relacionam com outra classe e vice-versa. A figura 2 representa a camada Modelo, ela contempla todos os objetos que serão utilizados pelo sistema, que neste projeto são, aluno, assistente de alunos, porteiro, funcionário, terceirizado, entre outros contidos na mesma. Sçao classes que possuem apenas atributos (nome, RG, idProntuário, tipoVaga entre outros) e seus métodos *Get* e *Set*.

A figura 3 representa a camada DAL (*Data Access Layer* – Camada de Acesso a Dados) que é responsável por interagir com o banco de dados, que neste projeto é onde há contexto entre o banco de dados em que o *Entity Framework* utiliza.

E, a firura 4, representa a camada DAO (*Data Access Object* – Objeto de Acesso a Dados), que é responsável pelos acessos dos objetos a fonte de dados. Nessa camada é onde existe o CRUD (adicionar, pesquisar, alterar e deletar objetos).

## DIAGRAMA DE SEQUÊNCIA

Um diagrama interativo com objetivo de representar a sequência dos processos, mais especificamente de mensagens passadas entre objetos no decorrer do tempo para a realização de uma operação no sistema. Como o projeto tem uma grande quantidade de métodos em classes diferentes, pode ser difícil determinar a sequência global do comportamento. O diagrama de sequência representa essa informação de uma forma simples e lógica. A seguir os diagramas de sequência que este projeto comporta.

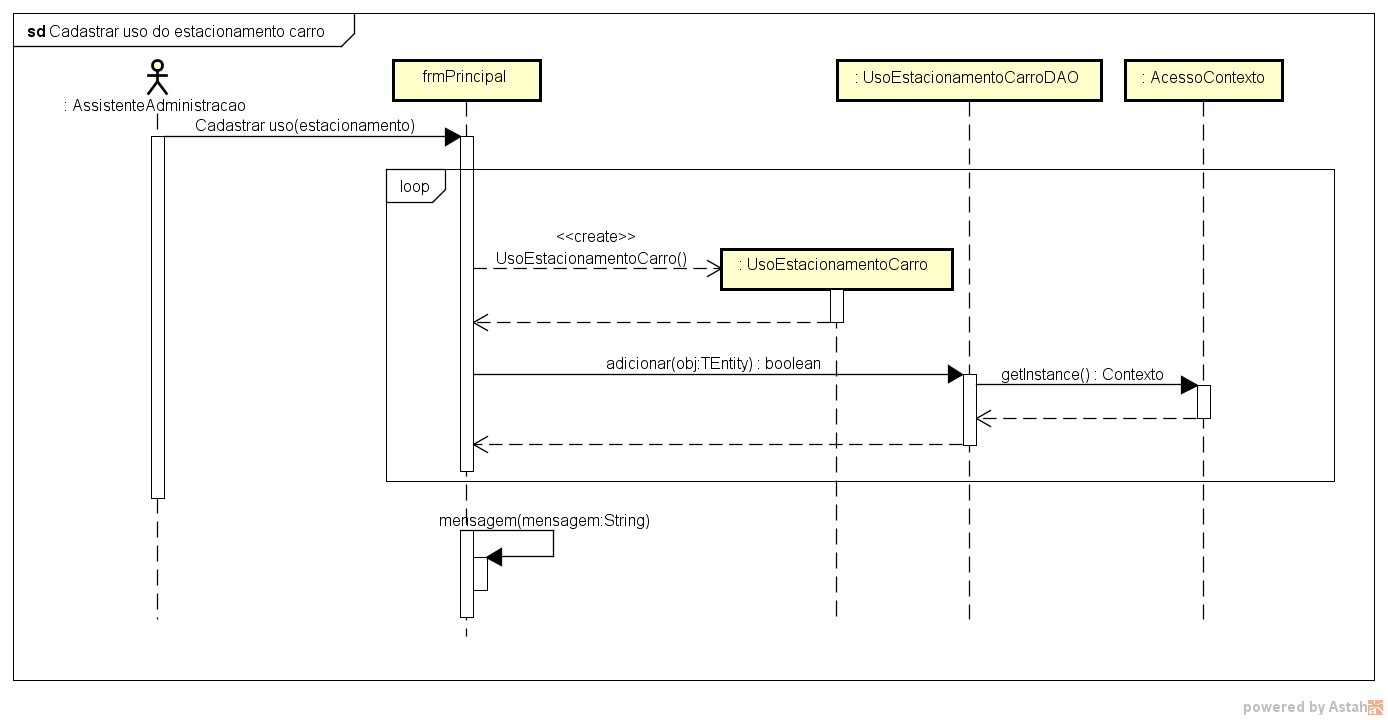


Figura 5 - DS - Cadastrar Uso do Estacionamento para Carro.

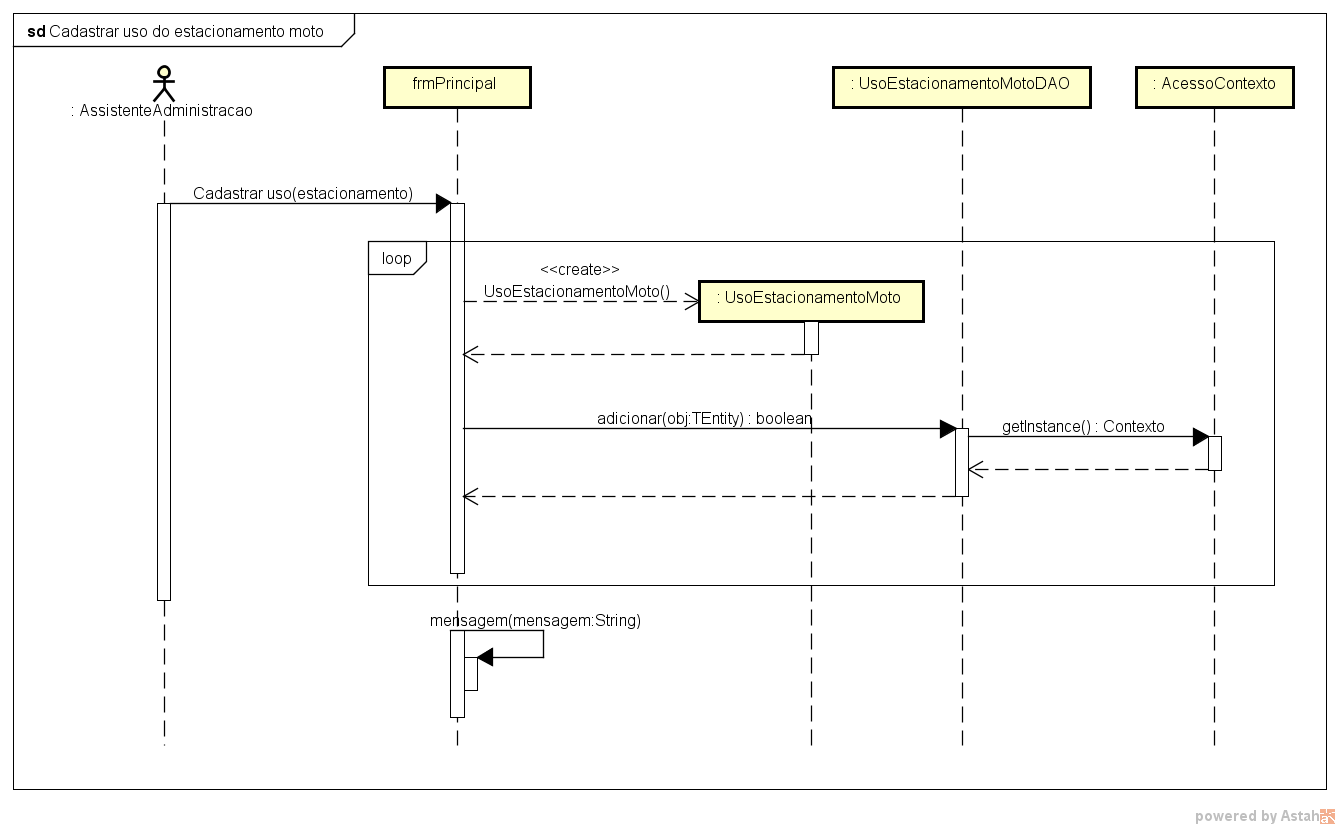


Figura 6 - DS - Cadastrar Uso do Estacionamento para Moto.

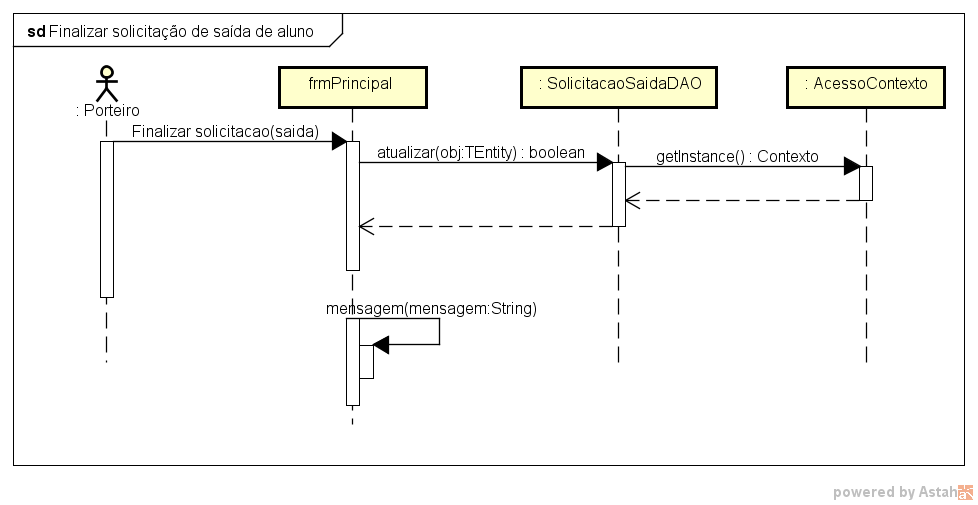


Figura 7 - DS - Finalizar Solicitação de Saída de Aluno.

