# 4. RESULTADOS E ANÁLISE DE MODELOS DE DADOS

No capítulo, será apresentado a análise e os modelos de dados do protótipo ‘CHECK-IN TOUR’.

# 4.1 DESCRIÇÕES DO SISTEMA

Visando ajudar a melhorar o modo de trabalho dos pontos turísticos o protótipo ‘CHECK-IN TOUR’ vem com intuito de:

A recepcionista fará o cadastro de todos os visitantes com os seguintes dados: nome, identidade, tipo de visitante, além disso, na hora do cadastro registra automaticamente a hora de entrada.

Poderá gerar relatórios específicos, podendo ser com informações diárias, mensais ou anuais.

# 4.1.1 Diagrama de caso de uso do sistema CHECK-IN TOUR

**Figura 1 – Diagrama de Caso de Uso.**

Fonte: Autoria Própria, 2022.

# 4.1.2 Documentação do caso de uso efetuar login

|  |  |
| --- | --- |
| Casos de uso: | UC01: Efetuar Login. |
| Objetivo: | Acessar o Sistema. |
| Atores: | Recepcionista e Administrador. |
| Prioridade: | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 🞎 | Essencial | 🞎 | Importante | 🞎 | Desejável | |
| Pré-condições: | 1. Login tem que estar criado. |
| Fluxo Principal: | 1. A recepcionista preenche os campos de login e password com seus dados. 2. A recepcionista clica no botão de Login. 3. O sistema executa o Fluxo Alternativo FA01 (Validar Login). 4. Esse caso de uso se encerra. |
| Fluxo Alternativo 01 (Validar Login): | 1. O sistema CHECK-IN TOUR executa o Fluxo de Exceção 01. 2. O sistema CHECK-IN TOUR valida os campos login e password na tabela usuarios no banco de dados checkintour. 3. Se os dados preenchidos não estiverem corretos com a base de dados, o sistema mostrara uma mensagem “Login ou senha inválidos!”. 4. Se os dados preenchidos estiverem corretos com a base de dados, o sistema passa para a tela de Dashboard do sistema CHECK-IN TOUR. 5. Esse caso de uso se encerra. |
| Fluxo de Exceção 01: | 1. Caso os campos estejam em branco o sistema apresenta a mensagem que é obrigatório os campos estarem preenchido. 2. Esse caso de uso se encerra. |
| Regras de negócio: | 1. Só será possível acessar o sistema quem tiver Login. |

# 4.1.3 Documentação do caso de uso cadastrar novo visitante

|  |  |
| --- | --- |
| Casos de uso: | UC02: Cadastrar Novo Visitante. |
| Objetivo: | Registrar um novo visitante. |
| Atores: | Recepcionista e Administrador. |
| Prioridade: | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 🞎 | Essencial | 🞎 | Importante | 🞎 | Desejável | |
| Pré-condições: | 1. Recepcionista tem que estar logada no sistema CHECK-IN TOUR. |
| Fluxo Principal: | 1. A recepcionista irá preencher os campos de nome, rg, escolher o tipo de visitante. 2. A recepcionista clica no botão Salvar. 3. O sistema CHECK-IN TOUR executa o Fluxo de Exceção 01. 4. O sistema CHECK-IN TOUR valida as informações e salva na tabela visitantes no banco de dados checkintour. 5. Na página apresenta a mensagem: Visitante cadastrado com sucesso!. 6. Esse caso de uso se encerra. |
| Fluxo Alternativo: | 1. Não Aplicável. |
| Fluxo de Exceção 01: | 1. Caso os campos estejam em branco o sistema apresenta a mensagem que é obrigatório os campos estarem preenchido. 2. Esse caso de uso se encerra. |
| Regras de negócio: | 1. Todo visitante deverá ser cadastrado. |

# Documentação do caso de uso pesquisar visitante

|  |  |
| --- | --- |
| Casos de uso: | UC03: Pesquisar Visitante. |
| Objetivo: | Pesquisar visitantes cadastrados. |
| Atores: | Recepcionista e Administrador. |
| Prioridade: | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 🞎 | Essencial | 🞎 | Importante | 🞎 | Desejável | |
| Pré-condições: | 1. Recepcionista tem que estar logada no sistema CHECK-IN TOUR. |
| Fluxo Principal: | 1. A recepcionista clicar na aba visitantes. 2. A recepcionista poderá preencher o Nome do visitante e em seguida clicará em buscar. 3. O sistema CHECK-IN TOUR valida as informações e faz a busca na tabela visitantes no banco de dados checkintour. 4. O sistema CHECK-IN TOUR apresenta na tela os dados do visitante. 5. Esse caso de uso se encerra. |
| Fluxo Alternativo: | 1. Não Aplicável. |
| Regras de negócio: | 1. Visitante deverá estar cadastrado. |

# Documentação do caso de alterar visitante

|  |  |
| --- | --- |
| Casos de uso: | UC04: Alterar Visitante. |
| Objetivo: | Alterar visitantes já cadastrados. |
| Atores: | Recepcionista e Administrador. |
| Prioridade: | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 🞎 | Essencial | 🞎 | Importante | 🞎 | Desejável | |
| Pré-condições: | 1. Recepcionista tem que estar logada no sistema CHECK-IN TOUR. |
| Fluxo Principal: | 1. A recepcionista clicará na aba visitantes. 2. A recepcionista clicará no botão Editar de algum visitante cadastrado. 3. O sistema exibirá os dados do visitante. 4. A recepcionista poderá alterar seus dados e em seguida clicará no botão Salvar. 5. O sistema CHECK-IN TOUR valida as informações e faz a alteração na tabela visitantes no banco de dados checkintour. 6. O sistema CHECK-IN TOUR apresenta na tela os dados do visitante já com os seus dados alterados. 7. Esse caso de uso se encerra. |
| Fluxo Alternativo: | 1. Não Aplicável. |
| Regras de negócio: | 1. Visitante deverá estar cadastrado. |

