

Sistema InLine - Documentação

Projeto de MAC0439

Laboratório de Banco de Dados

Segundo semestre de 2018

Matheus Tavares Bernardino - N°USP: 9292987

Lucas Seiki Oshiro - N°USP: 9298228

Marcos Vinicius do Carmo Sousa - N°USP: 9298274

Sobre o Sistema

Introdução

O sistema InLine se propõe a facilitar o processo de gerenciamento de filas em estabelecimentos. Em vez de filas presenciais, geralmente gerenciadas por tickets em papel ou pagers vibratórios, o sistema traz filas virtualizadas, onde os clientes podem acompanhar sua posição em tempo real sem a necessidade de estar fisicamente no estabelecimento desejado.

Funcionalidades e Características Principais

O sistema tem por finalidade principal a execução das seguintes funcionalidades e características:

- Implementação de filas por 'ordem de chegada' e por 'hora marcada'.
 - Filas por 'ordem de chegada' representam as filas tradicionais, onde os usuários entram na última posição e são chamados a partir da primeira. A ordenação da fila, porém, além da ordem de chegada, leva em conta uma combinação de atributos do cliente, de modo a tornar a fila mais inteligente.
 - Filas de 'hora marcada', constituem-se de uma sequência de usuários com ordem pré-agendada para atendimento no estabelecimento (como um agendamento tradicional em um consultório médico, por exemplo). A diferença nesta categoria é que o usuário é atualizado em tempo real sobre seu agendamento. Se o estabelecimento, por alterações no tempo previsto de atendimento dos clientes anteriores, precisar atrasar o agendamento de um usuário (ou existe possibilidade de antecipar), o usuário é notificado.
- Promover filas com 'acesso prioritário' para idosos com idade superior a 60 anos, aposentados, gestantes, pessoas com crianças de colo, deficientes e obesos.
- Permitir o 'acesso premium' para usuários que desejarem pagar uma taxa mensal ou comprarem créditos. No primeiro caso, os usuários têm 'acesso premium' a todas as filas com suporte a premium. No segundo, os usuários podem comprar créditos e gastá-los apenas nas filas com suporte a premium que desejarem (cada crédito dá direito a um acesso a uma fila). Cabe a cada estabelecimento decidir se suas filas terão suporte a 'acesso premium' ou não.
- Para clientes que chegarem a um estabelecimento sem possuir o app InLine ou acesso à internet, o sistema permitirá sua entrada na fila como um 'usuário não-cadastrado'. Sua entrada será feita pelo gerente do estabelecimento com o nome do cliente.

- Mostrar aos usuários, estabelecimentos próximos a ele e estabelecimentos onde as filas estão pequenas.
- Cada usuário na fila do tipo 'ordem de chegada' tem sua posição GPS monitorada de tempos em tempos, enquanto estiver na fila. Caso sua posição GPS não se aproxime do estabelecimento conforme sua vez for chegando na fila, ocorre uma penalização de posições.

Requisitos de Dados

Introdução

Planeja-se implementar as seguintes entidades:

- Usuário
- Empresa (especialização do usuário)
- Estabelecimento (especialização do usuário)
- Cliente
- Cliente cadastrado (especialização do usuário e de cliente)
- Cliente não-cadastrado (especialização de cliente)
- Cliente premium (especialização de cliente cadastrado)
- Serviço premium
- Serviço crédito (especialização de serviço premium)
- Serviço mensalidade (especialização de serviço premium)
- Fila (composição de sub-filas)
- Fila Agendada (especialização de fila)
- Agendamento
- Agendamento Concluído
- Fila Cronológica (especialização de fila)
- Entrada da fila cronológica
- Entrada concluída da fila cronológica
- Avaliação

Detalhamento das Entidades

- **Usuário**
Representa um usuário. Armazena o nome, email, senha e tipo, que pode ser 'empresa', 'estabelecimento' ou 'cliente'. O email identifica o usuário univocamente no sistema.
- **Empresa**

É uma especialização de usuário. Representa uma empresa (e uma conta empresarial) dentro do sistema. Possui CNPJ e endereço da sede (se aplicável). Possui um ou mais estabelecimentos.

