

# Motor de Categorização e Cache de Transações

## 1. Objetivo

Definir como funciona o motor de categorização de transações do app, priorizando:

- **Boa experiência mobile-first**
- **Redução de custo com IA**
- **Aprendizado automático e progressivo**
- **Consistência e previsibilidade do sistema**

O sistema deve categorizar transações automaticamente, aprender com o uso e evitar qualquer fluxo que exija confirmação manual em massa pelo usuário.

---

## 2. Princípios do Sistema

1. O usuário **nunca é obrigado a confirmar categorias**
  2. O sistema aprende **apenas quando o usuário corrige algo**
  3. IA é **último recurso**
  4. O cache é a principal fonte de categorização
  5. Regras do usuário sempre têm prioridade sobre regras globais
- 

## 3. Fluxo Geral do Usuário

1. Usuário importa um arquivo (CSV ou OFX)
  2. Sistema detecta o banco e processa o arquivo
  3. Todas as transações são categorizadas automaticamente
  4. Dashboard é exibido já pronto
  5. Usuário pode (opcionalmente) alterar alguma categoria
  6. O sistema aprende silenciosamente a partir dessas alterações
- 

## 4. Normalização de Transações

Antes de qualquer categorização, toda transação passa por normalização:

- Uppercase / lowercase padronizado

- Remoção de números, datas e cidades
- Remoção de prefixos comuns (ex: PAG\*, COMPRA\*, PIX\*)
- Extração do merchant principal

## Exemplo

"PAG\*IFOOD 12/01 SAO PAULO"  
→ "ifood"

Essa chave normalizada é usada em todo o sistema.

---

## 5. Estrutura de Cache de Categorização

### 5.1 Estrutura Base (Category Rule)

Cada regra de categorização possui:

- `normalizedKey`
- `category`
- `source: user | global | bank | ia`
- `userId` (opcional)
- `confidence`
- `timesUsed`
- `lastUsedAt`

### 5.2 Tipos de Cache

#### Cache do Usuário

- Possui `userId`
- Criado apenas quando o usuário altera uma categoria
- Prioridade máxima

#### Cache Global

- `userId = null`
  - Criado a partir de IA, banco ou regras administrativas
  - Usado quando não existe regra do usuário
- 

## 6. Ordem de Prioridade de Categorização

Para cada transação, o sistema resolve a categoria seguindo a ordem abaixo:

1. Regra do usuário (userId)
2. Regra global
3. Categoria fornecida pelo banco
4. Regras locais simples (regex / dicionário)
5. IA
6. Sem categoria

A primeira regra encontrada encerra o processo.

---

## 7. Uso da Categoria do Banco

Alguns arquivos (ex: CSV do C6) trazem a categoria da transação.

### Regras:

- Categoria do banco é tratada como **sugestão**
  - Possui confidence média
  - Pode ser usada sem IA
  - Se o usuário não corrigir após múltiplos usos, pode virar regra global
- 

## 8. Uso da IA

A IA é utilizada **somente quando nenhuma outra fonte resolve**.

### Diretrizes:

- Enviar apenas descrição normalizada e valor
  - Receber categoria + confidence
  - Não solicitar confirmação do usuário
  - Salvar no cache global apenas se:
    - confidence  $\geq$  limite definido
    - ou houver repetição sem correção
- 

## 9. Aprendizado Baseado no Usuário

### Correção Manual

Quando o usuário altera a categoria de uma transação:

- Uma regra é criada ou atualizada no cache do usuário
- Essa regra sobrescreve qualquer regra global futura
- Não afeta outros usuários

## Exemplo

```
"ifood" → "Lazer"  
userId = 123
```

---

## 10. Comportamento em Reimportações

Quando uma transação semelhante aparecer novamente:

- O sistema reutiliza a regra do usuário (se existir)
  - Caso contrário, usa regra global
  - IA não é chamada novamente
- 

## 11. Promoção de Regras

### Usuário → Global

- Nunca acontece automaticamente no MVP
- Pode ser feito manualmente no futuro com base em padrões

### Banco / IA → Global

- Permitido quando há repetição e ausência de correção
- 

## 12. Métricas Importantes

O sistema deve registrar:

- % de transações categorizadas sem IA
- % de uso do cache do usuário
- % de uso do cache global
- % de uso de categoria do banco
- % de uso de IA

Essas métricas guiam otimizações futuras e modelo premium.

---

## **13. Resumo Final**

- O usuário nunca é interrompido para confirmar
- Correção manual = aprendizado explícito
- IA é fallback
- Cache é o coração do sistema
- Prioridade do usuário é absoluta

Este design garante boa UX mobile, baixo custo operacional e aprendizado contínuo do sistema.