# Documentação do caso de excluir visitante

|  |  |
| --- | --- |
| Casos de uso: | UC05: Excluir Visitante. |
| Objetivo: | Excluir visitantes já cadastrados. |
| Atores: | Recepcionista e Administrador. |
| Prioridade: | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 🞎 | Essencial | 🞎 | Importante | 🞎 | Desejável | |
| Pré-condições: | 1. Recepcionista tem que estar logada no sistema CHECK-IN TOUR. |
| Fluxo Principal: | 1. A recepcionista clicará na aba visitantes. 2. A recepcionista clicará no botão Excluir de algum visitante cadastrado. 3. O sistema exibirá uma tela com a seguinte frase: “Deseja realmente excluir esse visitante?”. 4. O recepcionista confirmará sua exclusão no botão Excluir. 5. O sistema CHECK-IN TOUR valida as informações e faz a alteração visitante\_status para o valor 2 na tabela visitantes no banco de dados checkintour. 6. O sistema CHECK-IN TOUR apresenta na tela a lista atualizada dos visitantes ativos. 7. Esse caso de uso se encerra. |
| Fluxo Alternativo: | 1. Não Aplicável. |
| Regras de negócio: | 1. Visitante deverá estar cadastrado. |

# Documentação do caso de uso gerar relatório geral de visitante

|  |  |
| --- | --- |
| Casos de uso: | UC06: Gerar Relatório Geral de Visitante. |
| Objetivo: | Gerar relatório geral de visitante. |
| Atores: | Recepcionista e Administrador. |
| Prioridade: | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 🞎 | Essencial | 🞎 | Importante | 🞎 | Desejável | |
| Pré-condições: | 1. Recepcionista tem que estar logada no sistema CHECK-IN TOUR. |
| Fluxo Principal: | 1. A recepcionista clicar na aba de relatório. 2. A recepcionista clica na opção gerar relatório geral de visitante. 3. A recepcionista clicará em Buscar. 4. O sistema CHECK-IN TOUR valida as informações e faz a busca na tabela log\_visitas no banco de dados checkintour. 5. O sistema CHECK-IN TOUR apresenta na tela o relatório. 6. Esse caso de uso se encerra. |
| Fluxo Alternativo: | 1. Não Aplicável. |
| Regras de negócio: | 1. Pode ser gerado vários relatórios. |

# Documentação do caso de uso gerar relatório geral de visitante por data

|  |  |
| --- | --- |
| Casos de uso: | UC07: Gerar Relatório Geral de Visitante por Data. |
| Objetivo: | Gerar relatório geral de visitante. |
| Atores: | Recepcionista e Administrador. |
| Prioridade: | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 🞎 | Essencial | 🞎 | Importante | 🞎 | Desejável | |
| Pré-condições: | 1. Recepcionista tem que estar logada no sistema CHECK-IN TOUR. |
| Fluxo Principal: | 1. A recepcionista clicar na aba de relatório. 2. A recepcionista clica na opção gerar relatório geral de visitante. 3. A recepcionista preencherá os campos de data início e data fim e em seguida clicará em Buscar. 4. O sistema CHECK-IN TOUR valida as informações e faz a busca na tabela log\_visitas no banco de dados checkintour. 5. O sistema CHECK-IN TOUR apresenta na tela o relatório. 6. Esse caso de uso se encerra. |
| Fluxo Alternativo: | 1. Não Aplicável. |
| Regras de negócio: | 1. Pode ser gerado vários relatórios. |

# Documentação do caso de uso cadastrar novo usuário

|  |  |
| --- | --- |
| Casos de uso: | UC08: Cadastrar Novo Usuário. |
| Objetivo: | Cadastrar um novo usuario. |
| Atores: | Administrador. |
| Prioridade: | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 🞎 | Essencial | 🞎 | Importante | 🞎 | Desejável | |
| Pré-condições: | 1. O administrador tem que estar logada no sistema CHECK-IN TOUR. |
| Fluxo Principal: | 1. O administrador clicará na aba de usuário. 2. O administrador clicará no botão Novo usuário 3. O sistema exibirá um tela onde mostrará os campos de Nome, Login, Senha e tipo de usuario. 4. O administrador Preencherá e pedirá que o usuario digite sua senha de acesso ao sistema. 5. O sistema CHECK-IN TOUR valida as informações e faz a inserção na tabela usuarios no banco de dados checkintour. 6. O sistema CHECK-IN TOUR apresenta na tela os usuários já cadastrados . 7. Esse caso de uso se encerra. |
| Fluxo Alternativo: | 1. Não Aplicável. |
| Regras de negócio: | 1. Administrador preenchera todas as informações do novo usuário. |

# Documentação do caso de uso alterar usuário

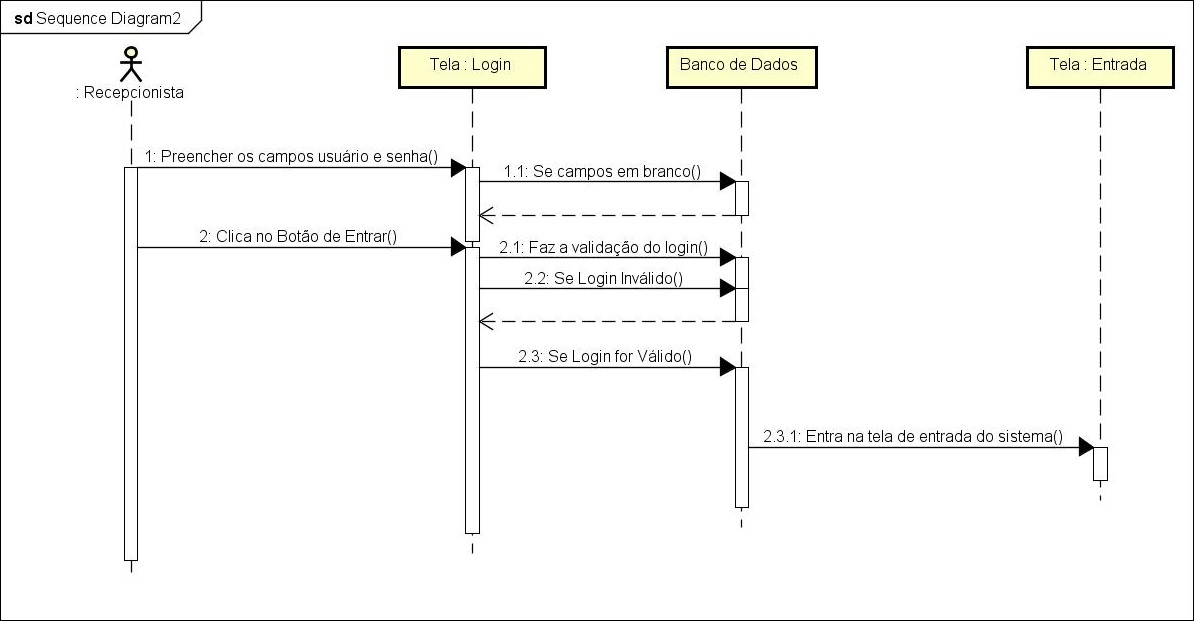
|  |  |
| --- | --- |
| Casos de uso: | UC09: Alterar Usuário. |
| Objetivo: | Alterar usuários já cadastrados. |
| Atores: | Administrador. |
| Prioridade: | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 🞎 | Essencial | 🞎 | Importante | 🞎 | Desejável | |
| Pré-condições: | 1. O administrador tem que estar logado no sistema CHECK-IN TOUR. |
| Fluxo Principal: | 1. O administrador clicará na aba usuário. 2. O administrador clicará no botão Editar de algum usuário cadastrado. 3. O sistema exibirá os dados do usuário. 4. O administrador poderá alterar seus dados e em seguida clicará no botão Salvar. 5. O sistema CHECK-IN TOUR valida as informações e faz a alteração na tabela usuários no banco de dados checkintour. 6. O sistema CHECK-IN TOUR apresenta na tela os dados do usuário já com os seus dados alterados. 7. Esse caso de uso se encerra. |
| Fluxo Alternativo: | 1. Não Aplicável. |
| Regras de negócio: | 1. Visitante deverá estar cadastrado. |