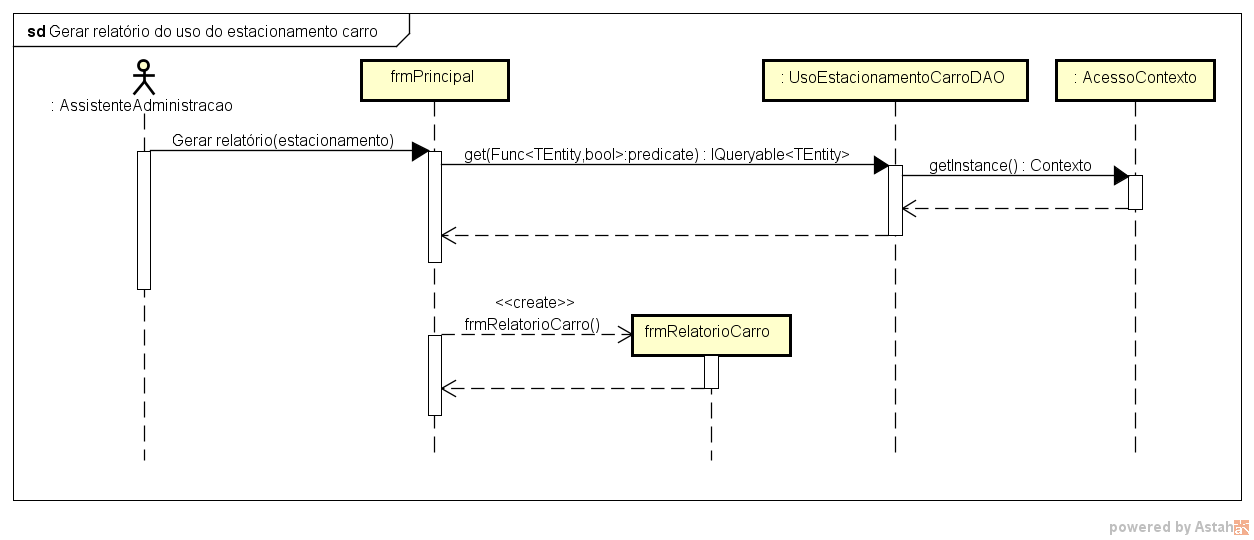


Figura 8 - DS - Gerar Relatorio do Uso do Estacionamento Carro.

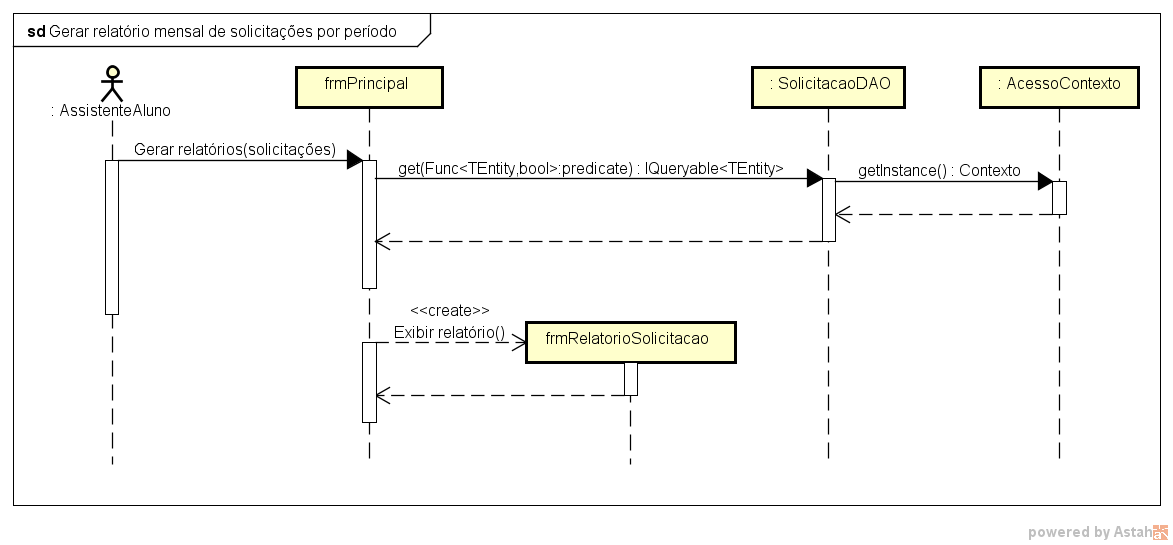


Figura 9 - DS - Gerar Relatório Mensal de Solicitações por Período.

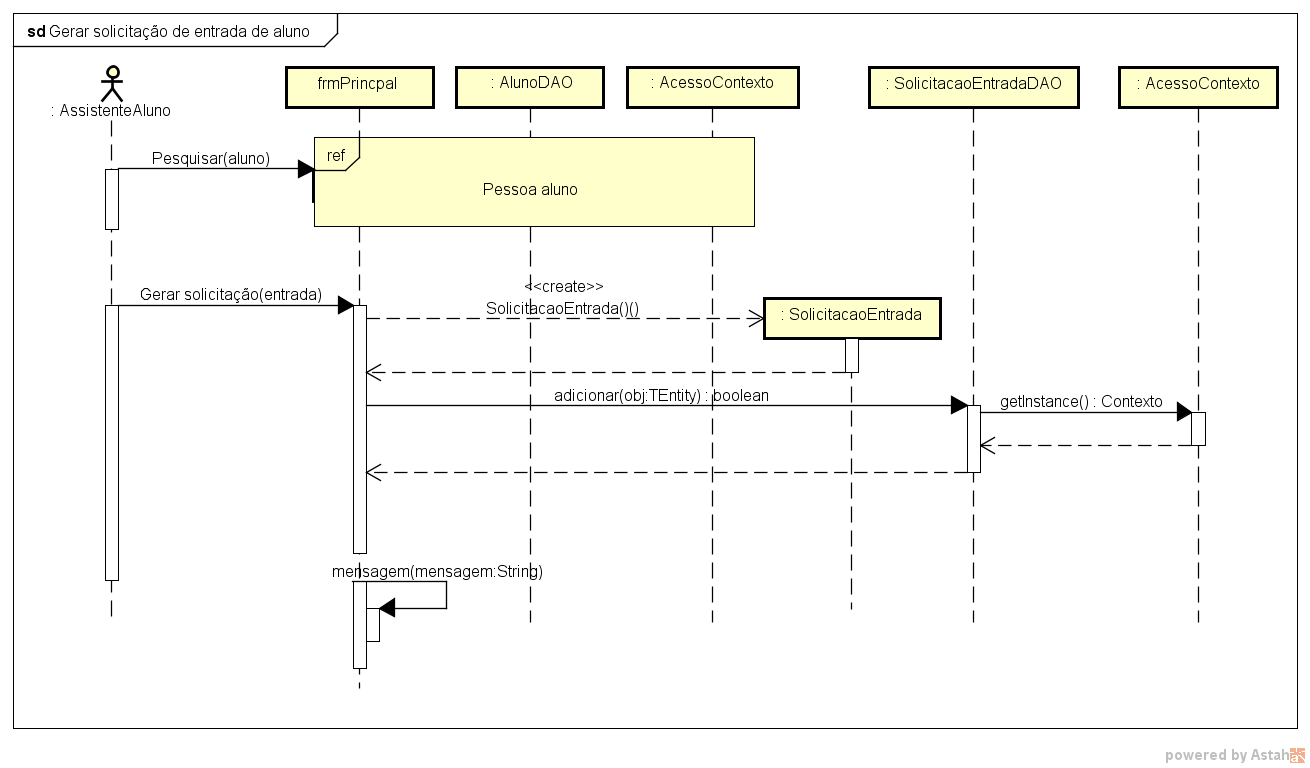


Figura 10 - DS - Gerar Solicitação de Entrada de Aluno.