- **Estabelecimento**

É uma especialização de usuário. Representa um estabelecimento, ou seja, uma unidade física de uma empresa (Também representa uma conta de administrador de estabelecimento). Armazena o endereço do estabelecimento, os horários (e dias) de funcionamento.

- **Cliente**

Representa um cliente de estabelecimentos. Armazena o celular do cliente e um identificador numérico que identifica o cliente univocamente no sistema.

- **Cliente cadastrado**

Representa um cliente cadastrado no sistema. Armazena a posição GPS do cliente, que deve valer NULL caso o usuário não esteja em nenhuma fila e ser atualizada de tempos em tempos se estiver em alguma fila.

- **Cliente não-cadastrado**

Representa um cliente não-cadastrado. Armazena nome do cliente.

- **Cliente premium**

Especialização de cliente cadastrado. Armazena número do cartão de crédito, data de vencimento e código de segurança.

- **Serviço premium**

Representa um serviço comprado por um usuário premium. Armazena a descrição do serviço (que pode ser 'creditos' ou 'mensalidade'), valor e data de aquisição, e um identificador.

- **Serviço crédito**

Especialização de serviço premium. Armazena a quantidade de créditos.

- **Serviço mensalidade**

Especialização de serviço premium. Armazena a data de cobrança para renovação e a data da mensalidade.

- **Fila**

Sequenciamento de clientes, em que todos os clientes têm interesse em um mesmo serviço. Tal sequenciamento permite que os clientes sejam atendidos

um por vez. Armazena o id do estabelecimento dono da fila, o id da fila, a data e hora de início, data e hora de fim e o tamanho da fila.

- **Fila Agendada**

Uma especialização de fila, na qual os horários de atendimento são escolhidos previamente. Desta forma, a posição é determinada pelo cliente e pela disponibilidade, e não de forma algorítmica. São armazenados os futuros agendamentos e agendamentos já concluídos.

- **Agendamento**

Representa o horário escolhido por um cliente para ser atendido. Armazena o horário escolhido, horário de criação do agendamento e a identificação do cliente que será atendido nesse horário. Só pode ser criado por um cliente.

- **Agendamento Concluído**

Representa um agendamento já finalizado. Armazena os mesmos atributos de agendamento e uma flag para indicar se o cliente foi atendido ou desistiu.

- **Fila Cronológica**

Uma especialização de fila, na qual a posição que os clientes ocupam não é agendada, dependendo portanto de sua localização, do momento da inscrição na fila, e de sua prioridade. Armazena uma flag para indicar se aceita clientes premium ou não e outra para indicar se aceita clientes preferenciais.

- **Entrada da fila cronológica**

Representa um cliente em uma fila cronológica. Armazena o id do cliente, sua distância física até o estabelecimento, sua data e hora de entrada na fila, sua posição na fila, se ele é premium e se é preferencial.

- **Entrada concluída da fila cronológica**

Representa um cliente que entrou na fila e saiu. Além de armazenar tudo que uma entrada não-concluída armazena, armazena uma flag para indicar se o cliente foi atendido ou desistiu e a data e hora em que saiu da fila.

- **Avaliação**

Representa uma avaliação de um cliente cadastrado sobre um estabelecimento. Guarda quantas estrelas e o comentário do cliente cadastrado.