# Documentação do caso de uso alterar usuário

|  |  |
| --- | --- |
| Casos de uso: | UC10: Excluir Usuário. |
| Objetivo: | Excluir usuários já cadastrados. |
| Atores: | Administrador. |
| Prioridade: | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 🞎 | Essencial | 🞎 | Importante | 🞎 | Desejável | |
| Pré-condições: | 1. O Administrador tem que estar logado no sistema CHECK-IN TOUR. |
| Fluxo Principal: | 1. O administrador clicará na aba usuários. 2. O administrador clicará no botão Excluir de algum usuário cadastrado. 3. O sistema exibirá uma tela com a seguinte frase: “Deseja realmente excluir esse usuário?”. 4. O administrador confirmará sua exclusão no botão Excluir. 5. O sistema CHECK-IN TOUR valida as informações e faz a alteração usuario\_status para o valor 2 na tabela usuarios no banco de dados checkintour. 6. O sistema CHECK-IN TOUR apresenta na tela a lista atualizada dos usuários ativos. 7. Esse caso de uso se encerra. |
| Fluxo Alternativo: | 1. Não Aplicável. |
| Regras de negócio: | 1. Usuário deverá estar cadastrado. |

# Diagrama de sequência efetuar login

**Figura 2 – Diagrama de sequência efetuar login.**



Fonte: Autoria Própria, 2022.

# Diagrama de sequência cadastrar novo visitante

# Figura 3 – Diagrama de sequência cadastrar novo visitante.

Fonte: Autoria Própria, 2022.

Fonte: Autoria Própria, 2022.

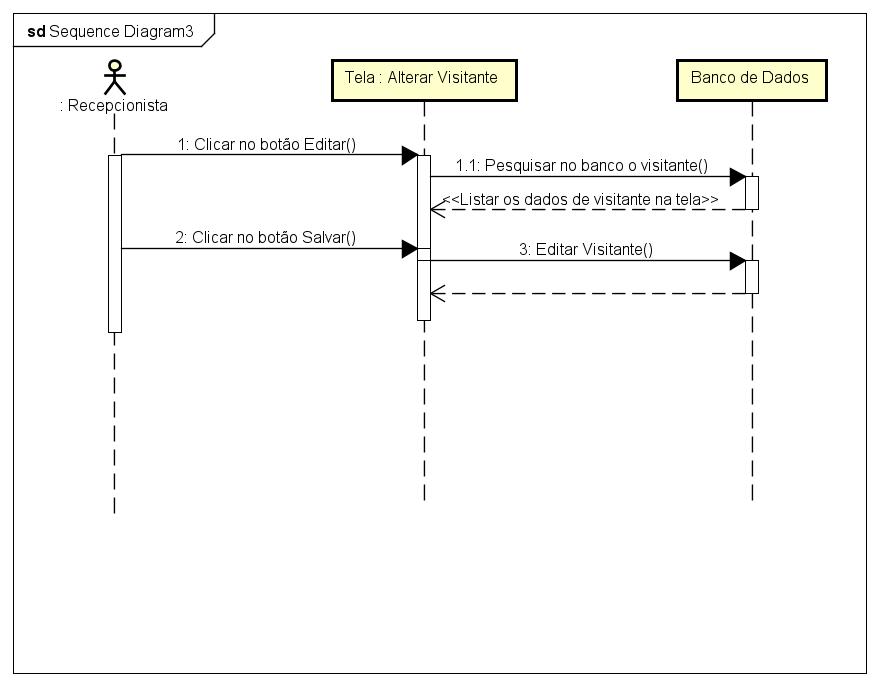
# Diagrama de sequência excluir visitante

# Figura 4 – Diagrama de sequência excluir visitante.

Fonte: Autoria Própria, 2022.

# Diagrama de sequência alterar visitante

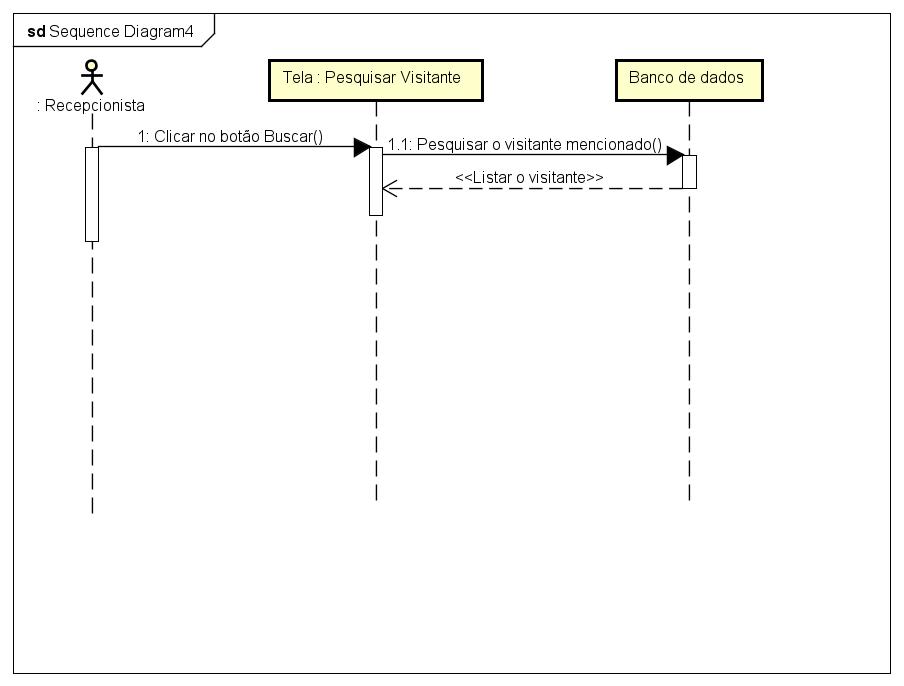
# Figura 5 – Diagrama de sequência alterar visitante.