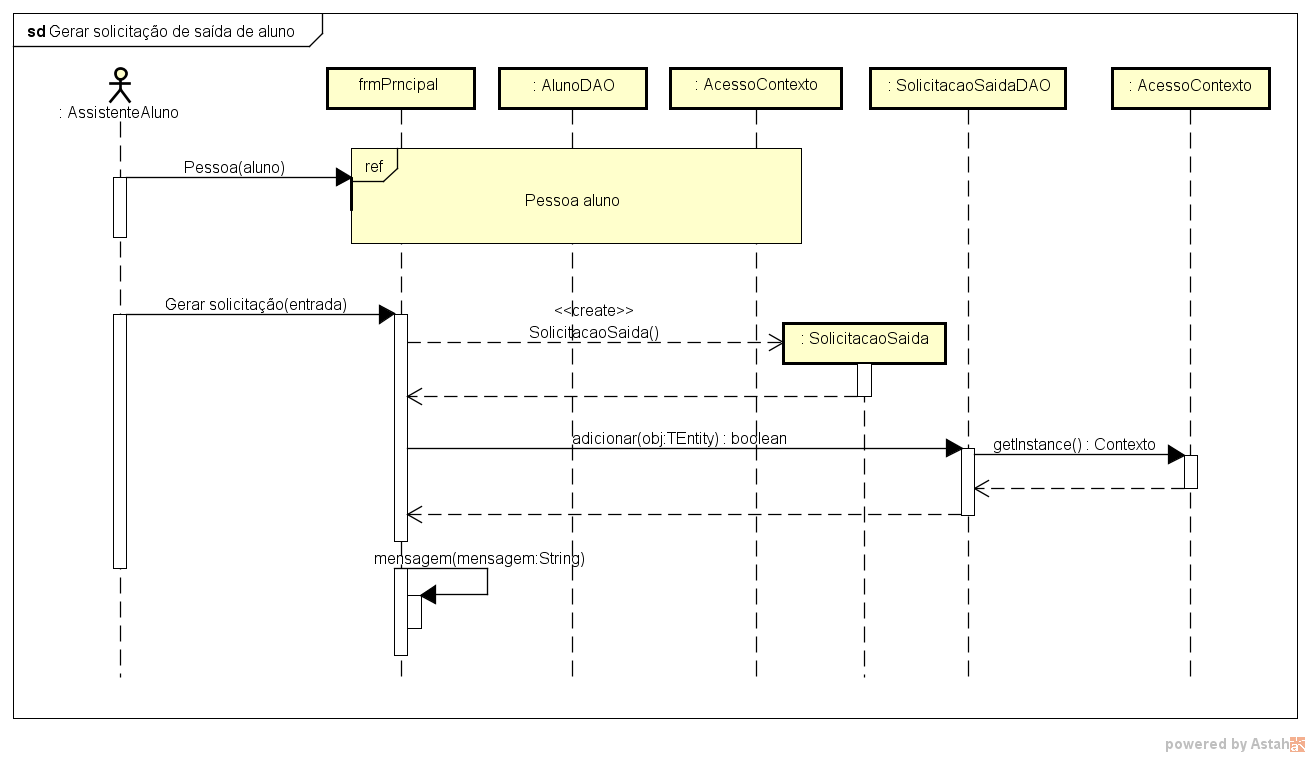


Figura 11 - DS - Gerar Relatorio de Saida de Aluno.

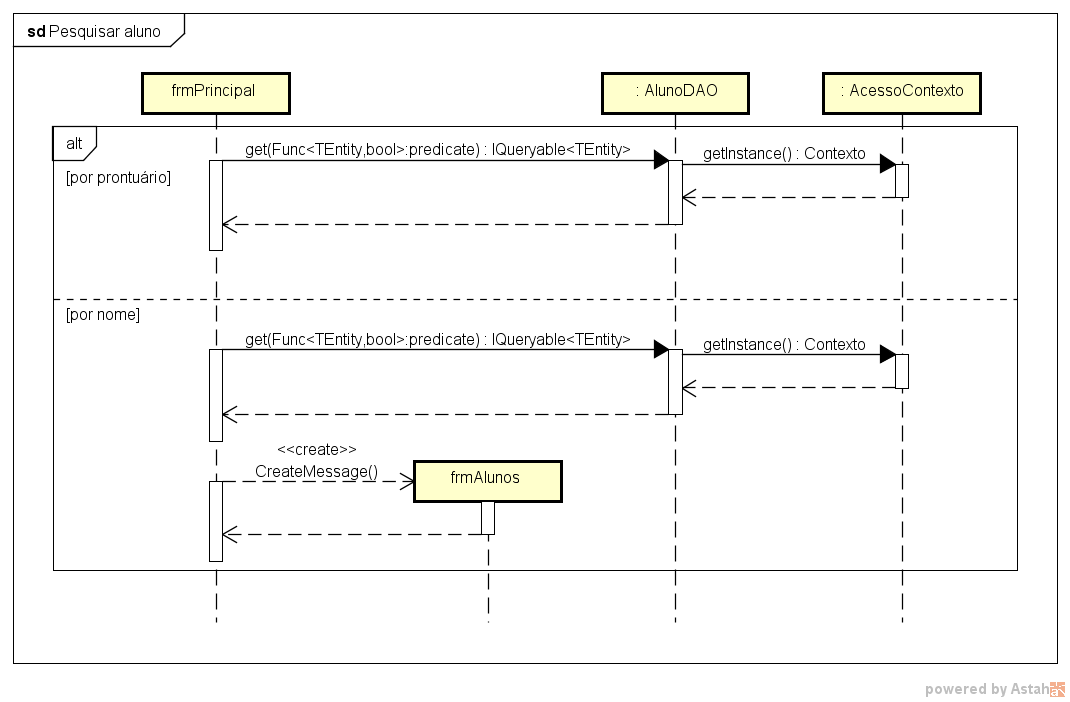


Figura 12 - DS - Pesquisar Aluno.

## DIAGRAMA de atividades

O Diagrama de atividade é um diagrama que representa os fluxos conduzidos por processamentos. É essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra. Comumente isso envolve a modelagem das etapas sequenciais em um processo computacional. A seguir os diagramas de atividades de cada fluxo do sistema.

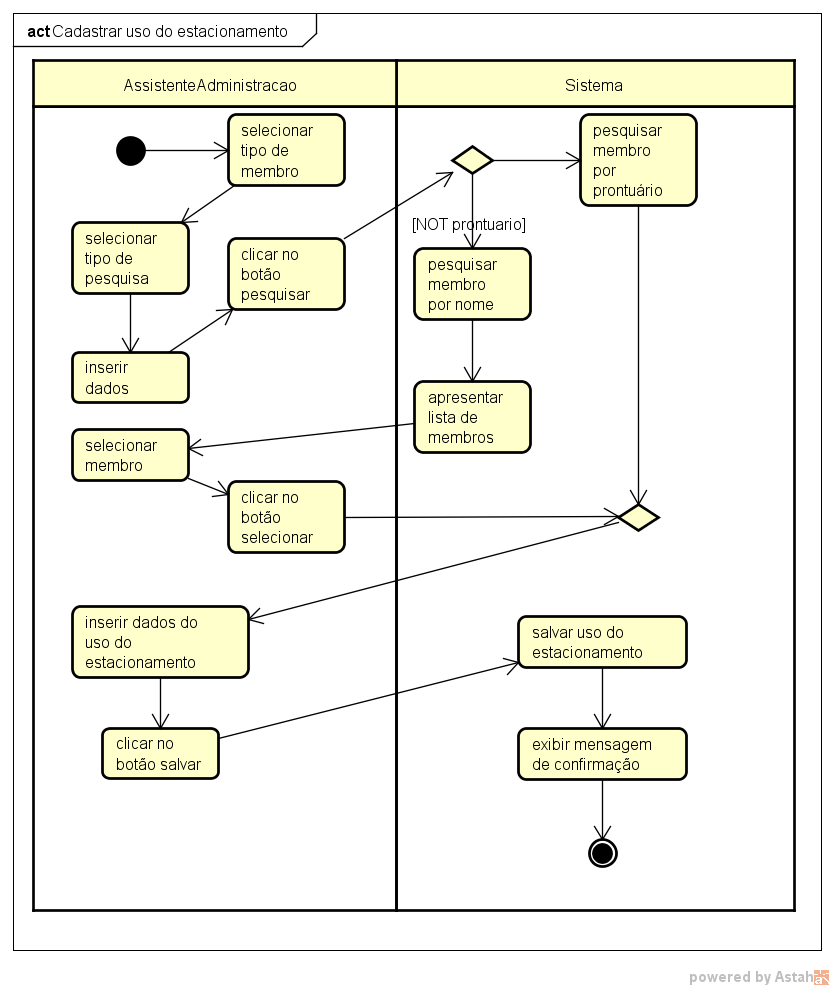


Figura 13 - DA - Cadastrar Uso do Estacionamento.

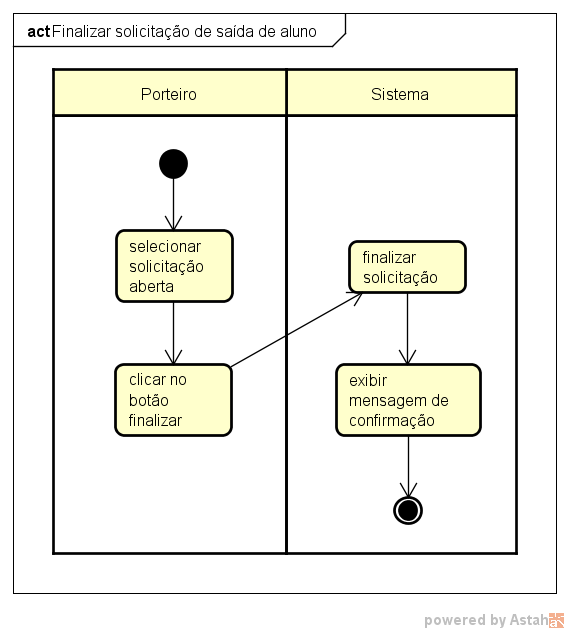


Figura 14 - DA - Finalizar Solicitação de Saída de Aluno.

