**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
Graduação em Ciência da Computação**

**Gerenciamento de Fila para Restaurantes**

**Participantes**

**Ana Flávia**

**Brenon Henrique**

**Carolina Lima**

**Diego Setúbal**

**Marco Braga**

**Rithie Natan**

|  |  |  |  |
| --- | --- | --- | --- |
| Histórico de Revisões | | | |
| Data | **Autor** | **Descrição** | **Versão** |
| 23/04/2020 | Rithie | [Descrever as principais alterações realizadas no documento, evidenciando as seções ou capítulos alterados] | 1.0 |
| 23/04/2020 | Rithie |  | 1.1 |
| 23/04/2020 | Rithie |  | 1.2 |
| 23/04/2020 | Rithie |  | 1.3 |

**SUMÁRIO**

[1 Apresentação 3](#_Toc509945066)

[1.1 Problema 3](#_Toc509945067)

[1.2 Objetivos do trabalho 3](#_Toc509945068)

[1.3 Definições e Abreviações 3](#_Toc509945069)

[2 Requisitos 3](#_Toc509945070)

[2.1 Restrições Arquiteturais 4](#_Toc509945073)

[2.2 Mecanismos Arquiteturais 4](#_Toc509945074)

[3 Modelagem e projeto arquitetural 5](#_Toc509945075)

[3.1 Visão Lógica 7](#_Toc509945077)

[3.2 Visão Física 7](#_Toc509945078)

[3.3 Modelo de dados (opcional) 8](#_Toc509945079)

4 [REFERÊNCIAS 9](#_Toc509945086)

[7 APÊNDICES 9](#_Toc509945087)

# Apresentação

Com a frequência de fluxo de pessoas em restaurantes, principalmente em casos de superlotação, observamos que estes estabelecimentos tem dificuldades com filas em reservas por falta de uma organização melhor. Por isso, a nossa motivação é criar um aplicativo web para que estes estabelecimentos tenham facilidades em controlar o fluxo de entrada e saída dos estabelecimentos.

## Problema

Após os relatos da insatisfação de clientes com filas de esperas e reservas em restaurantes, percebemos um gargalo na entrada destes estabelecimentos. O principal problema encontrado nestes gargalos, é culpa da falta de organização dos estabelecimentos em controlar as variáveis:

* Capacidade de ocupação do estabelecimento;
* Máximo de reservas por mesas;
* Estimativa de fluxo de pessoas;
* Média de tempo de ocupação para que uma mesa seja liberada;

## Objetivos do trabalho

O projeto arquitetural do sistema de gestão de filas para restaurantes consiste:

* Gestão de dados:
  + Gerentes de Restaurantes
  + Filas
  + Clientes dos restaurantes
* Arquitetura do sistema
  + Web application
* Padrões de projetos
  + Builder
  + Observer
  + Proxy
* Banco de Dados e Controle de Dados
  + Firebase
* Frameworks
  + React
* Linguagem
  + HTML5
  + CSS3
  + JavaScript
* Controle de Versões
  + Github
  + SEMVER (Versionamento Semântico) 2.0

Os itens apresentados acima, representam a descrição de forma resumida do projeto. Valido ressaltar que a principal ferramenta para acompanhar e organizar dados será o Firebase. É a principal ferramenta e o coração do projeto, é através dela que iremos construir a nossa conexão com o servidor, além de distribuir as ‘views’ de dados para os usuários.

# Requisitos da Arquitetura

## Restrições Arquiteturais

Restrições a arquitetura:

* A aplicação web será desenvolvida em React
* Padrão RESTful será utilizado para a comunicação da API
* Utilização do Firebase, no qual é um serviço de banco de dados NoSQL
* Controle de versões definidas, seguindo as regras do SEMVER 2.0

## Mecanismos Arquiteturais

|  |  |  |
| --- | --- | --- |
| **Análise** | **Design** | **Implementação** |
| Persistência | NoSQL | FireBase |
| Front-end | Camada responsável em disponibilizar meio de interação do usuário e o sistema | React, linguagem HTML5, CSS3 e JavaScript |
| Back-end | Camada responsável por implementar  as regras de negócio do sistema | JavaScript |
| Integração | Não utilizado |  |
| Log do sistema | Implementação dos recursos de log do componente de persistência. | Firebase |
| Teste de Software | Teste unitário, teste de interface, teste de caixa-preta | Selenium |
| Deploy | Configuração da IDE de deploy. | Visual Studio Code |
| Fluxo de trabalho | Suporte à circulação de documentos e outros itens de trabalho. | Trello, Github |