Fonte: Autoria Própria, 2022.

# Diagrama de sequência pesquisar visitante

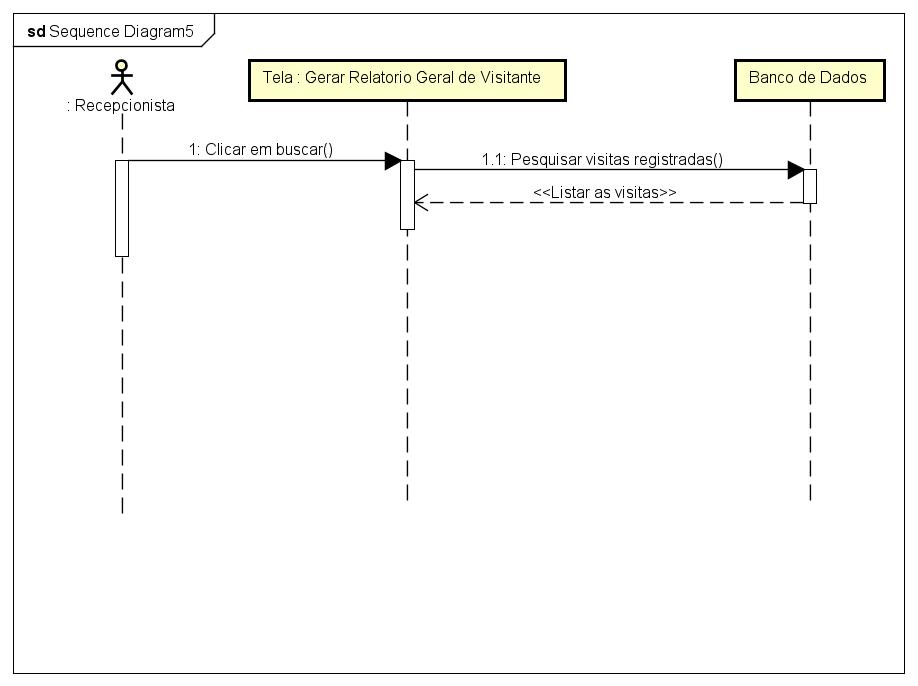
# Figura 6 – Diagrama de sequência pesquisar visitante.



Fonte: Autoria Própria, 2022.

# Diagrama de sequência gerar relatório geral de visitante

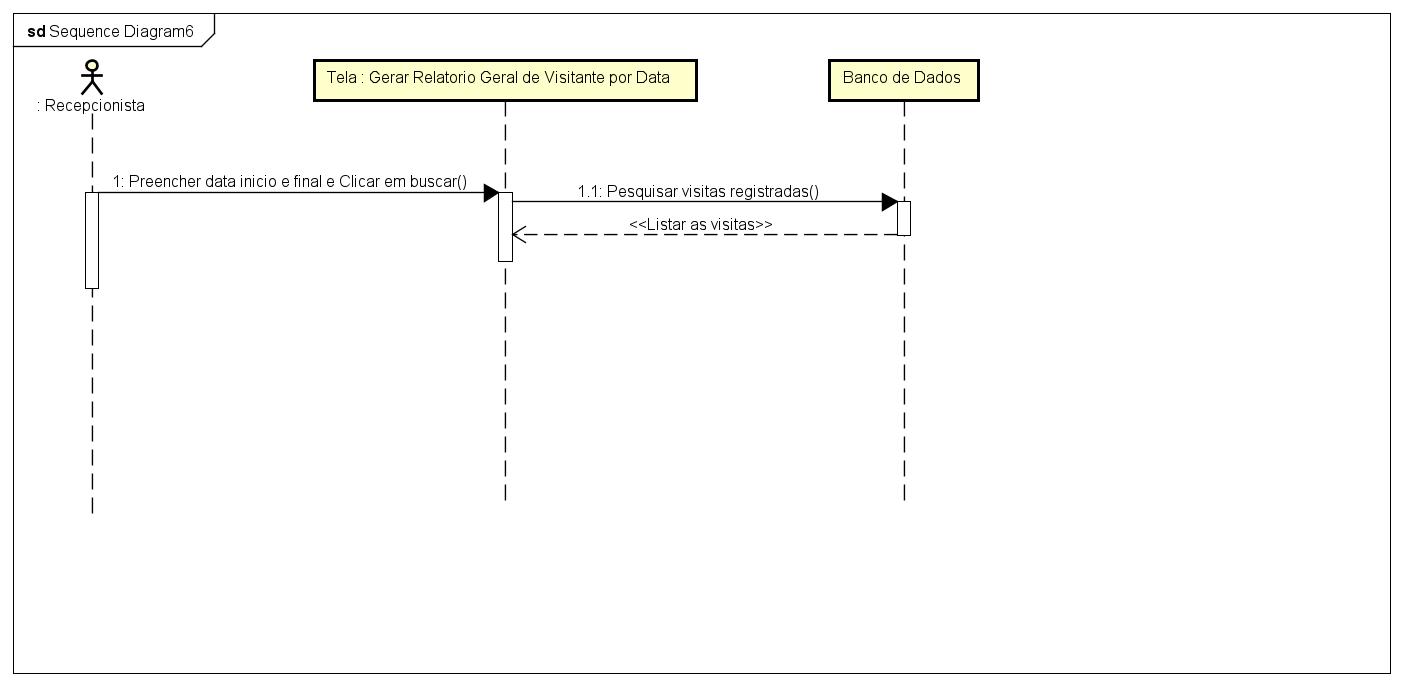
# Figura 7 – Diagrama de sequência gerar relatório geral de visitante



Fonte: Autoria Própria, 2022.