Requisitos Funcionais

- Usuário
 - Um usuário pode se cadastrar no sistema, atualizar e deletar sua conta.
- Empresa
 - Uma empresa pode se cadastrar no sistema e atualizar suas informações. Suas informações públicas podem ser vistas por todos os usuários (cadastrados ou não).
- Estabelecimento
 - Um estabelecimento pode ser cadastrado no sistema por uma empresa suas informações só podem ser atualizadas pela empresa. Suas informações públicas podem ser vistas por todos os usuários (cadastrados ou não).
- Cliente
 - Um cliente é criado pelo sistema quando um usuário do tipo 'cliente' for criado ou quando um cliente não-cadastrado for criado. No primeiro caso, ele pode ser alterado pelo usuário e, no segundo, pode ser alterado pelo estabelecimento que criou o cliente não-cadastrado.
- Cliente não-cadastrado
 - Um cliente não cadastrado pode ser criado e alterado apenas por estabelecimentos. Ele deve ser deletado pelo sistema quando retirado da fila em que está.
- Cliente premium
 - Cliente premium é um cliente cadastrado que adquiriu um serviço premium.
- Serviço premium
 - Criado pelo sistema, não pode ser alterado nem deletado.
- Fila (e Fila cronológica e Fila Agendada)
 - Pode ser criada e modificada apenas pelo dono do estabelecimento responsável. Não pode ser excluída.
- Agendamento
 - Pode ser criado, excluído e modificado pelo cliente que está agendando e pelo dono do estabelecimento ao qual o agendamento está associado.
- Agendamento Concluído
 - É criado pelo sistema, automaticamente, quando um cliente conclui um agendamento. Não pode ser modificado nem excluído.
- Entrada da fila cronológica

- É criado, modificado e excluído pelo cliente ou dono do estabelecimento responsável por aquela fila.
- Entrada concluída da fila cronológica
 - É criado pelo sistema, automaticamente, quando um cliente sai de uma fila cronológica. Não pode ser modificado nem excluído.
- Avaliação
 - Pode ser criado, editado e excluído apenas por um cliente cadastrado.

Principais consultas

- Consultar usuário pelo email.
- Obter telefone dos clientes não cadastrados.
- Consultar cliente pelo seu id.
- Consultar fila pelo id.
- Eliminar entrada de uma fila.
- Inserir entrada em uma fila.
- Criação de cliente não-cadastrado por estabelecimentos.
- Listar os estabelecimentos de uma empresa.
- Selecionar as filas de um estabelecimento.
- Selecionar e atualizar distância [até estabelecimento] de um usuário em uma fila.
- Selecionar posição GPS de um estabelecimento.
- Selecionar e atualizar posição de um usuário em uma fila.
- Renovar serviços premium.
- Selecionar o lucro bruto mensal.
- Consultar se um cliente é premium ou preferencial.
- Consultar quantos créditos um cliente premium tem.
- Consultar o tipo de prioridade de um cliente preferencial
- Criação de estabelecimentos por empresas.
- Alterar os horários de funcionamento de um estabelecimento.
- Agendar horários na fila [agendada].
- Verificar qual é o dia da cobrança da mensalidade de um cliente premium.
- Verificar se uma fila cronológica aceita clientes premium.
- Verificar a data de aquisição e a discriminação de um serviço premium.
- Verificar a data de término do serviço premium por mensalidade
- Consultar quantidade de clientes em uma fila.
- Calcular tempo médio dos clientes em uma fila.

Justificativa para a modelagem realizada

- Relacional:

Optamos por usar o modelo relacional para representar os usuários (tanto clientes como empresa e estabelecimentos) porque a base de usuários tem uma estrutura mais bem definida com poucas alterações que contém algumas restrições. Por exemplo: um cliente não cadastrado pode ser um cliente preferencial ou não, mas não um cliente premium.