# Modelagem e projeto arquitetural

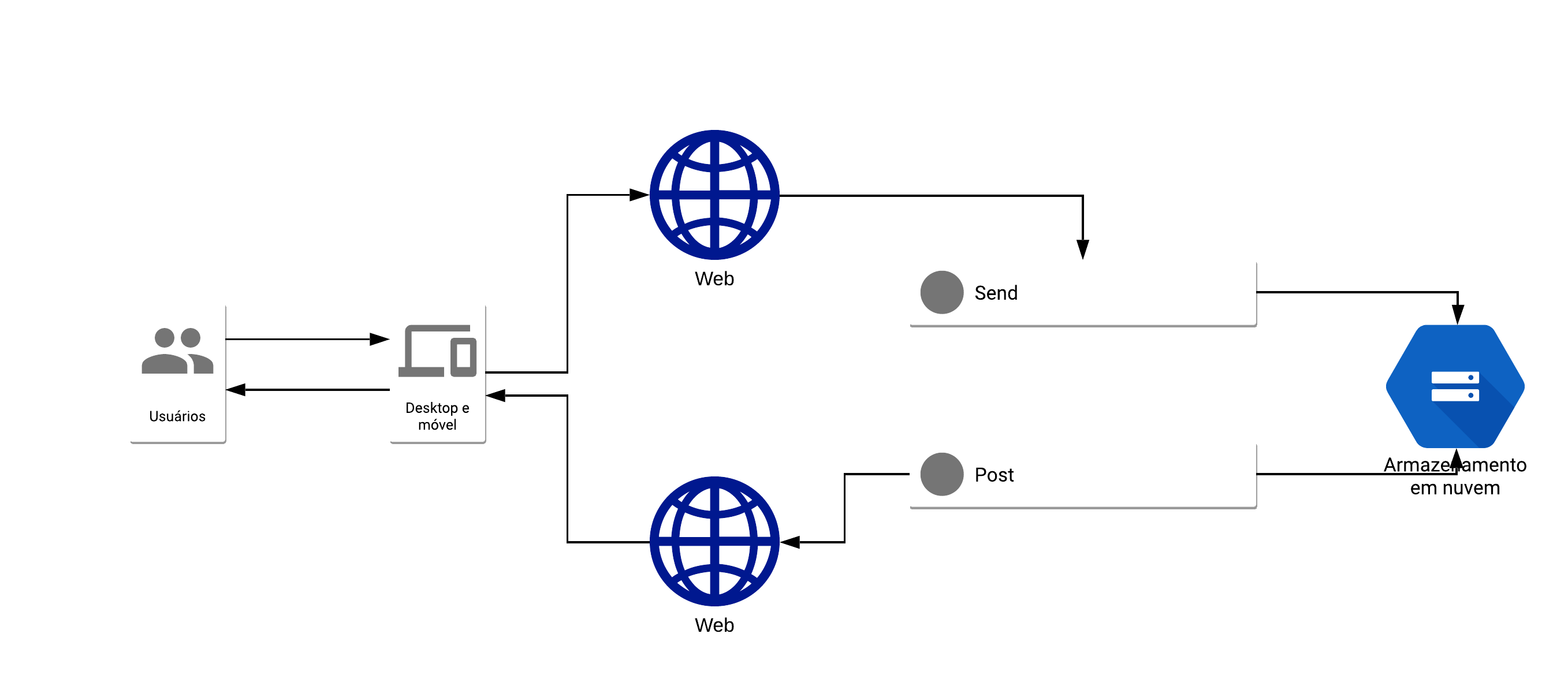


Figura 1Visão Geral da Solução

### Detalhamento dos casos de uso

**Caso de Uso:**

CRUD do Funcionário

**Sumário:**

Efetua o cadastro do funcionário.

**Atores:**

* Funcionário.

**Pré-condições:**

O funcionário deve ser o gerente.

**Fluxo de eventos:**

**Fluxo básico:**

* A empresa inicia o caso de uso.
* O sistema exibe a tela para inserir os dados do funcionário e suas permissões.
* O sistema valida os dados e salva o cadastro.
* O sistema finaliza o caso de uso.

**Fluxos alternativos:**

Não se aplica.

**Pós-condições:**

**Não se aplica.**

**-------------------------------------------------------------------//----------------------------------------------------------------**

**Caso de Uso:**

CRUD do Cliente

**Sumário:**

Efetua o cadastro do cliente.

**Atores:**

* Cliente.

**Pré-condições:**

Não se aplica

**Fluxo de eventos:**

**Fluxo básico:**

* O cliente inicia o caso de uso.
* O sistema exibe a tela para inserir os dados do cliente.
* O sistema valida os dados e salva o cadastro.
* O sistema finaliza o caso de uso.

**Fluxos alternativos:**

Não se aplica.