# Diagrama de sequência gerar relatório geral de visitante por data

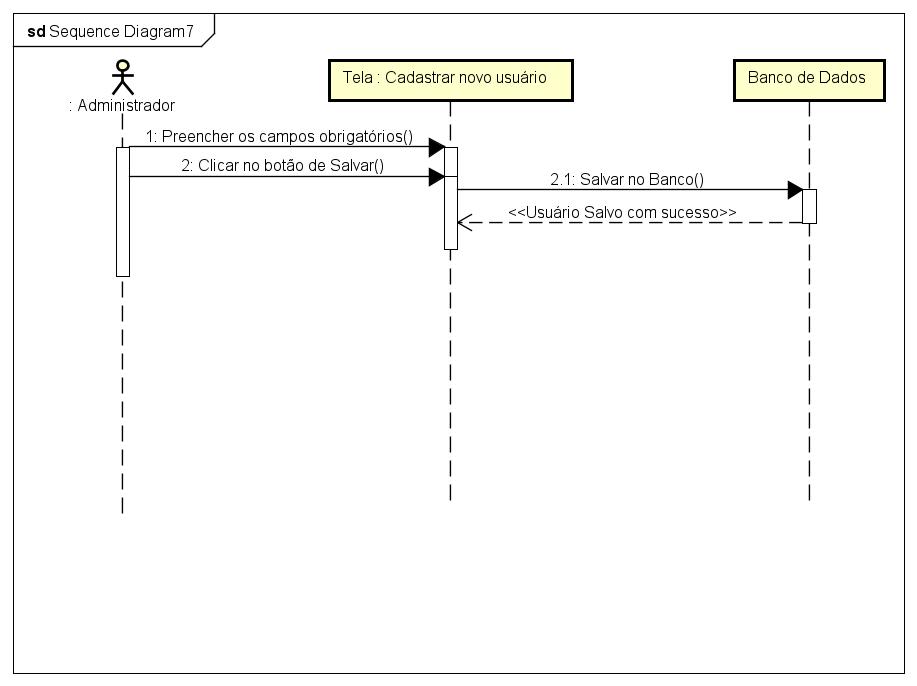
# Figura 8 – Diagrama de sequência gerar relatório geral de visitante por data



Fonte: Autoria Própria, 2022.

# Diagrama de sequência de cadastro de novo usuário

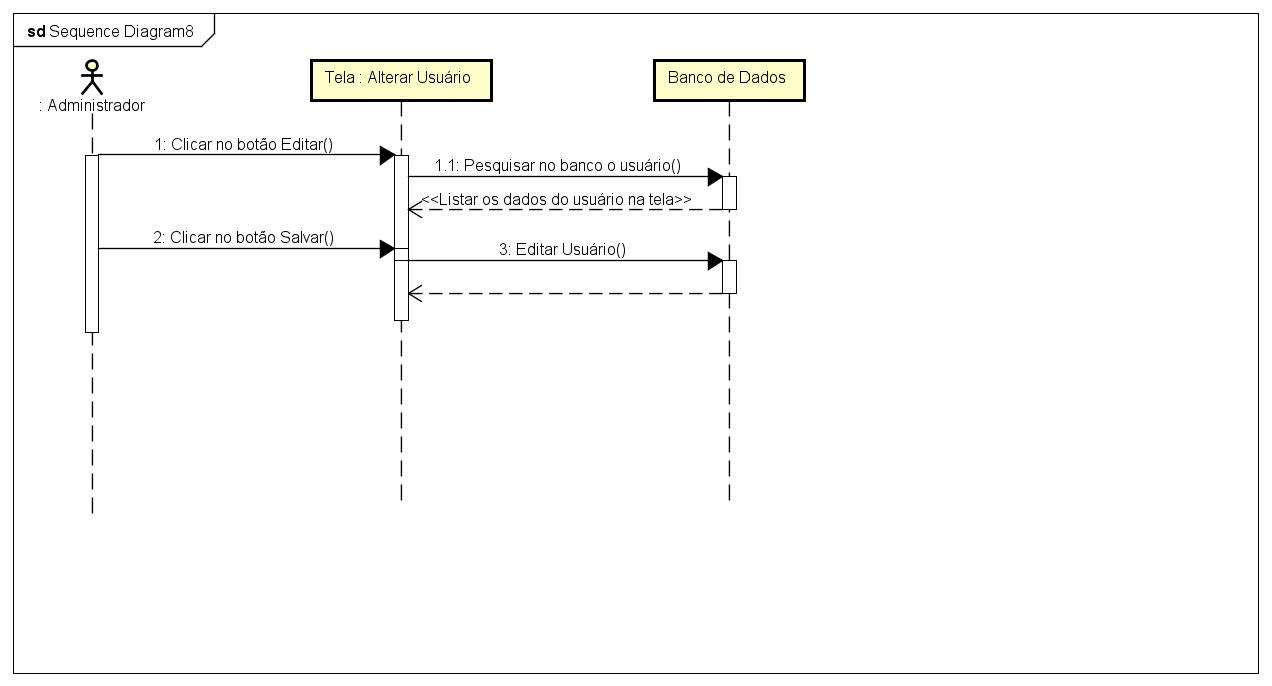
# Figura 9 – Diagrama de sequência de cadastro de novo usuário



Fonte: Autoria Própria, 2022.