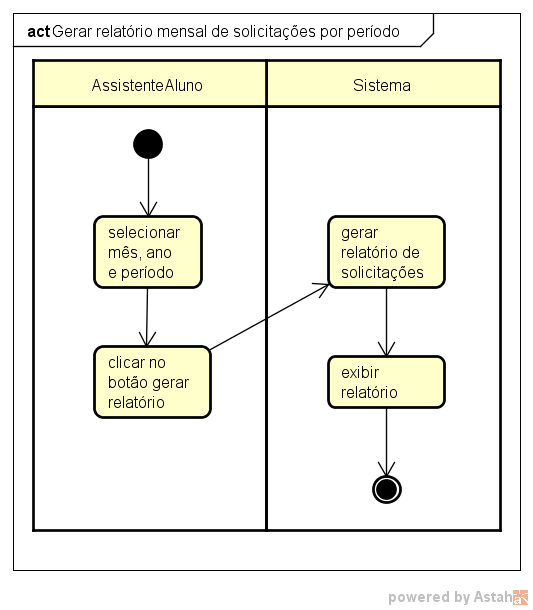


Figura 15 - DA - Gerar Relatório de Solicitações por Período.

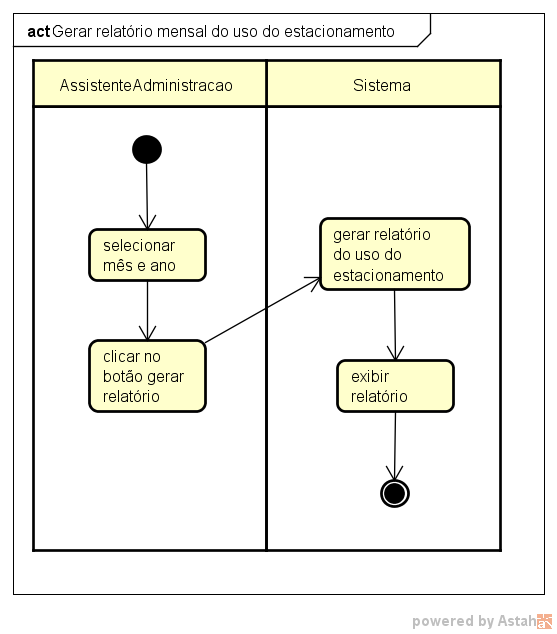


Figura 16 - DA - Gerar Relatório Mensal do Uso do Estacionamento.

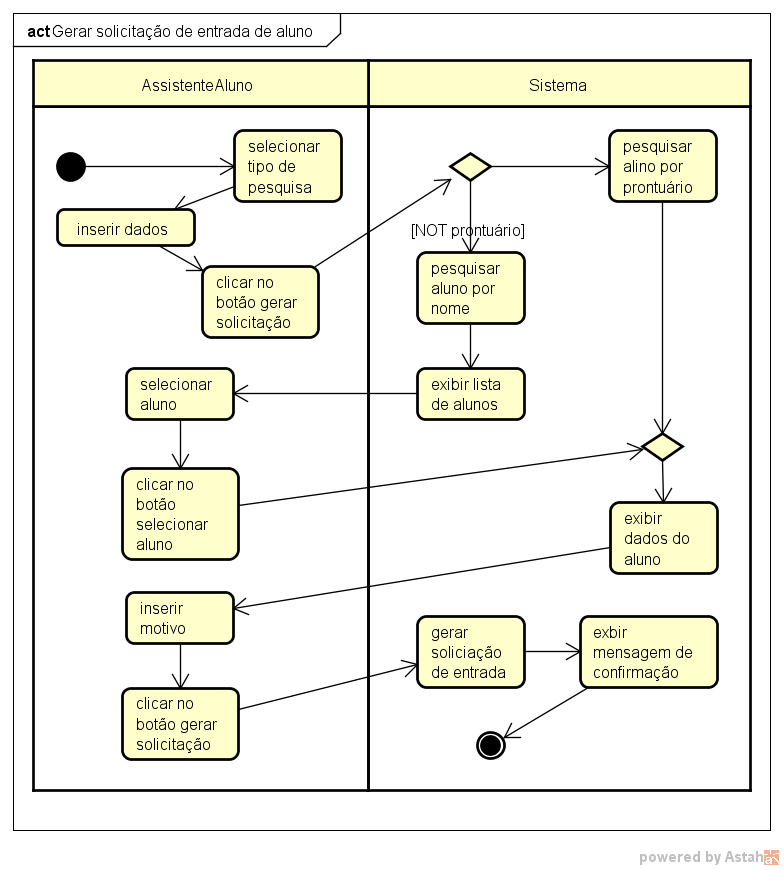


Figura 17 - DA - Gerar Solicitação de Entrada de Aluno.

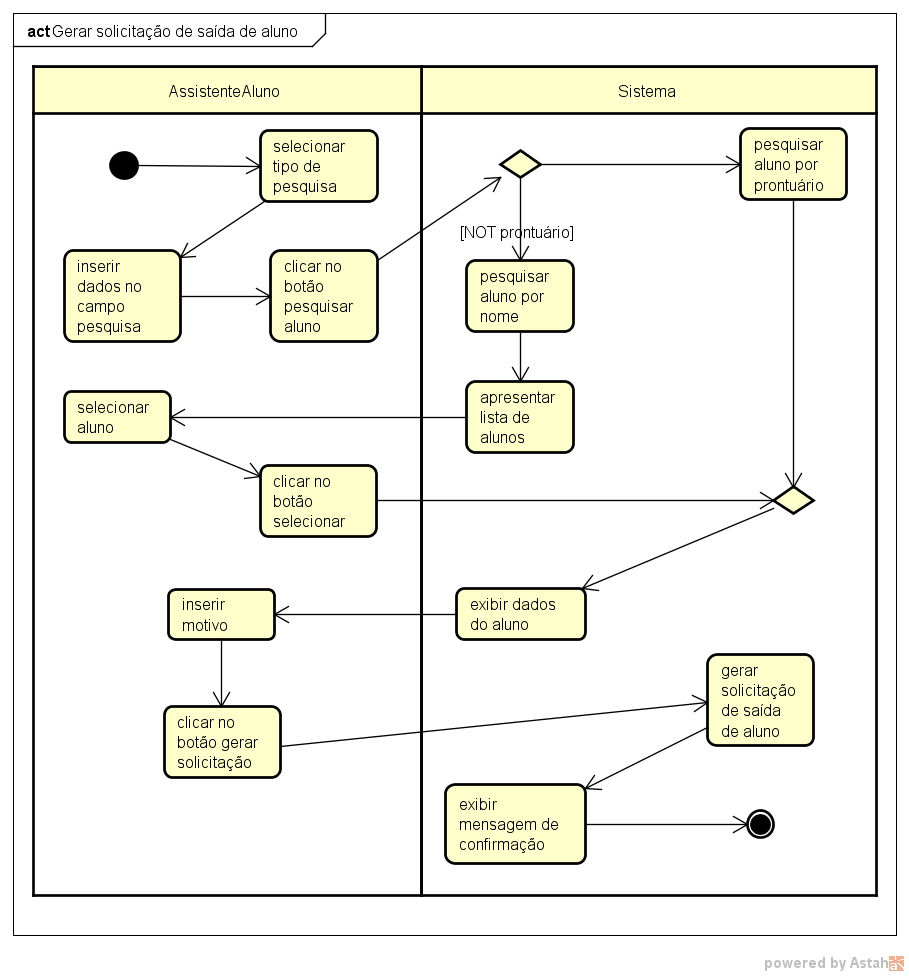


Figura 18 - DA - Gerar Solicitação de Saída de Aluno.

## Diagrama de máquina de estados

Este diagrama acompanha as mudanças sofridas nos estados de uma instância de uma determinada classe, representando a situação em que um objeto se encontra em um determinado momento durante o período que este participa de um processo. A seguir o diagrama de máquina de estados do sistema, representado a solicitação de saída de aluno.