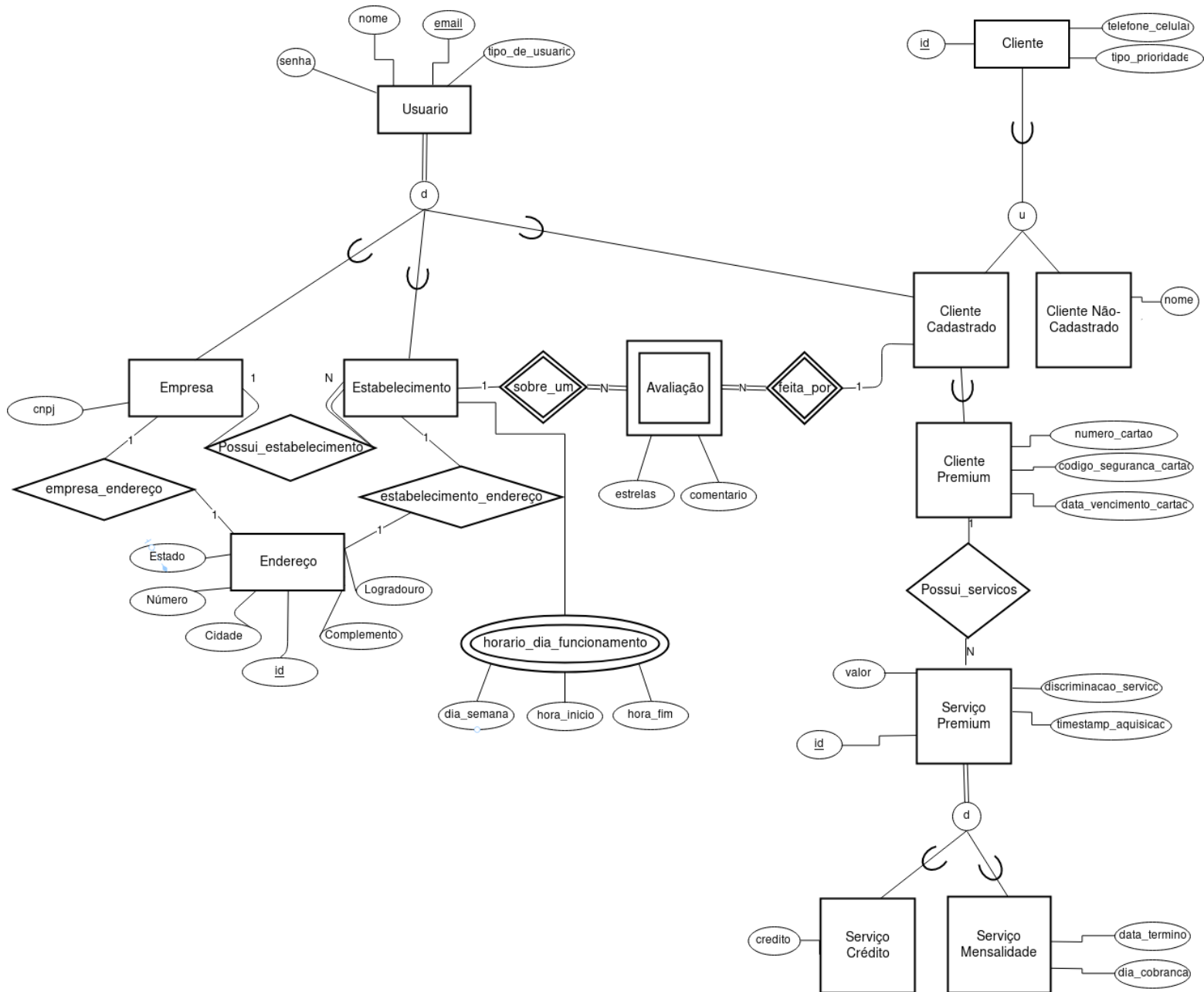
- NoSQL:

Armazenaremos as filas em NoSQL, visando obter maior performance. Como os dados das filas são independentes entre si, eles constituem um conjunto de dados propenso para ser distribuído de forma particionada, o que é possível com os NoSQL e proporciona aumento de performance (execuções paralelizadas).