# Diagrama de sequência de alterar usuário

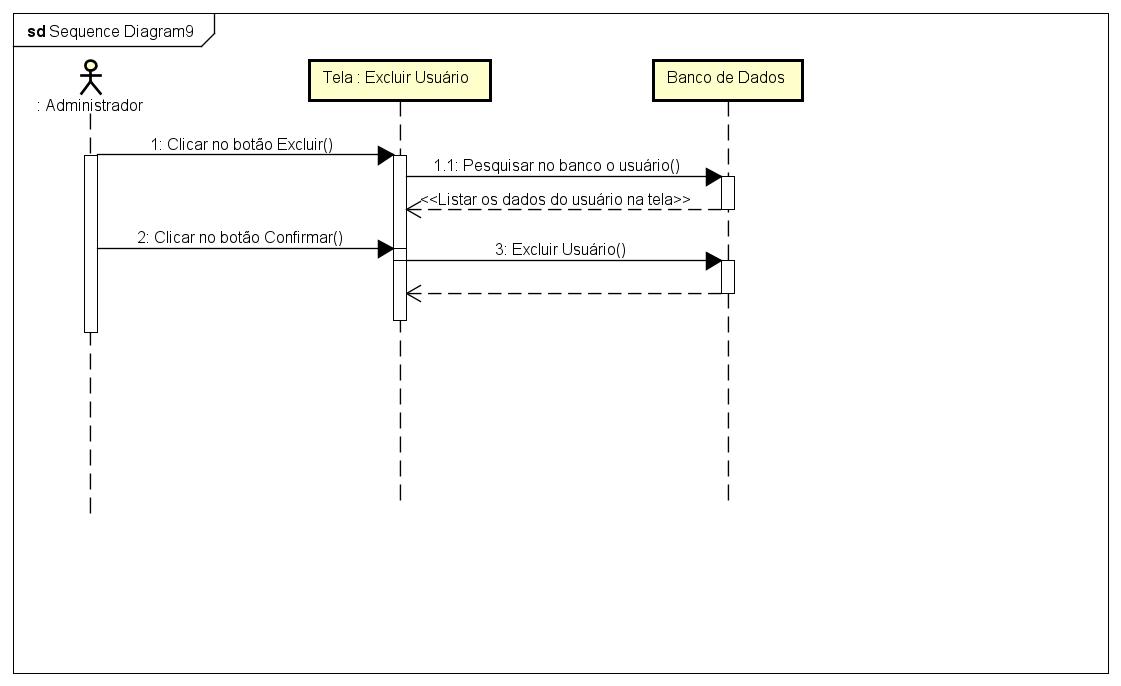
# Figura 10 – Diagrama de sequência de alterar usuário



Fonte: Autoria Própria, 2022.

# Diagrama de sequência de excluir usuário

# Figura 11 – Diagrama de sequência de excluir usuário

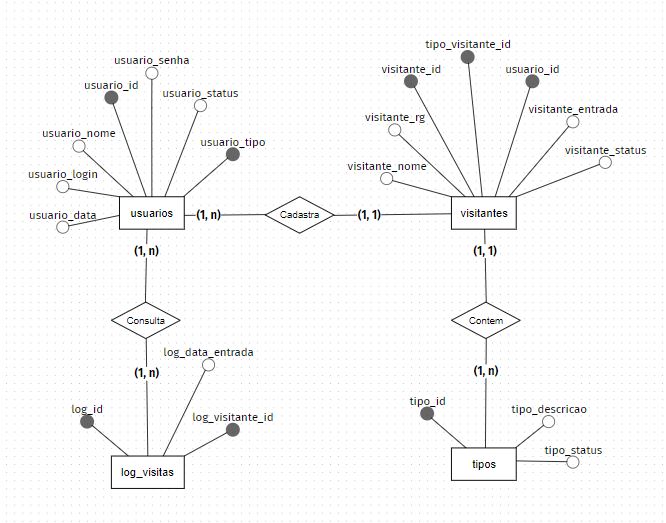


Fonte: Autoria Própria, 2022.

# 4.2 MODELO DE ENTIDADE E RELACIONAMENTO - MER

# 4.2.1 Modelo conceitual

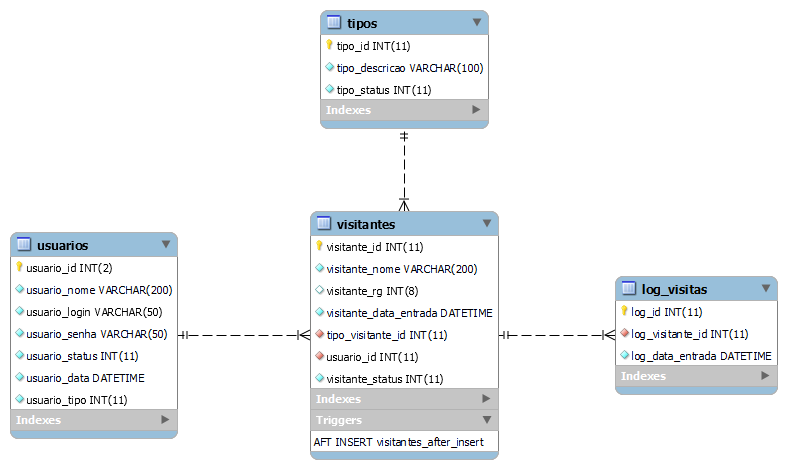
**Figura 12 – Modelo conceitual.**



Fonte: Autoria Própria, 2022.

# 4.2.2 Modelo lógico

**Figura 13 – Modelo lógico.**



**Figura 14 – Representação de relacionamentos**



Fonte: Autoria Própria, 2022.

# 4.2.3 Modelo físico

CREATE DATABASE IF NOT EXISTS `checkintour` CHARACTER SET armscii8 COLLATE armscii8\_bin;

USE `checkintour`;

CREATE TABLE IF NOT EXISTS `log\_visitas` (

`log\_id` int(11) NOT NULL AUTO\_INCREMENT,

`log\_visitante\_id` int(11) NOT NULL DEFAULT 0,

`log\_data\_entrada` datetime NOT NULL,

PRIMARY KEY (`log\_id`),

KEY `FK\_VISITANTE\_ID` (`log\_visitante\_id`),

CONSTRAINT `FK\_VISITANTE\_ID` FOREIGN KEY (`log\_visitante\_id`) REFERENCES `visitantes` (`visitante\_id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB AUTO\_INCREMENT=44 DEFAULT CHARSET=armscii8 COLLATE=armscii8\_bin COMMENT='Log de visita registra uma visita de um visitante ao ser cadastrado.';

CREATE TABLE IF NOT EXISTS `tipos` (

`tipo\_id` int(11) NOT NULL AUTO\_INCREMENT,

`tipo\_descricao` varchar(100) COLLATE armscii8\_bin NOT NULL,

`tipo\_status` int(11) NOT NULL DEFAULT 1,

PRIMARY KEY (`tipo\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=armscii8 COLLATE=armscii8\_bin COMMENT='tabela responsavel por armazenar se um visitante é estrangeiro ou brasileiro';

CREATE TABLE IF NOT EXISTS `usuarios` (

`usuario\_id` int(2) NOT NULL AUTO\_INCREMENT,

`usuario\_nome` varchar(200) COLLATE armscii8\_bin NOT NULL,

`usuario\_login` varchar(50) COLLATE armscii8\_bin NOT NULL,

`usuario\_senha` varchar(50) COLLATE armscii8\_bin NOT NULL,

`usuario\_status` int(11) NOT NULL DEFAULT 1,

`usuario\_data` datetime NOT NULL,

`usuario\_tipo` int(11) NOT NULL COMMENT '1 = admin, 2 = recep',

PRIMARY KEY (`usuario\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=armscii8 COLLATE=armscii8\_bin COMMENT='tabela de usuario para definicões de logins de acesso ao sistema.';

