Versão 1.0

Casos de Uso presentes no documento:

* Registro de Restaurante
* Registro de Cliente
* Login
* Visualização de Informações Sobre a Fila em que o Cliente Está
* Entrar na Fila de um Restaurante Selecionado

**Caso de Uso #1:**

Entrar na Fila de um Restaurante Selecionado

**Ator Primário:** Cliente (Usuário)

**Escopo:** Informações sobre cada restaurante e suas filas (Banco de Dados) / Visualização de Frame Principal (para atualizar e possibilitar a interação do cliente com a fila recém entrada) / Gerenciamento de Fila do Restaurante

**Level:** Objetivo de Usuário

**Stakeholders e Interesses:**

* Cliente (Usuário): Passar a obter o estado de “Dentro de Fila” associado a uma fila exclusiva e passar a poder visualizar e interagir com informações desta.
* Restaurante (Usuário): Receber a informação de que um novo cliente entrou em sua fila, de modo a integrá-lo ao gerenciamento da fila.

**Pré-condições:** Estar Logado no Sistema, ter o estado de “Dentro de Fila” como falso.

**Garantia Mínima:** Salvar log sobre erros ocorridos, notificar o usuário, manter as estruturas de dados e informações da mesma forma que no início do caso de uso.

**Garantia em Caso de Sucesso:** O usuário passa a obter o estado de “Dentro de Fila” e passa a estar associado a uma fila exclusiva. O usuário é integrado ao gerenciamento de fila na parte do restaurante da aplicação.

**Cenário Principal de Sucesso:**

1. Cliente (Usuário) clica em ‘Entrar Fila’, seja na página do restaurante selecionado ou na página de busca de restaurantes.
2. Verificação se o cliente está elegível para entrar em uma fila. (Verificações: o cliente está logado no sistema e o cliente não possui o estado “Dentro de Fila”)
3. Solicitação de ‘Entrada no Sistema de Filas’ no ‘Gerenciamento de Filas’ do Restaurante selecionado.
4. Em caso de sucesso, alteração do estado do estado “Dentro de Fila” do cliente para verdadeiro, alteração do ID do restaurante em que o cliente está na fila para o ID do restaurante selecionado e inserção do cliente no ‘Gerenciamento de Fila’ do restaurante selecionado. (Operação Atômica)
5. Envio de página com mensagem de sucesso e informações sobre a fila recém entrada.
6. Retorno à página em que o Cliente começou o Caso de Uso.

**Cenário Alternativo:**

2a. Caso o usuário não esteja logado, informá-lo que é necessária a realização do login para ingressar numa fila.

2b. Caso o usuário já esteja em outra filha, informá-lo que só é possível estar em uma fila.

3a. Caso a fila fique lotada no momento em que o cliente tenta ingressar na fila, notificá-lo do ocorrido e retornar à página do restaurante.

4a. Caso ocorra uma falha na comunicação com o banco de dados, informar o usuário e salvar log de erro.   
**Caso de Uso #2**:

Registro do Cliente

**Ator Primário**: Cliente

**Escopo**: Formulário de cadastro com campos para que o usuário digite suas informações / Acesso ao banco de dados para alocação dos dados / Envio de e-mail para o endereço recém-cadastrado através do formulário

**Objetivo de usuário**: Registrar-se na aplicação.

**Stakeholder e Interesses**:

* Cliente: Possuir um cadastro na aplicação, tendo suas informações registradas no banco de dados

**Pré-condições**: Estar na página do smartQueue

**Garantia mínima**: Salvar log sobre erros ocorridos, notificar o usuário e manter as estruturas de dados inalteradas.

**Garantia em Caso de Sucesso**: Cliente consegue se cadastrar, receber um e-mail de confirmação e possuir seus dados alocados no banco de dados.

**Cenário Principal de Sucesso:**

1. Usuário clica no botão “registrar”
2. Aplicação retorna um HTML com o formulário de cadastro
3. Usuário preenche os campos do formulário de cadastro
4. Usuário clica no botão “submit” para enviar o formulário de cadastro ao servidor
5. Inicializa um processo para salvar os dados – já validados - dos campos do formulário no banco de dados
6. Aplicação envia uma mensagem de confirmação de cadastro via e-mail para o endereço de e-mail recém-cadastrado
7. Exibe uma página HTML informando que um e-mail de confirmação de cadastro foi enviado

**Cenário Alternativo:**

4a. Caso o usuário preencha campos com conteúdo inválido (e.g. e-mail não existente), salvar os campos preenchidos corretamente e notificar o usuário, através de uma marcação, qual campo foi indevidamente preenchido e qual o motivo do conteúdo não ter sido aceito.

5a. Caso o banco de dados esteja cheio, notifica o usuário.