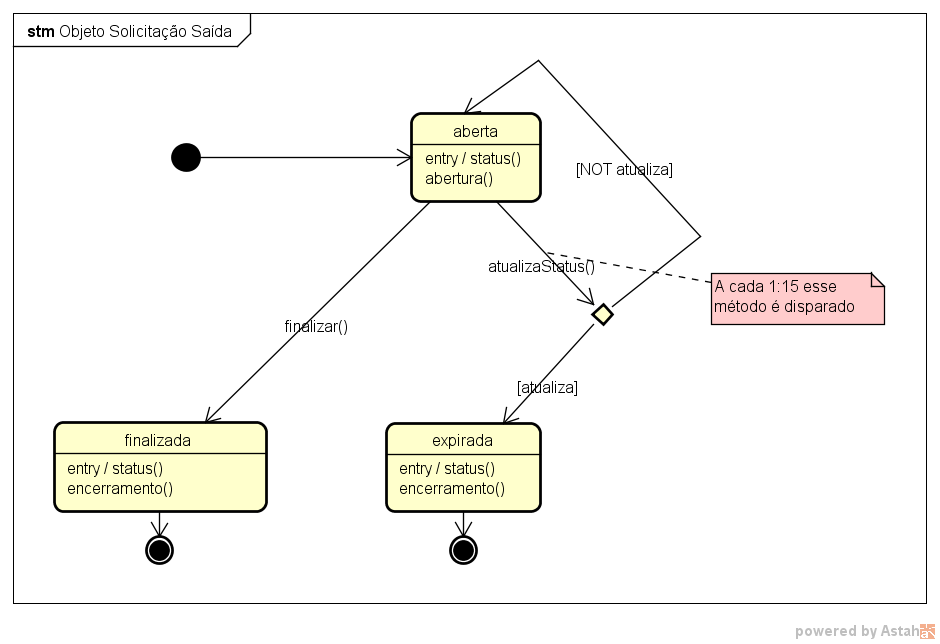


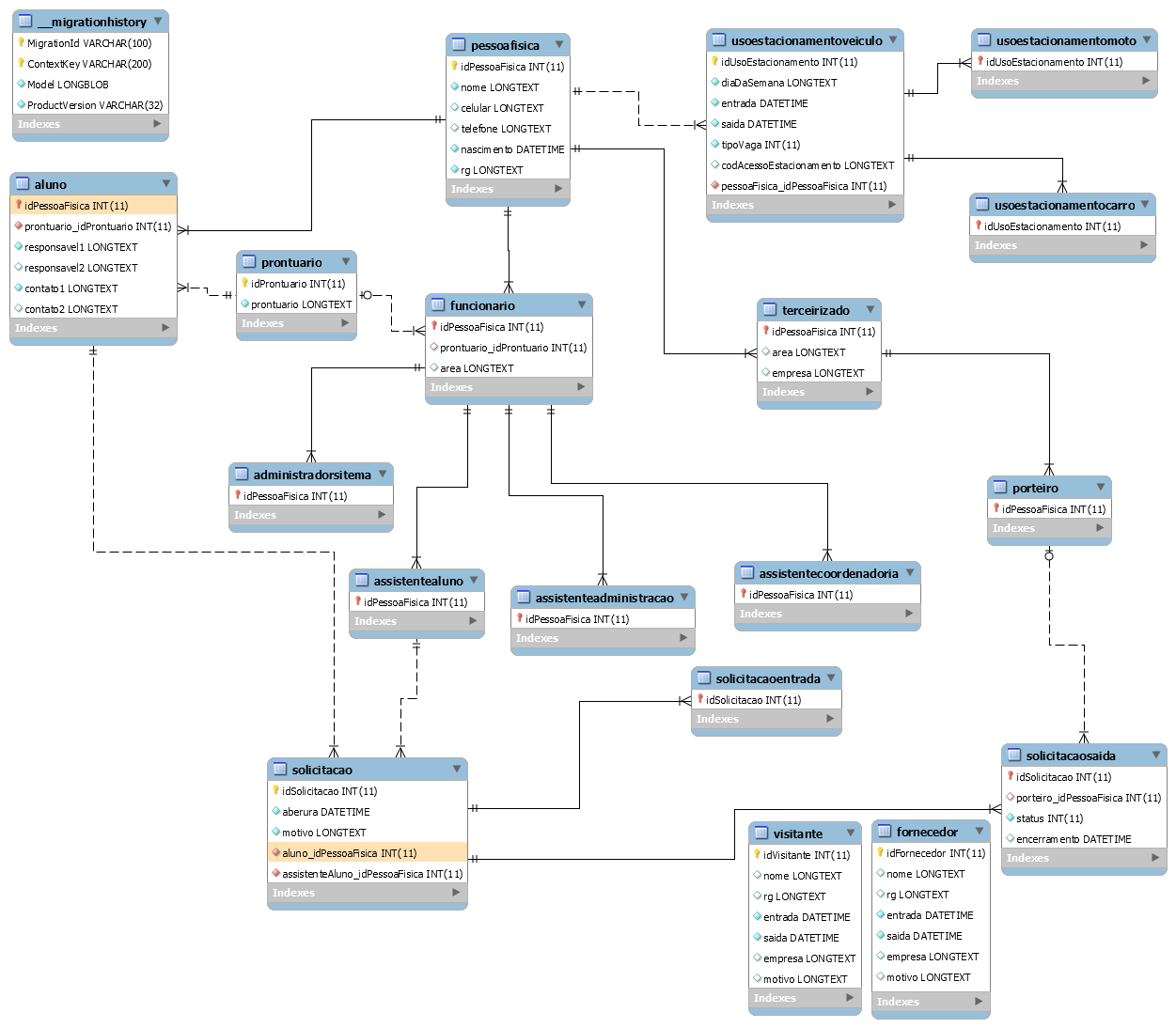
Figura 19 - DME - Objeto Solicitação de Saída.

## MODELO de dados

Este é responsável pelo armazenamento dos dados.

### Modelo de entidade e Relacionamento

A seguir encontra-se uma figura do MER, ou seja, o modelo lógico de relacionamento.



### CRIAÇÃO FÍSICA DA BASE DE DADOS

CREATE DATABASE IF NOT EXISTS `sistemaifsp` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;

USE `sistemaifsp`;

DROP TABLE IF EXISTS `\_\_migrationhistory`;

CREATE TABLE `\_\_migrationhistory` (

`MigrationId` varchar(100) NOT NULL,

`ContextKey` varchar(200) NOT NULL,

`Model` longblob NOT NULL,

`ProductVersion` varchar(32) NOT NULL,

PRIMARY KEY (`MigrationId`,`ContextKey`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `\_\_migrationhistory` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `administradorsitema`;

CREATE TABLE `administradorsitema` (

`idPessoaFisica` int(11) NOT NULL,

PRIMARY KEY (`idPessoaFisica`),

KEY `IX\_idPessoaFisica` (`idPessoaFisica`) USING HASH,

CONSTRAINT `FK\_AdministradorSitema\_Funcionario\_idPessoaFisica` FOREIGN KEY (`idPessoaFisica`) REFERENCES `funcionario` (`idPessoaFisica`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `administradorsitema` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `aluno`;

CREATE TABLE `aluno` (

`idPessoaFisica` int(11) NOT NULL,

`prontuario\_idProntuario` int(11) NOT NULL,

`responsavel1` longtext NOT NULL,

`responsavel2` longtext,

`contato1` longtext NOT NULL,

`contato2` longtext,

PRIMARY KEY (`idPessoaFisica`),

KEY `IX\_idPessoaFisica` (`idPessoaFisica`) USING HASH,

KEY `IX\_prontuario\_idProntuario` (`prontuario\_idProntuario`) USING HASH,

CONSTRAINT `FK\_Aluno\_PessoaFisica\_idPessoaFisica` FOREIGN KEY (`idPessoaFisica`) REFERENCES `pessoafisica` (`idPessoaFisica`),

CONSTRAINT `FK\_Aluno\_Prontuario\_prontuario\_idProntuario` FOREIGN KEY (`prontuario\_idProntuario`) REFERENCES `prontuario` (`idProntuario`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `aluno` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `assistenteadministracao`;

CREATE TABLE `assistenteadministracao` (

`idPessoaFisica` int(11) NOT NULL,

PRIMARY KEY (`idPessoaFisica`),

KEY `IX\_idPessoaFisica` (`idPessoaFisica`) USING HASH,

CONSTRAINT `FK\_AssistenteAdministracao\_Funcionario\_idPessoaFisica` FOREIGN KEY (`idPessoaFisica`) REFERENCES `funcionario` (`idPessoaFisica`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `assistenteadministracao` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `assistentealuno`;

CREATE TABLE `assistentealuno` (

`idPessoaFisica` int(11) NOT NULL,

PRIMARY KEY (`idPessoaFisica`),

KEY `IX\_idPessoaFisica` (`idPessoaFisica`) USING HASH,

CONSTRAINT `FK\_AssistenteAluno\_Funcionario\_idPessoaFisica` FOREIGN KEY (`idPessoaFisica`) REFERENCES `funcionario` (`idPessoaFisica`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `assistentealuno` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `assistentecoordenadoria`;

CREATE TABLE `assistentecoordenadoria` (

`idPessoaFisica` int(11) NOT NULL,

PRIMARY KEY (`idPessoaFisica`),

KEY `IX\_idPessoaFisica` (`idPessoaFisica`) USING HASH,

CONSTRAINT `FK\_AssistenteCoordenadoria\_Funcionario\_idPessoaFisica` FOREIGN KEY (`idPessoaFisica`) REFERENCES `funcionario` (`idPessoaFisica`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `assistentecoordenadoria` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `fornecedor`;

CREATE TABLE `fornecedor` (

`idFornecedor` int(11) NOT NULL AUTO\_INCREMENT,

`nome` longtext,

`rg` longtext,

`entrada` datetime NOT NULL,

`saida` datetime NOT NULL,

`empresa` longtext,

`motivo` longtext,

PRIMARY KEY (`idFornecedor`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `fornecedor` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `funcionario`;