**Pós-condições:**

**Não se aplica.**

## Visão Lógica

### Modelo de Classes

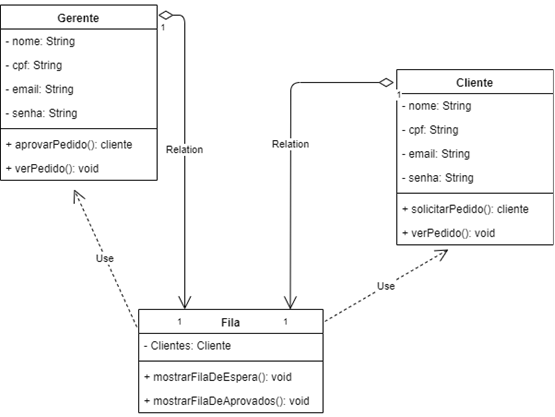


Figura 3Diagrama de classes

### Modelo de componentes

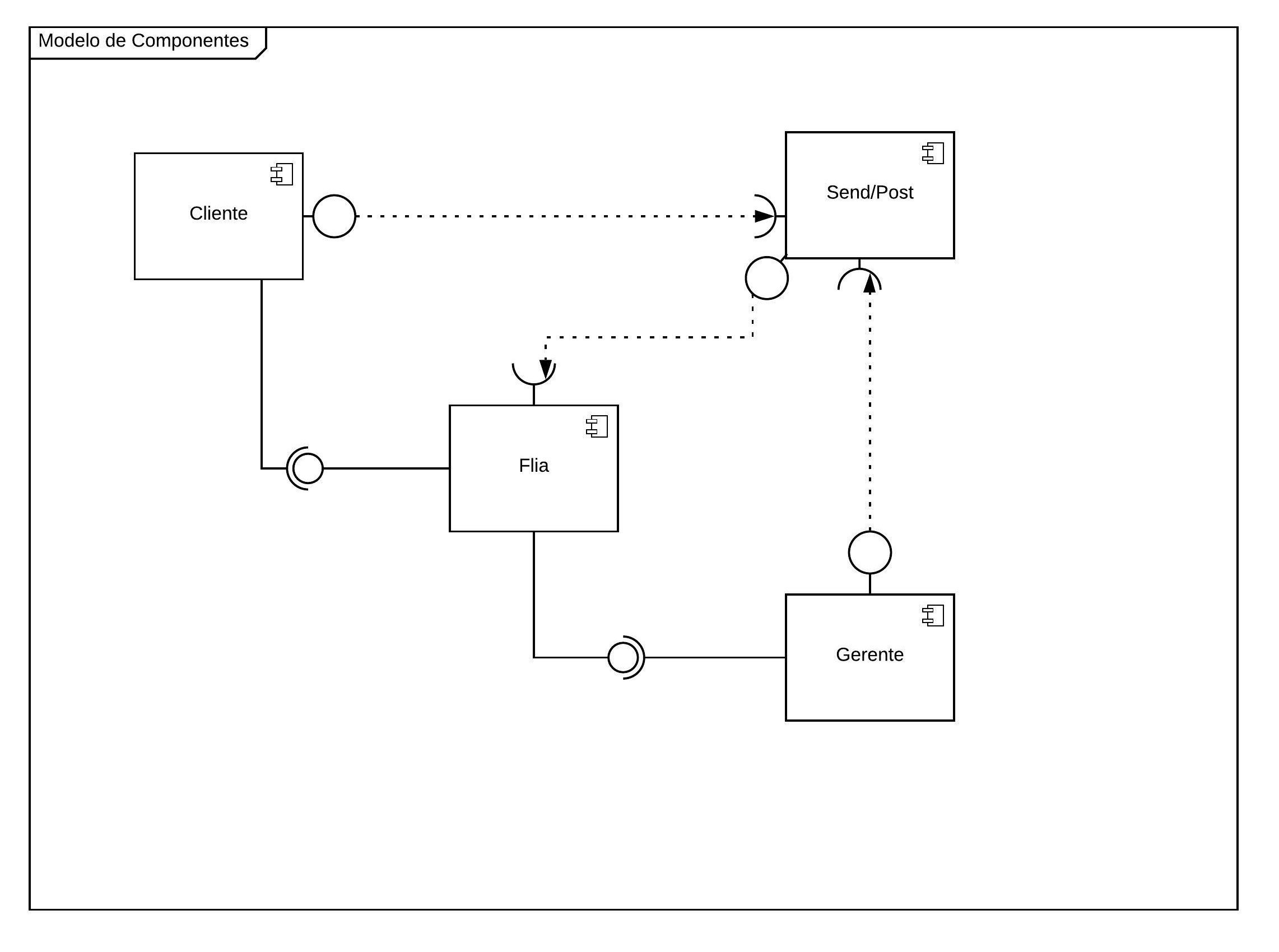


Figura 3Diagrama de Componentes

## Visão Física

### Modelo de implantação – Diagrama de Nodos

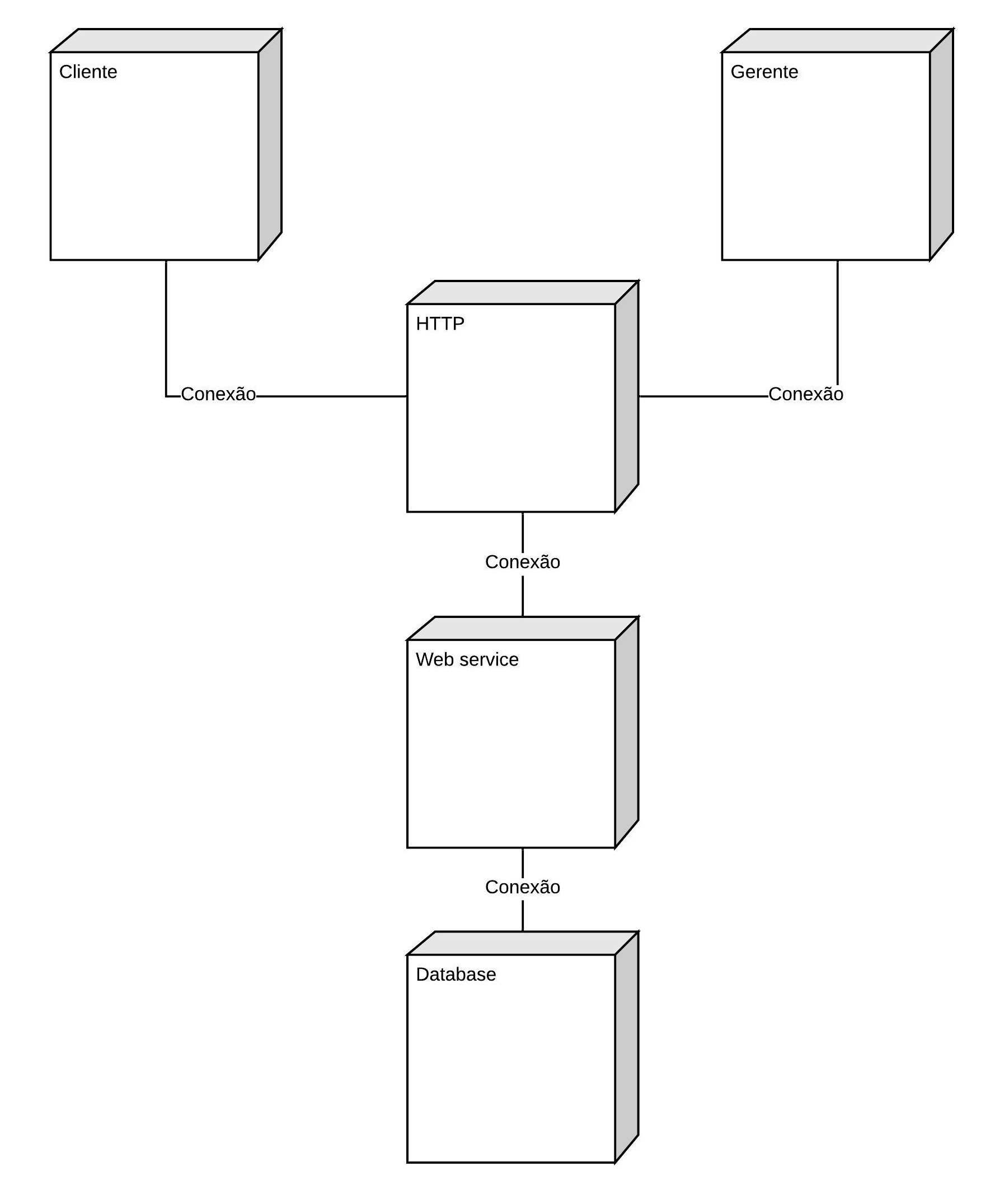


Figura 3Diagrama de Implantação

Conforme diagrama apresentado na acima, as entidades participantes da solução são:

* **Cliente** – Clientes dos restaurantes cadastrados na plataforma
* **Gerente** – Gerentes dos restaurantes cadastrados na plataforma
* **HTTP** – Website da plataforma, onde são fornecidas as interfaces para o cliente e para o gerente.
* **Web service** – Serviço de requisição web para interfacear a conexão de APIs e dados. Além de atualizar as informações para as interfaces de cliente/gerente.
* **Database** – Armazena os dados dos clientes e gerentes.

### Modelo de implantação – Diagrama de Componentes

