|  |  |
| --- | --- |
| **Curso:** | - |
| **Período/Turma:** | - |
| **Grupo:** | - |

**Integrantes do grupo:**

|  |  |  |
| --- | --- | --- |
| **RA** | **Nome completo** | **e-mail** |
| 1111392221006 | Daniel Viegas Cardamoni |  |
| 1111392221043 | Eloah Baracho dos Santos |  |
| 1111392221007  1111392221015 | Felipe Selva Rocha Alves  Leonardo da Silva Lima |  |

|  |
| --- |
| **Título do Projeto e/ou logotipo** |

**Objetivo do Projeto**

|  |
| --- |
| Descrição da finalidade do projeto, alinhada a especificação. Lembrando que o mesmo tem que ser realista, consistente e claro. |

O sistema de reservas de restaurantes possui a finalidade de proporcionar aos clientes de um restaurante maior conforto e organização no momento de suas refeições. Um sistema capaz de reduzir filas, tempo de espera e problemas com quantidades de assentos se faz necessário, uma vez que o antro gastronômico no Brasil tem crescido e se expandido.

**Descrição do Projeto**

|  |
| --- |
| Detalhes do projeto (*storytelling)* e da especificação dos requisitos de informação e regras de negócio do domínio do problema (Requisitos Func. e Não Func.) |

A necessidade de solucionar o problema da ausência de organização nos restaurantes de grandes metrópoles é urgente, tendo e vista que fatores negativos na experiência e no atendimento desses estabelecimentos definem sua sobrevida e, consequentemente, o seu sucesso.

|  |  |
| --- | --- |
| O problema de | Ausência de organização e disponibilidade (superlotação). |
| afeta | Clientes. |
| cujo impacto é | A péssima experiência e atendimento. |
| uma boa solução seria | Sistema de reservas capaz de organizar clientes por horários, número de assentos e número de mesas. |

**Requisitos Funcionais**

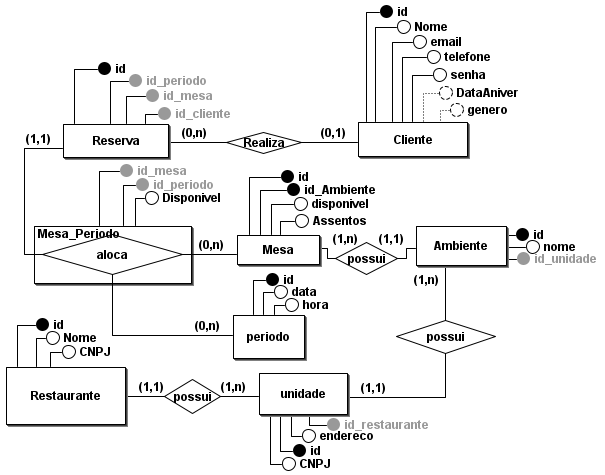
|  |  |  |  |
| --- | --- | --- | --- |
| **ID do**  **Requisito** | **Nível de**  **Prioridade** (**A / M / B)** | **Descrição do Requisito** | **Categoria do Requisito** |
| RF01 | *A* | *Sistema deve cadastrar cliente.* | *Autenticação* |
| RF02 | *A* | *Usuário deve ser identificado pelo login e senha.* | *Autenticação* |
| RF03 | *A* | *Usuário deve ser capaz de realizar as reservas.* | *Controle de usuário* |
| RF04 | *A* | *Usuário deve ser capaz de gerenciar suas reservas (editar e excluir).* | *Controle de usuário* |

**Regras de Negócio**

|  |  |
| --- | --- |
| RN01 | *Mesas Especificas podem apresentar Status Disponível ou Indisponível.* |
| RN02 | *Para reservar uma mesa, o mínimo de assentos precisará, necessariamente, ser 1.* |
| RN03 | *Fila de Espera não garantirá acesso a mesas.* |
| RN04 | *Cancelamentos só estarão disponíveis até dois dias antes da data marcada, a fim de manter a organização e coerência na disponibilidade das mesas. Caso o cliente não compareça, será necessário pagar uma taxa para efetuar uma nova reserva.* |
| RN05 | *A indisponibilidade de uma mesa possuirá o prazo de 2 (duas) horas. Após esse período, a mesa retornará ao status disponível.* |