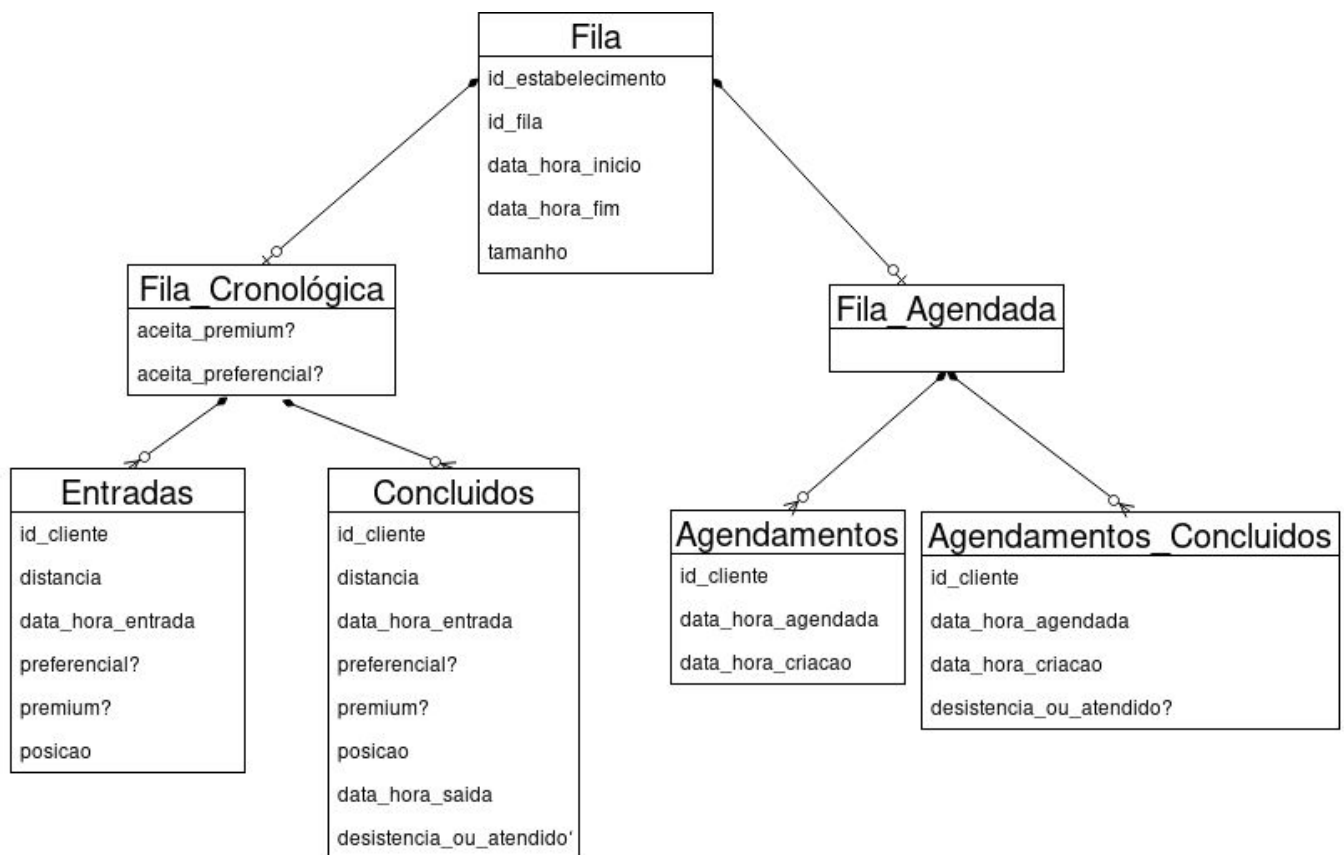
Além desse almejado escalonamento horizontal, como pretendemos manter todas as filas, mesmo já finalizadas (para fins estatísticos), é interessante usar um NoSQL por ser um SGBD adaptado para grandes volumes de dados.