CREATE TABLE IF NOT EXISTS `visitantes` (

`visitante\_id` int(11) NOT NULL AUTO\_INCREMENT,

`visitante\_nome` varchar(200) COLLATE armscii8\_bin NOT NULL,

`visitante\_rg` int(8) DEFAULT NULL,

`visitante\_data\_entrada` datetime NOT NULL,

`tipo\_visitante\_id` int(11) NOT NULL,

`usuario\_id` int(11) NOT NULL,

`visitante\_status` int(11) NOT NULL DEFAULT 1,

PRIMARY KEY (`visitante\_id`),

KEY `FK\_visitantes\_usuarios` (`usuario\_id`),

KEY `FK\_tipo\_visitante` (`tipo\_visitante\_id`),

CONSTRAINT `FK\_tipo\_visitante` FOREIGN KEY (`tipo\_visitante\_id`) REFERENCES `tipos` (`tipo\_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `FK\_visitantes\_usuarios` FOREIGN KEY (`usuario\_id`) REFERENCES `usuarios` (`usuario\_id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB AUTO\_INCREMENT=21 DEFAULT CHARSET=armscii8 COLLATE=armscii8\_bin COMMENT='tabela de visitantes';

SET @OLDTMP\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,NO\_ENGINE\_SUBSTITUTION';

DELIMITER //

CREATE TRIGGER `visitantes\_after\_insert` AFTER INSERT ON `visitantes` FOR EACH ROW BEGIN

INSERT INTO log\_visitas

( log\_visitante\_id,

log\_data\_entrada

)

VALUES

( NEW.visitante\_id,

SYSDATE()

);

END//

DELIMITER ;

SET SQL\_MODE=@OLDTMP\_SQL\_MODE;

# 4.3 PROJETO DE BANCO DE DADOS

# 4.3.1 Dicionário de dados

**Tabela**: usuarios

**Objetivos:** Armazenar dados referentes à tabela usuarios.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nome Coluna** | **Tipo** | **Tam** | **Null** | **PK** | **AI** | **FK** | **Tab. Origem** | **Descrição**  **Domínio** |
| usuario\_id | Int | 2 | N | S | S | N | N | Atributo único que define o usuário. |
| usuario\_nome | Varchar | 200 | N | N | N | N | N | Nome do usuário. |
| usuario\_login | Varchar | 50 | N | N | N | N | N | Identificador do login do usuário. |
| usuario\_senha | Varchar | 50 | N | N | N | N | N | Identificador da senha do usuário. |
| usuario\_status | Int | 11 | N | N | N | N | N | Armazena o status do usuário 1 ou 2. |
| usuario\_data | Datetime |  | N | N | N | N | N | Data que o usuário foi cadastrado. |
| usuario\_tipo | Int | 11 | N | N | N | N | N | Identifica se o usuário é administrador ou recepcionista. |

**Tabela**: visitantes

**Objetivos:** Armazenar dados referentes à tabela visitantes.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nome Coluna** | **Tipo** | **Tam** | **Null** | **PK** | **AI** | **FK** | **Tab. Origem** | **Descrição**  **Domínio** |
| visitante\_id | Int | 11 | N | S | S | N | N | Atributo único que define o visitante. |
| visitante\_nome | Varchar | 200 | N | N | N | N | N | Nome do visitante. |
| visitante\_rg | Varchar | 8 | N | N | N | N | N | Identificador da identidade do visitante. |
| visitante\_data\_entrada | Datetime |  | N | N | N | N | N | Data que o visitante foi cadastrado. |
| tipo\_visitante\_id | Int | 11 | N | N | N | S | tipos | Tipo de visitante, brasileiro ou estrangeiro. |
| usuario\_id | Int | 11 | N | N | N | S | usuarios | Usuário que realizou o cadastro. |
| visitante\_status | Int | 11 | N | N | N | N | N | Armazena o status do visitante 1 ou 2. |

**Tabela**: tipos

**Objetivos:** Armazenar dados referentes à tabela tipos.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nome Coluna** | **Tipo** | **Tam** | **Null** | **PK** | **AI** | **FK** | **Tab. Origem** | **Descrição**  **Domínio** |
| tipo\_id | Int | 11 | N | S | S | N | N | Atributo único que define o tipo. |
| tipo\_descricao | Varchar | 100 | N | N | N | N | N | Descricação do tipo. |
| tipo\_status | Int | 11 | N | N | N | N | N | Armazena o status do tipo 1 ou 2. |

**Tabela**: log\_visitas

**Objetivos:** Armazenar dados referentes à tabela log\_visitas.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nome Coluna** | **Tipo** | **Tam** | **Null** | **PK** | **AI** | **FK** | **Tab. Origem** | **Descrição**  **Domínio** |
| log\_id | Int | 11 | N | S | S | N | N | Atributo único que define o log. |
| log\_visitante\_id | Int | 11 | N | N | N | S | visitantes | Id do visitante. |
| log\_data\_entrada | Datetime |  | N | N | N | N | N | Armazena a data da entrada do visitante. |

# 4.3.2 Mapeamento

|  |  |
| --- | --- |
| **PK(#)** – PRIMARY KEY (CHAVE PRIMÁRIA). | |
| TABELA: | usuarios. |
| ATRIBUTOS: | (usuario\_id#,usuario\_nome, usuario\_login, usuario\_senha, usuario\_status, usuario\_data, usuario\_tipo). |
| Obs: | O # representa a chave primária da tabela usuarios. |

|  |  |
| --- | --- |
| **PK(#)** – PRIMARY KEY (CHAVE PRIMÁRIA).  **FK(##)** – FOREING KEY (CHAVE ESTRANGEIRA) | |
|  | |
| TABELA: | visitantes. |
| ATRIBUTOS: | (visitante\_id#, visitante\_nome, vistante\_rg, visitante\_data\_entrada, tipo\_visitante\_id##, usuario\_id##, visitante\_status). |
| Obs: | O # representa a chave primária da tabela visitantes.O ## representa a chave estrangeira da tabela tipos e usuarios. |

|  |  |
| --- | --- |
| **PK(#)** – PRIMARY KEY (CHAVE PRIMÁRIA). | |
|  | |
| TABELA: | tipos. |
| ATRIBUTOS: | (tipo\_id#, tipo\_descricao, tipo\_status). |
| Obs: | O # representa a chave primária da tabela tipos. |

|  |  |
| --- | --- |
| **PK(#)** – PRIMARY KEY (CHAVE PRIMÁRIA). | |
|  | |
| TABELA: | log\_visitas. |
| ATRIBUTOS: | (log\_id#, log\_visitante\_id##, log\_data\_entrada). |
| Obs: | O # representa a chave primária da tabela log\_visitas.O ## representa a chave estrangeira da tabela visitantes. |