CREATE TABLE `funcionario` (

`idPessoaFisica` int(11) NOT NULL,

`prontuario\_idProntuario` int(11) DEFAULT NULL,

`area` longtext,

PRIMARY KEY (`idPessoaFisica`),

KEY `IX\_idPessoaFisica` (`idPessoaFisica`) USING HASH,

KEY `IX\_prontuario\_idProntuario` (`prontuario\_idProntuario`) USING HASH,

CONSTRAINT `FK\_Funcionario\_PessoaFisica\_idPessoaFisica` FOREIGN KEY (`idPessoaFisica`) REFERENCES `pessoafisica` (`idPessoaFisica`),

CONSTRAINT `FK\_Funcionario\_Prontuario\_prontuario\_idProntuario` FOREIGN KEY (`prontuario\_idProntuario`) REFERENCES `prontuario` (`idProntuario`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `funcionario` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `pessoafisica`;

CREATE TABLE `pessoafisica` (

`idPessoaFisica` int(11) NOT NULL AUTO\_INCREMENT,

`nome` longtext NOT NULL,

`celular` longtext,

`telefone` longtext,

`nascimento` datetime NOT NULL,

`rg` longtext NOT NULL,

PRIMARY KEY (`idPessoaFisica`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `pessoafisica` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `porteiro`;

CREATE TABLE `porteiro` (

`idPessoaFisica` int(11) NOT NULL,

PRIMARY KEY (`idPessoaFisica`),

KEY `IX\_idPessoaFisica` (`idPessoaFisica`) USING HASH,

CONSTRAINT `FK\_Porteiro\_Terceirizado\_idPessoaFisica` FOREIGN KEY (`idPessoaFisica`) REFERENCES `terceirizado` (`idPessoaFisica`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `porteiro` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `prontuario`;

CREATE TABLE `prontuario` (

`idProntuario` int(11) NOT NULL AUTO\_INCREMENT,

`prontuario` longtext NOT NULL,

PRIMARY KEY (`idProntuario`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `prontuario` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `solicitacao`;

CREATE TABLE `solicitacao` (

`idSolicitacao` int(11) NOT NULL AUTO\_INCREMENT,

`aberura` datetime NOT NULL,

`motivo` longtext NOT NULL,

`aluno\_idPessoaFisica` int(11) NOT NULL,

`assistenteAluno\_idPessoaFisica` int(11) NOT NULL,

PRIMARY KEY (`idSolicitacao`),

KEY `IX\_aluno\_idPessoaFisica` (`aluno\_idPessoaFisica`) USING HASH,

KEY `IX\_assistenteAluno\_idPessoaFisica` (`assistenteAluno\_idPessoaFisica`) USING HASH,

CONSTRAINT `FK\_62d7a8c35f8d465397062d549e6b4055` FOREIGN KEY (`assistenteAluno\_idPessoaFisica`) REFERENCES `assistentealuno` (`idPessoaFisica`),

CONSTRAINT `FK\_Solicitacao\_Aluno\_aluno\_idPessoaFisica` FOREIGN KEY (`aluno\_idPessoaFisica`) REFERENCES `aluno` (`idPessoaFisica`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `solicitacao` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `solicitacaoentrada`;

CREATE TABLE `solicitacaoentrada` (

`idSolicitacao` int(11) NOT NULL,

PRIMARY KEY (`idSolicitacao`),

KEY `IX\_idSolicitacao` (`idSolicitacao`) USING HASH,

CONSTRAINT `FK\_SolicitacaoEntrada\_Solicitacao\_idSolicitacao` FOREIGN KEY (`idSolicitacao`) REFERENCES `solicitacao` (`idSolicitacao`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `solicitacaoentrada` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `solicitacaosaida`;

CREATE TABLE `solicitacaosaida` (

`idSolicitacao` int(11) NOT NULL,

`porteiro\_idPessoaFisica` int(11) DEFAULT NULL,

`status` int(11) NOT NULL,

`encerramento` datetime DEFAULT NULL,

PRIMARY KEY (`idSolicitacao`),

KEY `IX\_idSolicitacao` (`idSolicitacao`) USING HASH,

KEY `IX\_porteiro\_idPessoaFisica` (`porteiro\_idPessoaFisica`) USING HASH,

CONSTRAINT `FK\_SolicitacaoSaida\_Porteiro\_porteiro\_idPessoaFisica` FOREIGN KEY (`porteiro\_idPessoaFisica`) REFERENCES `porteiro` (`idPessoaFisica`),

CONSTRAINT `FK\_SolicitacaoSaida\_Solicitacao\_idSolicitacao` FOREIGN KEY (`idSolicitacao`) REFERENCES `solicitacao` (`idSolicitacao`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `solicitacaosaida` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `terceirizado`;

CREATE TABLE `terceirizado` (

`idPessoaFisica` int(11) NOT NULL,

`area` longtext,

`empresa` longtext,

PRIMARY KEY (`idPessoaFisica`),

KEY `IX\_idPessoaFisica` (`idPessoaFisica`) USING HASH,

CONSTRAINT `FK\_Terceirizado\_PessoaFisica\_idPessoaFisica` FOREIGN KEY (`idPessoaFisica`) REFERENCES `pessoafisica` (`idPessoaFisica`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `terceirizado` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `usoestacionamentocarro`;

CREATE TABLE `usoestacionamentocarro` (

`idUsoEstacionamento` int(11) NOT NULL,

PRIMARY KEY (`idUsoEstacionamento`),

KEY `IX\_idUsoEstacionamento` (`idUsoEstacionamento`) USING HASH,

CONSTRAINT `FK\_31d6fea9b07642dda67d960bcd51462a` FOREIGN KEY (`idUsoEstacionamento`) REFERENCES `usoestacionamentoveiculo` (`idUsoEstacionamento`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `usoestacionamentocarro` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `usoestacionamentomoto`;

CREATE TABLE `usoestacionamentomoto` (

`idUsoEstacionamento` int(11) NOT NULL,

PRIMARY KEY (`idUsoEstacionamento`),

KEY `IX\_idUsoEstacionamento` (`idUsoEstacionamento`) USING HASH,

CONSTRAINT `FK\_4d5d7cbb84bd415d86ece839ef82ffbe` FOREIGN KEY (`idUsoEstacionamento`) REFERENCES `usoestacionamentoveiculo` (`idUsoEstacionamento`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `usoestacionamentomoto` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `usoestacionamentoveiculo`;

CREATE TABLE `usoestacionamentoveiculo` (

`idUsoEstacionamento` int(11) NOT NULL AUTO\_INCREMENT,

`diaDaSemana` longtext NOT NULL,

`entrada` datetime NOT NULL,

`saida` datetime NOT NULL,

`tipoVaga` int(11) NOT NULL,

`codAcessoEstacionamento` longtext,

`pessoaFisica\_idPessoaFisica` int(11) NOT NULL,

PRIMARY KEY (`idUsoEstacionamento`),

KEY `IX\_pessoaFisica\_idPessoaFisica` (`pessoaFisica\_idPessoaFisica`) USING HASH,

CONSTRAINT `FK\_774e8d9cb8ff4513948a1fee109e0ab7` FOREIGN KEY (`pessoaFisica\_idPessoaFisica`) REFERENCES `pessoafisica` (`idPessoaFisica`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `usoestacionamentoveiculo` WRITE;

UNLOCK TABLES;

DROP TABLE IF EXISTS `visitante`;

CREATE TABLE `visitante` (

`idVisitante` int(11) NOT NULL AUTO\_INCREMENT,

`nome` longtext,

`rg` longtext,

`entrada` datetime NOT NULL,

`saida` datetime NOT NULL,

`empresa` longtext,

`motivo` longtext,

PRIMARY KEY (`idVisitante`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `visitante` WRITE;

UNLOCK TABLES;