Um ponto baixo é que os NoSQL têm melhor performance em consulta do que alteração. Mas projetamos as estruturas e o algoritmo de ordenação da fila cronológica pensando em contornar essa questão.

Modelo Conceitual (parte Relacional)



Modelo Conceitual (parte NoSQL)



O Algoritmo e Funcionamento das Filas Cronológicas

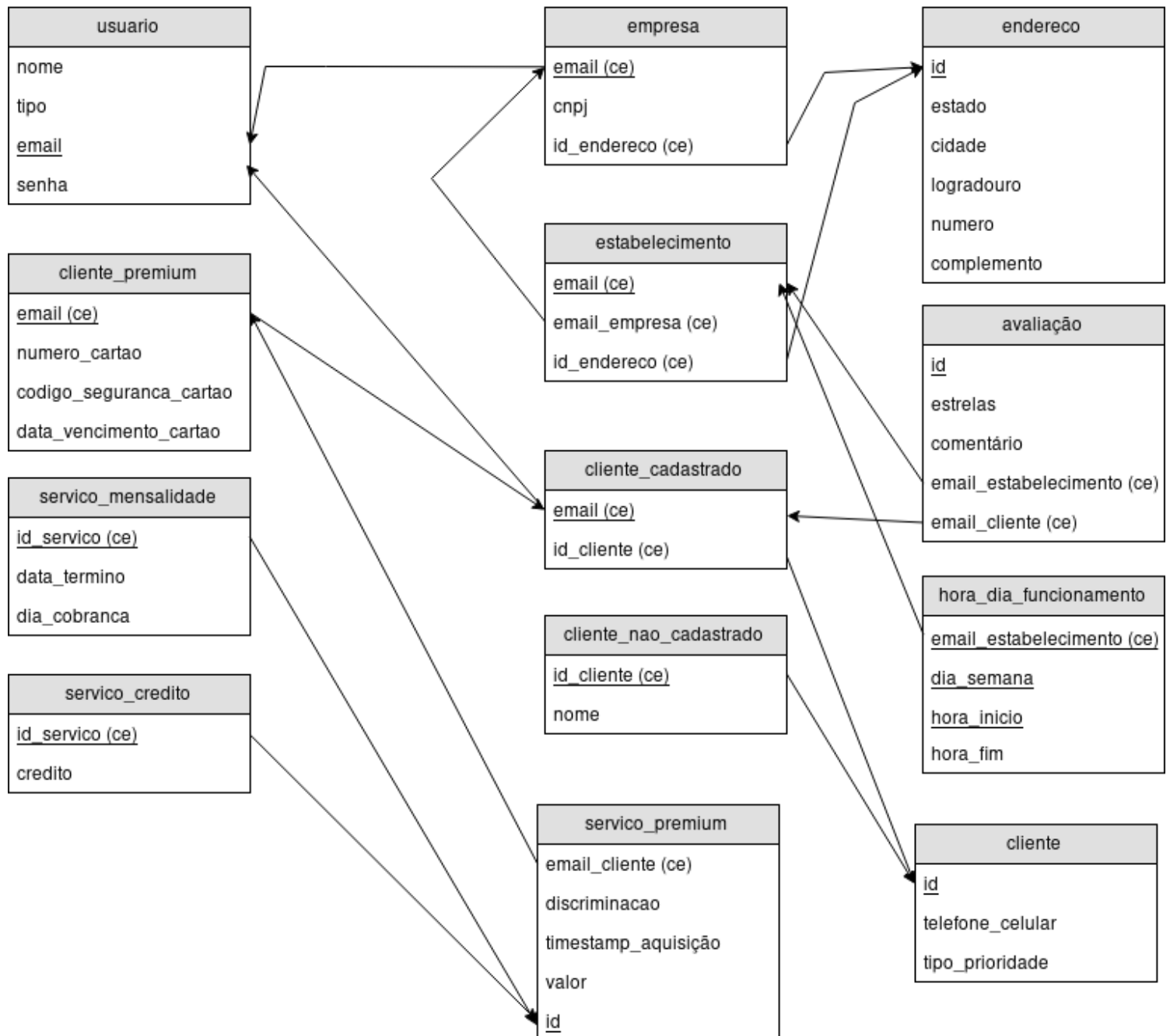
As filas cronológicas possuem um vetor que representa a fila propriamente dita (entradas) e outro para representar os que saíram da fila (concluidos). A posição na fila não é determinada pela posição no vetor e sim pelo atributo 'posicao'. Cada vez que um cliente entra na fila, ele é inserido no vetor de entradas, em ordem qualquer. Quando ele é atendido ou decide sair da fila, ele é removido do entradas e inserido no concluidos (também em ordem qualquer). (Isso é semelhante na fila agendada). Sempre que alguém entra ou sai da fila, o atributo 'tamanho' é atualizado.