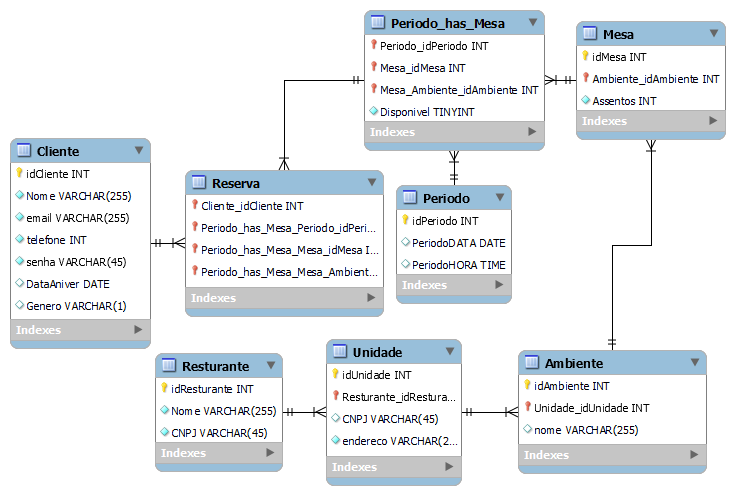
**Modelagem Conceitual**

|  |
| --- |
| Descrição das entidades necessárias;  Descrição de cada atributo necessário para cada entidade  Diagrama de Entidade e Relacionamento (DER) |



**Modelagem Lógica**

|  |
| --- |
| Determinar o identificador de cada tabela  Determinar o relacionamento de cada Tabela  Diagrama Modelo Relacional (MR) |



**Modelagem Física**

|  |
| --- |
| Script SQL da criação das Tabelas, Atributos, Chaves primárias e Estrangeiras (DDL) |

-- -----------------------------------------------------

-- Schema ReservasOnline

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `ReservasOnline` DEFAULT CHARACTER SET utf8 ;

USE `ReservasOnline` ;

-- -----------------------------------------------------

-- Table `ReservasOnline`.`Cliente`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `ReservasOnline`.`Cliente` (

`idCliente` INT NOT NULL AUTO\_INCREMENT,

`Nome` VARCHAR(255) NOT NULL,

`email` VARCHAR(255) NOT NULL,

`telefone` INT NOT NULL,

`senha` VARCHAR(45) NOT NULL,

`DataAniver` DATE NULL,

`Genero` VARCHAR(1) NULL,

PRIMARY KEY (`idCliente`),

UNIQUE INDEX `idCliente\_UNIQUE` (`idCliente` ASC) ,

UNIQUE INDEX `email\_UNIQUE` (`email` ASC) );

-- -----------------------------------------------------

-- Table `ReservasOnline`.`Resturante`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `ReservasOnline`.`Resturante` (

`idResturante` INT NOT NULL,

`Nome` VARCHAR(255) NOT NULL,

`CNPJ` VARCHAR(45) NOT NULL,

PRIMARY KEY (`idResturante`));

-- -----------------------------------------------------

-- Table `ReservasOnline`.`Unidade`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `ReservasOnline`.`Unidade` (

`idUnidade` INT NOT NULL,

`Resturante\_idResturante` INT NOT NULL,

`CNPJ` VARCHAR(45) NULL,

`endereco` VARCHAR(255) NOT NULL,

PRIMARY KEY (`idUnidade`, `Resturante\_idResturante`),

INDEX `fk\_Unidade\_Resturante\_idx` (`Resturante\_idResturante` ASC) ,

CONSTRAINT `fk\_Unidade\_Resturante`

FOREIGN KEY (`Resturante\_idResturante`)

REFERENCES `ReservasOnline`.`Resturante` (`idResturante`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

-- -----------------------------------------------------

-- Table `ReservasOnline`.`Ambiente`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `ReservasOnline`.`Ambiente` (

`idAmbiente` INT NOT NULL,

`Unidade\_idUnidade` INT NOT NULL,

`nome` VARCHAR(255) NULL,

PRIMARY KEY (`idAmbiente`, `Unidade\_idUnidade`),

INDEX `fk\_Ambiente\_Unidade1\_idx` (`Unidade\_idUnidade` ASC) ,

CONSTRAINT `fk\_Ambiente\_Unidade1`

FOREIGN KEY (`Unidade\_idUnidade`)

REFERENCES `ReservasOnline`.`Unidade` (`idUnidade`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

-- -----------------------------------------------------

-- Table `ReservasOnline`.`Mesa`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `ReservasOnline`.`Mesa` (