### DICIONÁRIO DE DADOS

A seguir encontra-se a tabela detalhada do modelo de dados lógico do sistema.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | \_\_migrationhistory | | | | |
| **Descrição** | Tabela gerada pelo ententy framework | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| MigrationId | |  | VARCHAR | 100 | PK |
| ContextKey | |  | VARCHAR | 200 | PK |
| Model | |  | LONGBLOB |  | NOT NULL |
| ProductVersion | |  | VARCHAR | 32 | NOT NULL |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | pessoafisica | | | | |
| **Descrição** | Armazenará os dados em comum de uma pessoa física | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idPessoaFisica | | Código da pessoa física. | INTEGER | 11 | PK |
| nome | | Nome da pessoa física. | LONGTEXT |  | NOT NULL |
| celular | | Numero de contado do dispositivo móvel. | LONGTEXT |  | NOT NULL |
| telefone | | Numero de contado do aparelho fixo. | LONGTEXT |  | NOT NULL |
| nascimento | | Data de nascimento de pessoa física. | DATETIME |  | NOT NULL |
| rg | | Número da carteira de identidade. | LONGTEXT |  | NOT NULL |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | aluno | | | | |
| **Descrição** | Armazenará os dados de aluno | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idPessoaFisica | | Código da pessoa física | INTEGER | 11 | PK |
| Prontuário\_idprontuario | | Código do prontuário. | INTEGER | 11 | FK |
| responsavel1 | | Nome de um responsável pelo aluno. | LONGTEXT |  | NOT NULL |
| responsavel2 | | Nome de outro responsável pelo aluno. | LONGTEXT |  | NOT NULL |
| contato1 | | Numero de contato do responsável. | DATETIME |  | NOT NULL |
| contato2 | | Numero de contato do responsável. |  |  |  |
| rg | | Código da carteira de identificação. | LONGTEXT |  | NOT NULL |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | funcionario | | | | |
| **Descrição** | Armazenará os dados do funcionário. | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idPessoaFisica | | Código da pessoa física | INTEGER | 11 | PK |
| prontuario\_prontuario | | Código de prontuário | INTEGER | 11 | FK |
| area | | Setor de atuação | LONGTEXT |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | administradorsistema | | | | |
| **Descrição** | Armazenará os dados do administrador do sistema. | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idPessoaFisica | | Código da pessoa física | INTEGER | 11 | PK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | assistentealuno | | | | |
| **Descrição** | Armazenará os dados da assistente de aluno | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idPessoaFisica | | Código da pessoa física | INTEGER | 11 | PK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | assistenteadministracao | | | | |
| **Descrição** | Armazenará os dados do assistente da administração | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idPessoaFisica | | Código da pessoa física | INTEGER | 11 | PK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | assistentecoordenadoria | | | | |
| **Descrição** | Armazenará os dados do assistente da coordenadoria. | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idPessoaFisica | | Código da pessoa física | INTEGER | 11 | PK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | porteiro | | | | |
| **Descrição** | Armazenará os dados do porteiro. | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idPessoaFisica | | Código da pessoa física | INTEGER | 11 | PK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | prontuario | | | | |
| **Descrição** | Armazenará os prontuários | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idPessoaFisica | | Código de prontuário. | INTEGER | 11 | PK |
| prontuario | | Número do prontuário. | LONGTEXT |  | NOT NULL |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | terceirizado | | | | |
| **Descrição** | Armazenará os dados dos terceirizados | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idPessoaFisica | | Código de prontuário. | INTEGER | 11 | PK |
| area | | Area de atuação | LONGTEXT |  |  |
| empresa | | Local onde é registrado como funcionário. | LONGTEXT |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | | solicitacao | | | |
| **Descrição** | | Armazenará os dados da solicitação | | | |
| **Observações** | |  | | | |
| **Campos** | | | | | |
| **Nome** | **Descrição** | | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idSolicitacao | Código da solicitação. | | INTEGER | 11 | PK |
| abertura | Horário em que começa a valer a solicitação. | | DATETIME |  | NOT NULL |
| motivo | Texto explicando a motivação da saída. | | LONGTEXT |  | NOT NULL |
| aluno\_idPessoaFisica | Código de aluno (pessoa física). | | INTEGER | 11 | FK |
| assistenteAluno\_idPessoaFisica | Código de assistente de aluno (pessoa física). | | INTEGER | 11 | FK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | solicitacaoentrada | | | | |
| **Descrição** | Armazenará as solicitações de entrada. | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idSolicitacao | | Código da solicitação. | INTEGER | 11 | PK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | solicitacaosaida | | | | |
| **Descrição** | Armazenará as solicitações de saída. | | | | |
| **Observações** |  | | | | |
| **Campos** | | | | | |
| **Nome** | | **Descrição** | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idSolicitacao | | Código da solicitação. | INTEGER | 11 | PK |
| Porteiro\_idPessoaFisica | | Código do porteiro (pessoa física). | INTEGER | 11 | FK |
| status | | Conta se aluno já saiu ou não. | INTEGER | 11 | NOT NULL |
| encerramento | | A hora efetiva da saída de aluno | DATETIME |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | | usoestacionamentoveiculo | | | |
| **Descrição** | | Armazenará o uso do estacionamento. | | | |
| **Observações** | |  | | | |
| **Campos** | | | | | |
| **Nome** | **Descrição** | | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idUsoEstacionamento | Código do uso de estacionamento. | | INTEGER | 11 | PK |
| pessoaFisica\_idPessoaFisica | Código da pessoa física. | | INTEGER | 11 | FK |
| diaDaSemana | Os dias que será usado o estacionamento | | LONGTEXT |  | NOT NULL |
| entrada | A hora da entrada do veiculo. | | DATETIME |  | NOT NULL |
| saida | Hora da saída do veiculo. | |  |  | NOT NULL |
| tipoVaga | O tipo da vaga | | INTEGER | 11 | NOT NULL |
| codAcessoEstacionamento | Código que determina o uso do estacionamento. | | longtext |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | | usoestacionamentomoto | | | |
| **Descrição** | | Armazenará o uso do estacionamento. | | | |
| **Observações** | |  | | | |
| **Campos** | | | | | |
| **Nome** | **Descrição** | | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idUsoEstacionamento | Código do uso de estacionamento. | | INTEGER | 11 | PK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | | usoestacionamentocarro | | | |
| **Descrição** | | Armazenará o uso do estacionamento. | | | |
| **Observações** | |  | | | |
| **Campos** | | | | | |
| **Nome** | **Descrição** | | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idUsoEstacionamento | Código do uso de estacionamento. | | INTEGER | 11 | PK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | | visitante | | | |
| **Descrição** | | Armazenará os dados de visitante. | | | |
| **Observações** | |  | | | |
| **Campos** | | | | | |
| **Nome** | **Descrição** | | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idVisitante | Código do visitante. | | INTEGER | 11 | PK |
| nome | Nome do visitante. | | LONGTEXT |  |  |
| rg | Documento de identificação do visitante. | | LONGTEXT |  |  |
| entrada | Hora de entrada do visitante. | | DATETIME |  | NOT NULL |
| saida | Horário da saída do visitante. | | DATETIME |  | NOT NULL |
| empresa | Nome da empresa | | LONGTEXT |  |  |
| motivo | Motivação da visita | | LONGTEXT |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela** | | fornecedor | | | |
| **Descrição** | | Armazenará os dados de fornecedor. | | | |
| **Observações** | |  | | | |
| **Campos** | | | | | |
| **Nome** | **Descrição** | | **Tipo** | **Tamanho** | **Observações (PK / FK)** |
| idFornecedor | Código do fornecedor. | | INTEGER | 11 | PK |
| nome | Nome do fornecedor. | | LONGTEXT |  |  |
| rg | Documento de identificação do fornecedor. | | LONGTEXT |  |  |
| entrada | Hora de entrada do fornecedor. | | DATETIME |  | NOT NULL |
| saida | Horário da saída do fornecedor. | | DATETIME |  | NOT NULL |
| empresa | Nome da empresa | | LONGTEXT |  |  |
| motivo | Motivação da visita | | LONGTEXT |  |  |

### SISTEMAS E COMPONENTES EXTERNOS UTILIZADOS

No momento está sendo analisado o possível uso no NAMBEI, para integrar os alunos matruculados no SCP-IFSP.

# cronograma

A seguir encontra-se as tabelas com o cronograma proposto e o cronograma cumprido, respectivamente.

## Cronograma Proposto

Na tabela a seguir encontra-se marcado com um “X”, os meses planejados para o desenvolvimento das tarefas para a conclusão deste projeto.

1. cronograma do projeto proposto

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Mês 3** | **Mês 4** | **Mês 5** | **Mês 6** | **Mês 7** | **Mês 8** | **Mês 9** | **Mês 10** | **Mês 11** | **Mês 12** |
| **A1** | X |  |  |  |  |  |  |  |  |  |
| **A2** | X | X |  |  |  |  |  |  |  |  |
| **A3** |  | X | X |  |  |  |  |  |  |  |
| **A4** |  |  | X | X | X |  |  |  |  |  |
| **A5** |  |  | X | X | X |  |  |  |  |  |
| **A6** |  |  | X | X | X | X | X |  |  |  |
| **A7** |  |  |  | X | X | X | X |  |  |  |
| **A8** |  |  | X | X |  |  | X |  |  |  |
| **A9** |  |  |  |  |  |  |  | X | X |  |
| **A10** |  |  |  |  | X |  | X |  | X |  |
| **A11** |  |  |  |  |  |  |  |  |  | X |

1. Atividades do cronograma do projeto proposto

|  |  |
| --- | --- |
| **Atividade** | **Tarefa** |
| **A1** | Escolha do tema. Elaboração/Entrega da proposta de TCC. |
| **A2** | Levantamento e Análise de Requisitos. |
| **A3** | Identificação dos softwares utilizados. |
| **A4** | Criação das interfaces com o usuário. |
| **A5** | Estudo aprofundado das linguagens envolvidas e seus respectivos códigos-fontes. |
| **A6** | Início do desenvolvimento do sistema. |
| **A7** | Elaboração do Banco de Dados. |
| **A8** | Elaboração dos diagramas UML. |
| **A9** | Finalização do sistema, testes e correção de erros. |
| **A10** | Revisão do Sistema. |
| **A11** | Entrega do Sistema. |

## Cronograma realizado

Nas tabelas a seguir encontra-se as tarefas realizadas ao decorrer do desenvolvimento do projeto.

1. cronograma do projeto REalizado(Smartsheet)



1. cronograma do projeto Realizado(runrun.it)



# orçamentO

Tendo em vista que o projeto é elaborado de forma acadêmica e institucional, em total benefício de uma Instituição pública, contando com uma elaboração técnica dos alunos envolvidos no projeto, não acarretará custos e/ou ganhos financeiros para o Instituto.