As filas cronológicas são filas de prioridade, ordenadas por um score, calculado a partir dos parâmetros 'distancia', 'data_hora_entrada', 'preferencial?' e 'premium?'. Quando alguém entra, recebe como posição a última da fila.

As posições são recalculadas a cada K minutos (entre 1 e 3 minutos) a partir da hora de criação da fila. Neste intervalo de tempo, uma rotina é executada no servidor recalculando o score de todos na fila e reescrevendo suas posições. Isso é relativamente custoso mas apenas ocorrerá a cada K minutos, em momentos diferentes para cada fila e tanto o algoritmo quando os dados podem ser distribuídos em várias máquinas (execução paralela).

Para que o algoritmo funcione corretamente, é necessário que a posição GPS de cada cliente, seja enviada para os servidores com certa frequência, para que seu atributo 'distancia' (sua distância até o estabelecimento), esteja sempre atualizada. Especialmente na hora de recalculer os scores e as posicoes na fila.

Modelo Lógico (parte relacional)



Script para Criação de Modelo Físico (parte NoSQL)

```
DROP SCHEMA IF EXISTS inline CASCADE;
CREATE SCHEMA inline;

SET search_path TO inline;

CREATE TYPE prioridade_t AS ENUM ('-',
                                   'aposentado',
                                   'deficiente',
                                   'idoso',
                                   'gestante',
                                   'obeso',
                                   'criança de colo');

CREATE TYPE usuario_t AS ENUM ('empresa',
                                'estabelecimento',
                                'cliente cadastrado');

CREATE TYPE servico_t AS ENUM ('credito', 'mensalidade');

CREATE TABLE endereco (
  id      serial,
  estado   varchar(2) NOT NULL,
  cidade   varchar(50) NOT NULL,
  logradouro varchar(100) NOT NULL,
  numero    int      NOT NULL,
  complemento varchar(10),
  PRIMARY KEY (id),
  UNIQUE (estado, cidade, logradouro, numero, complemento)
);

CREATE TABLE usuario (
  nome  varchar(50) NOT NULL,
  tipo  usuario_t NOT NULL,
  email varchar(50),
  senha varchar(50) NOT NULL,
  PRIMARY KEY (email)
);

CREATE TABLE empresa (
  email      varchar(50),
  cnpj       varchar(50) NOT NULL,
  id_endereco int      NOT NULL,
  PRIMARY KEY (email),
  FOREIGN KEY (email) REFERENCES usuario ON DELETE CASCADE,
  FOREIGN KEY (id_endereco) REFERENCES endereco ON DELETE RESTRICT,
  UNIQUE (id_endereco)
);
```

```

CREATE TABLE estabelecimento (
    email          varchar(50),
    email_empresa varchar(50) NOT NULL,
    id_endereco   int          NOT NULL,
    posicao_gps    varchar(50) NOT NULL,
    PRIMARY KEY (email),
    FOREIGN KEY (email) REFERENCES usuario ON DELETE CASCADE,
    FOREIGN KEY (email_empresa) REFERENCES empresa ON DELETE CASCADE,
    FOREIGN KEY (id_endereco) REFERENCES endereco ON DELETE RESTRICT,
    UNIQUE (id_endereco)
);

