

## ***1. Requisitos Funcionais (RF)***

### **Módulo: Registro de Entrada**

**RF001 - Check-in Rápido:** O sistema deve permitir o registro rápido da entrada de um veículo.

*Campos obrigatórios:* Placa (validação de formato), Modelo, Data/Hora de Entrada (gerada automaticamente pelo sistema).

*Campos opcionais:* Cor, Observações.

**RF002 - Identificação de Veículo:** O sistema deve verificar se a placa informada já está no pátio (evitar duplicidade de veículo ativo).

### **Módulo: Cadastro de Tarifas**

**RF003 - Gerenciar Tarifas:** O sistema deve permitir o CRUD (Criar, Ler, Atualizar, Deletar) de tarifas.

*Campos:* Nome (ex: "Hora Comum", "Diária"), Tipo (Hora, Fração, Diária), Valor, Vigência (data inicial/final).

*Exemplo:* "Hora" = R\$ 5,00; "Fração de 15min" = R\$ 1,50; "Diária" = R\$ 30,00.

**RF004 - Tarifa Ativa:** O sistema deve utilizar apenas a tarifa vigente para o período do cálculo (considerar data da saída).

### **Módulo: Registro de Saída e Cobrança**

**RF005 - Checkout:** O sistema deve localizar um veículo pela placa para realizar a saída.

**RF006 - Cálculo Automático:** Ao registrar a saída, o sistema deve:

Calcular o tempo de permanência (diferença entre Data/Hora Saída e Data/Hora Entrada).

Aplicar a regra de tarifação vigente:

*Por hora:* Arredondar para cima (ex: 1h10min = 2 horas).

*Por fração:* Calcular número de frações.

*Diária:* Atingiu 24h? Cobrar diária(s).

Exibir o valor calculado antes de finalizar.

**RF007 - Registro de Pagamento:** O sistema deve registrar a forma de pagamento (Dinheiro, Cartão, Pix) e finalizar a saída, movendo o veículo do "pátio" para o "histórico".

### **Módulo: Relatórios e Controle**

**RF008 - Veículos no Pátio:** O sistema deve listar todos os veículos que estão atualmente no estacionamento, com placa, modelo e hora de entrada.

**RF009 - Relatório de Faturamento:** O sistema deve gerar um relatório de caixa com total faturado em um período selecionado (dia, semana, mês), podendo filtrar por forma de pagamento.

**RF010 - Histórico de Movimentações:** O sistema deve permitir consultar veículos que já saíram por período.

## ***2. Requisitos Não Funcionais (RNF)***

### **RNF001 - Usabilidade (Operação)**

**Agilidade:** As telas de entrada e saída devem priorizar a agilidade, com o mínimo de cliques possível e foco no campo "Placa" (pode incluir leitura de QR code ou busca automática ao digitar).

**Validação em Tempo Real:** O campo de placa deve validar o formato (padrão Mercosul ou antigo) instantaneamente.

### **RNF002 - Performance**

**Cálculo de Saída:** O tempo de resposta do cálculo da tarifa deve ser imediato (< 1 segundo).

**Relatórios:** Relatórios de longo período devem carregar em até 3 segundos.

### **RNF003 - Disponibilidade**

**Operação Off-line?** (Opcional) Se o sistema for para um estacionamento real, considerar se ele precisa funcionar sem internet e sincronizar depois.

## **RNF004 - Precisão**

**Arredondamento:** O sistema deve ter regras claras e consistentes de arredondamento de tempo (para cima, para baixo, por fração) para evitar disputas com clientes.

**Integridade Referencial:** Um veículo não pode ser "deletado" depois que sai; deve ser movido para histórico (Soft Delete ou tabela separada).

## **RNF005 - Tecnologia**

**Banco de Dados:** Relacional (SQL) para garantir consistência nos cálculos financeiros.

**API:** Recomenda-se REST para comunicação entre frontend e backend.

**Registro de Log:** Todas as operações de entrada e saída devem ser logadas para auditoria.

## **Sprint 1: Base e Configurações**

### **Equipe Modelagem BD e Tarifas:**

Criar modelo relacional: veículos (ou movimentações), tarifas, histórico de pagamentos.

Definir como tratar veículos ativos vs. histórico (ex: flag ativo ou tabelas separadas).

Configurar ambiente de banco de dados.

### **Equipe CRUD de Tarifas:**

Desenvolver API para CRUD de tarifas.

Desenvolver tela administrativa para gerenciar tarifas (listar, criar, editar, excluir).

Implementar lógica de "tarifa vigente" (datas de validade).

## **Sprint 2: Operação Principal (Entrada e Saída)**

### **Equipe Tela de Entrada (Check-in rápido):**

Desenvolver interface focada na digitação da placa (sugestão: ao digitar placa, campo modelo aparece em foco).

Salvar registro no banco com data/hora atual.

Validar se veículo já não está no pátio.

### **Equipe Tela de Saída (Checkout e Cálculo):**

Criar tela de busca por placa.

Exibir dados da entrada e tempo decorrido em tempo real.

Implementar **lógica de negócio do cálculo** (função que recebe entrada, saída e tarifa, retorna valor).

Integrar com a tabela de tarifas vigentes.

### **Sprint 3: Relatórios e Finalização**

#### **Equipe de Relatórios:**

**Backend:** Criar endpoints para:

Listar veículos no pátio (consulta simples).

Somar faturamento por período (GROUP BY data, SUM(valor)).

#### **Frontend:**

Tela "Veículos no Pátio" com atualização automática (polling a cada 30s ou websocket).

Tela "Relatório de Faturamento" com filtros de data e exibição em tabela + gráfico simples.

#### **Refinamento Geral (Toda equipe):**

Testar fluxo completo: Cadastrar tarifa -> Entrada -> Saída -> Verificar no relatório.

Implementar validações extras (ex: não permitir saída sem tarifa cadastrada).

- Ajustes de UI/UX.