5b. Caso ocorra uma falha ao acessar o banco de dados, notifica o usuário e faz um log para a futura correção.

**Caso de Uso #3:**

Registro do Restaurante

**Ator Primário**: Restaurante

**Escopo**: Formulário de cadastro com campos para que o usuário digite suas informações / Acesso ao banco de dados para alocação dos dados / Envio de e-mail para o endereço recém-cadastrado através do formulário

**Objetivo de usuário**: Registrar-se na aplicação.

**Stakeholders e Interesses**:

* Restaurante: Possuir um cadastro na aplicação, tendo suas informações registradas no banco de dados

**Pré-condições**: Estar na página do smartQueue

**Garantia mínima**: Salvar log sobre erros ocorridos, notificar o usuário e manter as estruturas de dados inalteradas.

**Garantia em Caso de Sucesso**: Restaurante consegue se cadastrar, receber um e-mail de confirmação e possuir seus dados alocados no banco de dados.

**Cenário Principal de Sucesso:**

1. Usuário clica no botão “registrar”
2. Aplicação retorna um HTML com o formulário de cadastro
3. Usuário preenche os campos do formulário de cadastro
4. Usuário clica no botão “submit” para enviar o formulário de cadastro ao servidor
5. Inicializa um processo para salvar os dados – já validados - dos campos do formulário no banco de dados
6. Aplicação envia uma mensagem de confirmação de cadastro via e-mail para o endereço de e-mail recém-cadastrado
7. Exibe uma página HTML informando que um e-mail de confirmação de cadastro foi enviado

**Cenário Alternativo:**

4a. Caso o usuário preencha campos com conteúdo inválido (e.g. e-mail não existente), salvar os campos preenchidos corretamente e notificar o usuário, através de uma marcação, qual campo foi indevidamente preenchido e qual o motivo do conteúdo não ter sido aceito.

5a. Caso o banco de dados esteja cheio, notifica o usuário.

5b. Caso ocorra uma falha ao acessar o banco de dados, notifica o usuário e faz um log para a futura correção.

**Caso de Uso #4:**

Visualização do Frame Principal

**Ator Primário:** Cliente (Usuário)

**Escopo:** Acesso ao banco de dados / Visualização da fila em que o usuário está

**Level:** Objetivo de usuário

**Stakeholders e Interesses:**

* Cliente: ter controle sobre a fila virtual em que entrou.

**Pré-condição:** Estar logado no sistema.

**Garantia Mínima:** Saber se está ou não na fila.

**Garantia em caso de sucesso:** Visualização de sua posição na fila / quantos estão na sua frente.

**Cenário Principal de Sucesso:**

1. Usuário abre página do restaurante onde está localizada a fila
2. Aplicação acessa o banco de dados para checar se o usuário está em uma fila.
3. Caso o usuário esteja em uma fila, banco de dados retorna informações sobre a fila em que o usuário está.
4. Aplicação exibe página HTML com a posição do usuário na fila e quantos estão na sua frente.

**Cenário Alternativo:**

2.a Falha no acesso ao banco de dados

Aplicação exibe mensagem de erro e salva o log.

3.a Falha na transmissão de informações do banco de dados para a aplicação

Aplicação exibe mensagem de erro e salva o log.

**Caso de Uso #5:**

Login

**Ator Primário:** Cliente e Restaurante (Usuário)

**Escopo:** Interface de usuário

**Level:** Objetivo de usuário

**Stakeholders e Interesses:**

* Restaurante: Pode editar os dados de sua conta e manusear sua fila virtual.
* Cliente: Pode editar os dados de sua conta e entrar em filas virtuais.

**Pré-Condição:** Ter uma conta registrada e validada.

**Garantia mínima:** Exibição de mensagem de erro notificando o porquê da falha ao realizar o log-in.

**Garantia em caso de sucesso:** Usuário pode desfrutar dos privilégios do site relativos à sua conta.

**Cenário Principal de Sucesso**

1. Usuário clica no botão "login" após preencher os campos.
2. Aplicação recebe as informações referentes aos campos preenchidos no formulário e compara com dados do banco de dados.
3. Aplicação confirma e valida o ‘username’ e a ‘senha’ digitados.
4. Aplicação exibe uma página HTML com mensagem de boas-vindas, tal como o menu para modificações de dados pessoais.

**Cenário Alternativo**

1.a Usuário e/ou senha não inseridos.

* Aplicação recusa a ação. O módulo de conexão com o banco de dados só é executado caso haja conteúdo nos campos.

2.a Falha na conexão com o banco de dados

* Exibe-se mensagem de erro explicando o ocorrido e salva log de erro.

3.a Usuário e/ou senha inválidos

* Exibe-se mensagem de erro pedindo para o usuário preencher os campos novamente.