```

```

CREATE TABLE cliente (
    id            serial,
    telefone_celular varchar(30),
    tipo_prioridade prioridade_t,
    PRIMARY KEY (id)
);

```

```

CREATE TABLE cliente_cadastrado (
    email          varchar(50),
    id_cliente int,
    PRIMARY KEY (email),
    FOREIGN KEY (email) REFERENCES usuario ON DELETE CASCADE,
    FOREIGN KEY (id_cliente) REFERENCES cliente ON DELETE RESTRICT
);

```

```

CREATE TABLE cliente_nao_cadastrado (
    id_cliente int,
    nome        varchar(50),
    PRIMARY KEY (id_cliente),
    FOREIGN KEY (id_cliente) REFERENCES cliente ON DELETE CASCADE
);

```

```

CREATE TABLE avaliacao (
    id            serial,
    estrelas       int          NOT NULL CHECK (estrelas >= 0 AND estrelas <= 5),
    comentario text,
    email_estabelecimento varchar(50) NOT NULL,
    email_cliente        varchar(50) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (email_estabelecimento) REFERENCES estabelecimento ON DELETE CASCADE,
    FOREIGN KEY (email_cliente) REFERENCES cliente_cadastrado ON DELETE CASCADE,
    UNIQUE (email_estabelecimento, email_cliente)
);

```

```

CREATE TABLE hora_dia_funcionamento (
    email_estabelecimento varchar(50),
    dia_semana             char(3),
    hora_inicio            time,
    hora_fim               time,
    FOREIGN KEY (email_estabelecimento) REFERENCES estabelecimento ON DELETE CASCADE,

```

```

PRIMARY KEY (email_estabelecimento, dia_semana, hora_inicio)
);

-- Tabelas não utilizadas na implementação

CREATE TABLE cliente_premium (
  email varchar(50),
  numero_cartao varchar(30) NOT NULL,
  codigo_seguranca_cartao int NOT NULL,
  data_vencimento_cartao date NOT NULL,
  PRIMARY KEY (email),
  FOREIGN KEY (email) REFERENCES cliente_cadastrado ON DELETE CASCADE
);

CREATE TABLE servico_premium (
  email_cliente varchar(50),
  discriminacao_servico_t NOT NULL,
  timestamp_aquisicao date NOT NULL,
  valor float NOT NULL,
  id serial,
  PRIMARY KEY (id),
  FOREIGN KEY (email_cliente) REFERENCES cliente_cadastrado ON DELETE RESTRICT
);

CREATE TABLE servico_credito (
  id_servico int,
  credito int NOT NULL,
  PRIMARY KEY (id_servico),
  FOREIGN KEY (id_servico) REFERENCES servico_premium ON DELETE CASCADE
);

CREATE TABLE servico_mensalidade (
  id_servico int,
  data_termino date NOT NULL,
  dia_cobranca date NOT NULL,
  PRIMARY KEY (id_servico),
  FOREIGN KEY (id_servico) REFERENCES servico_premium ON DELETE CASCADE
);

```

Tecnologias Utilizadas para Implementação

Para o servidor, utilizaremos o interpretador de javascript Node, com o framework Express para desenvolvimento web backend e a biblioteca React para gerar a GUI. Para a parte mobile, utilizaremos a tecnologia de PWA.

Os bancos de dados utilizados serão o PostgreSQL e o MongoDB. A interface deles com o Node será feita, respectivamente com o node-postgres e o mongoose.

A hospedagem dos bancos e do servidor web será feita no google cloud (versão de avaliação).