# CONSIDERAÇÕES FINAIS

No período do Projeto Final ll, foi desenvolvido o protótipo ‘CHECK-IN TOUR’, para facilitar o controle de entrada de visitantes nos pontos turísticos, devido à grande demanda de visitantes, muitos pontos turísticos não têm o controle de quantas pessoas passam por ali, pois o controle é feito manualmente.

Levando-se em consideração, foi desenvolvido o protótipo utilizando um padrão MVC (Modelo-Visão-Controle). No processo de desenvolvimento foi utilizado o SCRUM (Metodologia Ágil para Gestão e Planejamento de Projetos de Software), para agilizar o processo sem perder o foco. O protótipo foi feito utilizando a linguagem UML (Linguagem de Modelagem Unificada) e seu desenvolvimento foi através da linguagem de programação PHP (Personal Home Page).

Através do protótipo ‘CHECK-IN TOUR’, os pontos turísticos passam a ter um controle maior de seus visitantes podendo assim gerar relatórios precisos para saber em qual dia, mês ou ano obteve o maior ou menor número de visitas.

# REFERÊNCIAS BIBLIOGRÁFICAS

ALVAREZ, Miguel, **O que é HTML?**, 2004, Disponível em: <http://www.criarweb.com/artigos/7.php>. Acesso em: 19 de setembro de 2022.

PEREIRA, Ana P., **O que é CSS?**, 2009. Disponível em: <https://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm>. Acesso em: 15 de setembro de 2022.

CARVALHO, Flávia, **WAMPSERVER - Autoria e Designer na Internet I**, 1. ed. Rio Grande sdo SUl: Elsevier, 2010.

COAR, Ken, **Apache Guia Prático**, 1. Ed. Rio de Janeiro: Alta Books,2008.

DATE, C.J.; **Introdução a Sistemas De Bancos De Dados**, 8. ed. Rio de Janeiro: Elsevier, 2004.

DATE, C.J.; **Introdução a Sistemas de Bancos de Dados**, tradução da 4. ed. norte-americana, Editora Campus, 1991.

ELMASRI, Ramez; NAVATHE, ShamkantB, **Sistemas de Banco de Dados**, 6. ed. São Paulo: Addison Wesley, 2011.

BOLZAN, Willian, **DbDesigner**, 2009, Disponível em: <http://www.devmedia.com.br/dbdesigner-modelagem-e-implementacao-de-banco-de-dados/30897>. Acesso em: 15 de setembro de 2022.

GUEDES, Gilleanes T. A.; **Uml 2 – Guia Prático**, 2. ed. São Paulo: Novatec, 2014.

KORTH, H.F. e SILBERSCHATZ, A.; **Sistemas de Bancos de Dados**, Makron Books, 2. Ed. revisada, 1994.

LAMIM, Jonathan, **MVC - O padrão de arquitetura de software**, Santa Cruz do Sul: Oficinadanet, 2012, Disponível em: <https://www.oficinadanet.com.br/artigo/1687/mvc\_-\_o\_padrao\_de\_arquitetura\_de\_software>. Acesso em: 15 de setembro de 2022.

MILANI, Andre, **Construindo aplicações Web em PHP e MySql**, 1. ed. São Paulo: Novatec, 2010

MILANI, Andre, **Construindo aplicações Web em PHP e MySql,** 2. ed. São Paulo: Novatec, 2016.

PINTO, Pedro, **Sublime**, 2013, Disponível em: <https://pplware.sapo.pt/software/sublime-text-e-este-o-melhor-editor-de-texto-do-mundo/>. Acesso em: 15 de setembro de 2022.

SCHWABER, Ken, **Scrum**, 2004, Disponível em: <http://www.desenvolvimentoagil.com.br/scrum/>. Acesso em: 15 de setembro de 2022.

SEABRA, Bruno, **O QUE É O ASTAH?,** 2015, Disponível em:<http://www.startupsstars.com/2015/10/o-que-e-o-astah-posttecnico-por-bruno-seabra/>. Acesso em: 15 de setembro de 2022.

SOUZA, Hermes, **Criando, Inserindo e Exibindo Php + MySql**, 2008. Disponível em: <www.argohost.net>. Acesso em: 15 de setembro de 2022.

WELLCARE, **Controle de entrada e saída de funcionários e visitantes**,Disponível em: <http://www.wellcare.com.br/solucoes/controle-de-entrada-e-saida-de-funcionarios-e-visitantes>. Acesso em: 15 de setembro de 2022.

# APÊNDICE

**Apêndice A – Cronograma do Desenvolvimento do Projeto.**

|  |
| --- |
| ANO 2022 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MÊS | Jul | | | | AGO | | | | SET | | |
| SEMANA | **1°** | **2°** | **3°** | **4°** | **1°** | **2°** | **3°** | **4°** | **1°** | **2°** | **3°** |
| Escolha do Tema | **x** | **x** |  |  |  |  |  |  |  |  |  |
| Justificativa |  | **x** |  |  |  |  |  |  |  |  |  |
| Objetivos Gerais e Específicos |  | **x** |  |  |  |  |  |  |  |  |  |
| Referenciais Teóricos (Ferramentas) |  | **x** | **x** |  |  |  |  |  |  |  |  |
| Metodologia |  | **x** | **x** |  |  |  |  |  |  |  |  |
| Análise do Sistema |  |  |  | **x** | **x** |  |  |  |  |  |  |
| Projeto do Banco de Dados |  |  |  |  |  | **x** | **x** |  |  |  |  |
| Projeto de Interfaces |  |  |  |  |  | **x** | **x** |  |  |  |  |
| Desenvolvimento do Sistema |  |  |  |  |  |  | **x** | **x** | **x** | **x** | **x** |