`idMesa` INT NOT NULL,

`Ambiente\_idAmbiente` INT NOT NULL,

`Assentos` INT NOT NULL,

PRIMARY KEY (`idMesa`, `Ambiente\_idAmbiente`),

INDEX `fk\_Mesa\_Ambiente1\_idx` (`Ambiente\_idAmbiente` ASC) ,

CONSTRAINT `fk\_Mesa\_Ambiente1`

FOREIGN KEY (`Ambiente\_idAmbiente`)

REFERENCES `ReservasOnline`.`Ambiente` (`idAmbiente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

-- -----------------------------------------------------

-- Table `ReservasOnline`.`Periodo`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `ReservasOnline`.`Periodo` (

`idPeriodo` INT NOT NULL,

`PeriodoDATA` DATE NULL,

`PeriodoHORA` TIME NULL,

PRIMARY KEY (`idPeriodo`));

-- -----------------------------------------------------

-- Table `ReservasOnline`.`Periodo\_has\_Mesa`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `ReservasOnline`.`Periodo\_has\_Mesa` (

`Periodo\_idPeriodo` INT NOT NULL,

`Mesa\_idMesa` INT NOT NULL,

`Mesa\_Ambiente\_idAmbiente` INT NOT NULL,

`Disponivel` TINYINT NOT NULL,

PRIMARY KEY (`Periodo\_idPeriodo`, `Mesa\_idMesa`, `Mesa\_Ambiente\_idAmbiente`),

INDEX `fk\_Periodo\_has\_Mesa\_Mesa1\_idx` (`Mesa\_idMesa` ASC, `Mesa\_Ambiente\_idAmbiente` ASC) ,

INDEX `fk\_Periodo\_has\_Mesa\_Periodo1\_idx` (`Periodo\_idPeriodo` ASC) ,

CONSTRAINT `fk\_Periodo\_has\_Mesa\_Periodo1`

FOREIGN KEY (`Periodo\_idPeriodo`)

REFERENCES `ReservasOnline`.`Periodo` (`idPeriodo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Periodo\_has\_Mesa\_Mesa1`

FOREIGN KEY (`Mesa\_idMesa` , `Mesa\_Ambiente\_idAmbiente`)

REFERENCES `ReservasOnline`.`Mesa` (`idMesa` , `Ambiente\_idAmbiente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

-- -----------------------------------------------------

-- Table `ReservasOnline`.`Reserva`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `ReservasOnline`.`Reserva` (

`Cliente\_idCliente` INT NOT NULL,

`Periodo\_has\_Mesa\_Periodo\_idPeriodo` INT NOT NULL,

`Periodo\_has\_Mesa\_Mesa\_idMesa` INT NOT NULL,

`Periodo\_has\_Mesa\_Mesa\_Ambiente\_idAmbiente` INT NOT NULL,

PRIMARY KEY (`Cliente\_idCliente`, `Periodo\_has\_Mesa\_Periodo\_idPeriodo`, `Periodo\_has\_Mesa\_Mesa\_idMesa`, `Periodo\_has\_Mesa\_Mesa\_Ambiente\_idAmbiente`),

INDEX `fk\_Cliente\_has\_periodo\_Cliente1\_idx` (`Cliente\_idCliente` ASC) ,

INDEX `fk\_Reserva\_Periodo\_has\_Mesa1\_idx` (`Periodo\_has\_Mesa\_Periodo\_idPeriodo` ASC, `Periodo\_has\_Mesa\_Mesa\_idMesa` ASC, `Periodo\_has\_Mesa\_Mesa\_Ambiente\_idAmbiente` ASC) ,

CONSTRAINT `fk\_Cliente\_has\_periodo\_Cliente1`

FOREIGN KEY (`Cliente\_idCliente`)

REFERENCES `ReservasOnline`.`Cliente` (`idCliente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Reserva\_Periodo\_has\_Mesa1`

FOREIGN KEY (`Periodo\_has\_Mesa\_Periodo\_idPeriodo` , `Periodo\_has\_Mesa\_Mesa\_idMesa` , `Periodo\_has\_Mesa\_Mesa\_Ambiente\_idAmbiente`)

REFERENCES `ReservasOnline`.`Periodo\_has\_Mesa` (`Periodo\_idPeriodo` , `Mesa\_idMesa` , `Mesa\_Ambiente\_idAmbiente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);