## Softwares necessários

A seguir encontra - se a tabela com os softwares essenciais para a realização do projeto.

1. orçamento do projeto

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Versão | Custo unitário (R$) | Custo total (R$) |
| Astah | Community 6.0 | Gratuíto | Gratuíto |
| Visual Studio | 2015 Enterprise | Licença Estudante | Licença Estudante |
| Microsoft Word | Online | Gratuíto | Gratuíto |
| GitHub | - | Gratuíto | Gratuíto |
| Smartsheet | - | - | Gratuíto por 15 dias |
| Runrun.it | - | Gratuíto | Gratuíto |
| MySql WorkStation | 6.9 | Gratuíto | Gratuíto |
| Entity Framework | 6.0 | Gratuito | Gratuito |

# Plano de TESTE

O SCP-IFSP passará pelos testes unitário e integração esses vão lidar com a qualidade funcional, das bases de dados, interface gráfica e do controle de acesso.

Para a execução dos testes serão utilizadas máquinas, mais idênticas possível, em termos de hardware, àquelas que estão e/ou serão implantadas no instituto, a fim de garantir a previsibilidade de performance e compatibilidade.

## Identificação do Projeto

A tabela a seguir identifica a documentação e disponibilidade para desenvolver o plano de testes:

1. CHECKLIST de documentos para teste

|  |  |  |
| --- | --- | --- |
| **Documento** | **Criado ou Disponível** | **Recebido ou Revisado** |
| Especificação de Requisitos | ◼ Sim 🞏Não | 🞏Sim ◼Não |
| Plano de Projeto | ◼ Sim 🞏Não | 🞏 Sim ◼Não |
| Modelo de Análise | ◼ Sim 🞏Não | ◼ Sim 🞏Não |
| Documento de Arquitetura | ◼ Sim 🞏Não | ◼ Sim 🞏Não |
| Protótipo | ◼ Sim 🞏Não | 🞏 Sim ◼Não |
| Manual do Usuário | 🞏 Sim ◼Não | 🞏 Sim ◼Não |
| Lista de Riscos | 🞏 Sim ◼Não | 🞏 Sim ◼Não |

## Requisitos a Testar

A lista abaixo mostra aqueles itens que foram identificados como alvos de teste, ou seja, representa o que será testado.

### Teste do Banco de Dados

* Verifique que as informações do usuário podem ser cadastradas, consultadas, alteradas e removidas;
* Verifique que as informações úteis obtidas podem ser atualizadas e que as mesmas podem ser apresentadas;
* Verifique que as informações específicas de cada usuário podem ser acompanhadas;
* Verifique que um novo usuário pode ser cadastrado;
* Verifique que os usuários cadastrados possam ser agrupados por categoria;
* Verifique que as informações possam ser visualizadas, cadastradas, removidas e atualizadas pelo administrador do sistema;
* Verifique que as informações úteis cadastradas possam ser consultadas;
* Verifique que o sistema é capaz de buscar e manter atualizadas as informações.

### Teste Funcional

* Verifique que as informações úteis obtidas pelo modulo responsável são automaticamente e periodicamente atualizadas.
* Verifique que qualquer usuário pode acessar sua própria conta através de login e senha.
* Verifique que o relatório da entrada e saída é correto.
* Verifique que as informações podem ser acessadas em qualquer dos módulos disponíveis.

### Teste da Interface do Usuário

* Navegue através de todos os use cases, verificando que cada tela de interface gráfica pode ser rapidamente entendida e facilmente utilizada;
* Verifique que todas as palavras e expressões apresentadas estão em conformidade com as devidas normas sintáticas e gramaticais;
* Verefique que os dados inseridos serão devidamente preenchidos nos campos (campo numérico (de **0** a **9**), campo alfabético (de **Aa** a **Zz**) e campos alfanuméricos).

### Teste de Segurança e de Controle de Acesso

* Verificar que usuários não cadastrados não podem acessar informações restritas dos cadastrados;
* Verificar que além do administrador, ninguém mais pode inserir, atualizar ou remover dados do sistema;
* Verificar que os usuários do sistema podem acessar apenas as funcionalidades e dados associados ao seu próprio tipo de usuário;
* Verificar que a atualização do sistema pode ser feita apenas a partir da rede interna do instituto.

### Teste de Instalação

* Verifique que a instalação do sistema ocorre normalmente em todas as máquinas;
* Verifique que qualquer terminal do sistema do instituto é capaz de rodar o SCP-IFSP normalmente;
* Verifique que o sistema é capaz de obter e atualizar as informações úteis a que se propõe a disponibilizar;
* Verifique que a atualização dos dados no servidor se reflete em todos os terminais do instituto;
* Verifique que o espaço disponível em disco para informações deve ser capaz de armazenar todos os dados/atualizações que forem cadastrados.

## Estratégia de Teste

Essas transações são definidas como funções específicas que um usuário final do sistema é suposto de executar ao usar a aplicação, tais como adicionar ou modificar uma determinada informação.

### Teste de Integridade de Dados e do Banco de Dados

|  |  |
| --- | --- |
| Objetivo do Teste: | Garantir que os métodos e processos de acesso ao banco de dados funcionem de maneira apropriada e sem dados corrompidos. |
| Técnica: | * Invocar cada método e processo de acesso ao banco de dados, alimentando cada um com dados ou requisições de dados válidos e inválidos; * Inspecionar o banco de dados para garantir que os dados foram populados como pretendido, que todos os eventos do banco de dados ocorreram apropriadamente, ou revisar os dados retornados para garantir que os dados corretos foram recuperados corretamente. |
| Critério de Finalização: | Todos os métodos e processos de acesso à base de dados funcionarem como projetados e sem qualquer corrupção de dados. |
| Considerações Especiais: | Processos devem ser invocados manualmente. |

### Teste de Função

|  |  |
| --- | --- |
| Objetivo do Teste: | Garantir a funcionalidade apropriada do alvo do teste, incluindo navegação, entrada de dados, processamento e recuperação. |
| Técnica: | Executar cada caso de uso, fluxo de caso de uso, usando dados válidos e inválidos, para verificar o seguinte:   * Os resultados esperados ocorrem quando dados válidos são usados; * As mensagens de erro ou aviso apropriadas são exibidas quando dados inválidos são usados; * Cada regra de negócio é aplicada apropriadamente. |
| Critério de Finalização: | Todos os testes planejados foram executados.  Todos os defeitos identificados foram tratados. |
| Considerações Especiais: | Nenhuma. |

### Teste da Interface do Usuário

|  |  |
| --- | --- |
| Objetivo do Teste: | Verificar o seguinte:   * A navegação através das funções e os requisitos do negócio apropriadamente, incluindo janela-a-janela, campo-a-campo (tecla tab e movimentos do mouse); * Objetos e características da janela, tais como menus, tamanho, posição, estado e foco conformam-se aos padrões. |
| Técnica: | Criar ou modificar os testes para cada janela para verificar a navegação e os estados de objeto apropriados para cada janela e objetos da aplicação. |
| Critério de Finalização: | É verificado que cada janela permanece consistente com a versão de comparação ou dentro de padrões aceitáveis. |
| Considerações Especiais: | Nenhuma. |

### Teste de Segurança e Controle de Acesso

|  |  |
| --- | --- |
| Objetivo do Teste: | Segurança do Nível de Aplicação: verifique que um ator pode acessar apenas aquelas funções ou dados para os quais o seu tipo de usuário tem permissão. |
| Técnica: | * Segurança do Nível de Aplicação: identifique e liste cada tipo de usuário e as funções ou dados para os quais cada tipo tem permissão; * Crie testes para cada tipo de usuário e verifique cada permissão criando transações específicos para cada tipo de usuário; * Modifique o tipo de usuário e repita os testes para os mesmos usuários. Em cada caso, verifique que funções ou dados adicionais estão corretamente disponíveis ou negados. |
| Critério de Finalização: | Para cada tipo de ator existente, as funções ou dados apropriados estão disponíveis, e todas as transações funcionam como esperado e rodam nos Testes de Função anteriores. |
| Considerações Especiais: | Nenhuma. |

### Teste de Instalação

|  |  |
| --- | --- |
| Objetivo do Teste: | Verifique que os alvos de teste instalam apropriadamente em cada configuração de hardware necessária sobre a seguinte condição:   * Uma nova instalação, em uma nova máquina, que nunca fora anteriormente instalada com o SCP-IFSP. |
| Técnica: | * Começar ou executar a instalação. * Usando um subgrupo predeterminado de scripts de teste de funções, rode as transações. |
| Critério de Finalização: | As transações do SCP-IFSP executam de forma bem-sucedida, sem falha. |
| Considerações Especiais: | Quais transações do SCP-IFSP devem ser selecionadas para abranger um teste de confiança de que a aplicação SCP-IFSP foi instalada de forma bem-sucedida e que nenhum componente importante de software está faltando? |

# iMPLANTAÇÃO

# CONCLUSÃO

# GLOSSÁRIO

ReferÊncias

GUEDES, Gilleanes T. A. *UML 2 Uma abordagem prática*. 2ª Edição. São Paulo: Novatec Editora Ltda, 2011.

**MANUAL DO USUÁRIO**