

Definição

Proposta Refinada — Sistema de Gestão de Horas Extras e Benefícios

1. Visão Geral

O sistema tem como objetivo **registrar, processar e converter horas extras dos funcionários em saldo de benefício interno**, que pode ser utilizado no restaurante da empresa.

A hora extra é o elemento central do sistema, e todo o fluxo parte dela.

2. Conceitos Fundamentais

Hora Extra

Registro das horas trabalhadas além da jornada regular de um funcionário em um determinado dia.

- O registro é feito **por dia**
 - Considera o **total de horas extras realizadas no dia**
 - Serve como base para cálculo de benefícios
-

Regra de Conversão

Define como as horas extras são convertidas em benefício financeiro interno.

Regra vigente:

- Até **2 horas extras no dia** → conversão equivalente a **50% do valor da hora normal**
- Se a jornada diária de horas extras **ultrapassar 2 horas** → conversão equivalente a **100% do valor da hora normal**

O cálculo é feito com base no **valor da hora normal do funcionário**, não sobre salário mensal ou diária.

A regra é aplicada **por dia**, considerando o total de horas extras realizadas naquele dia.

Saldo de Benefício

Resultado do processamento das horas extras.

- É um **crédito interno**
 - Fica associado ao funcionário
 - Pode ser consultado pelo funcionário
 - Não expõe informações salariais
-

Consumo no Restaurante

O restaurante interno utiliza exclusivamente o **saldo de benefício**.

- Apenas debita valores do saldo
 - Não tem acesso a:
 - salário
 - horas extras
 - regras de conversão
-

Funcionário

O funcionário tem acesso apenas à **consulta de informações**:

- Histórico de horas extras
- Saldo de benefício disponível

Não pode editar registros nem alterar regras.

Gestor

O gestor tem acesso a relatórios de acompanhamento, como:

- Quantidade de funcionários realizando hora extra
 - Volume de horas extras por período
 - Distribuição por setor
 - Impacto financeiro estimado
-

3. Fluxo Geral do Sistema

1. O funcionário realiza horas extras
2. As horas extras são registradas no sistema

3. Ao final do dia, o sistema consolida o total de horas extras
 4. A regra de conversão é aplicada
 5. O saldo de benefício é atualizado
 6. O funcionário pode consultar seu saldo
 7. O restaurante pode debitar valores do saldo
 8. O gestor acompanha os relatórios
-

4. Princípios Importantes do Sistema

- O cálculo é feito **com base no total diário de horas extras**
 - A conversão é baseada no **valor da hora normal do funcionário**
 - As regras de conversão são **centralizadas e controladas**
 - O restaurante não tem acesso a dados sensíveis
 - O sistema mantém histórico e rastreabilidade
-

5. Benefícios para a Empresa

- Controle claro e auditável das horas extras
 - Conversão automática e padronizada em benefício
 - Redução de conflitos e erros manuais
 - Transparência para funcionários e gestores
 - Base sólida para futuras expansões (novas regras, novos benefícios)
-

6. Escopo Inicial (MVP)

Inclui:

- Cadastro de funcionários
- Registro de horas extras
- Processamento diário
- Controle de saldo de benefício
- Consumo no restaurante
- Relatórios básicos gerenciais

Não inclui (neste momento):

- Pagamento em folha
 - Múltiplos tipos de benefícios
 - Regras variáveis por sindicato
-

7. Considerações Finais

O sistema foi desenhado para ser:

- Simples para o usuário final
 - Seguro em relação a dados sensíveis
 - Flexível para futuras mudanças de regras
 - Escalável conforme o crescimento da empresa
-

Se quiser, no próximo passo eu posso:

- Transformar isso em **documento formal de requisitos**
- Criar um **diagrama ER**
- Escrever um **escopo técnico para orçamento**

Esse material já está **pronto para envio ao cliente**.

Telas

TELAS — FUNCIONÁRIO

O funcionário **apenas consulta informações.**

1 Tela: Dashboard do Funcionário

 **Objetivo:** visão rápida da situação atual.

 Exibe:

- Saldo de benefício disponível
- Total de horas extras no mês
- Última hora extra registrada
- Último consumo no restaurante

 Permissão:

- Apenas visualização

 Ações:

- Ir para “Minhas Horas Extras”
 - Ir para “Histórico de Consumo”
-

2 Tela: Minhas Horas Extras

 **Objetivo:** consultar histórico detalhado.

 Exibe:

- Lista por data
- Total de horas no dia

- Regra aplicada (50% ou 100%)
- Valor convertido naquele dia
- Status (processada / pendente / ajustada)

 Permissão:

- Apenas visualização

 Ações:

- Filtrar por período
 - Ver detalhes de um dia específico
-

3 Tela: Histórico de Consumo

 **Objetivo:** acompanhar uso do saldo.

 Exibe:

- Data
- Valor debitado
- Tipo de operação (consumo / estorno)
- Saldo após operação

 Permissão:

- Apenas visualização

 Ações:

- Filtrar por período
-



TELAS — GESTOR

O gestor tem visão operacional e estratégica.

1 Tela: Dashboard Gerencial

⌚ Objetivo: visão geral da empresa/setor.

📌 Exibe:

- Total de horas extras no mês
- Total de saldo gerado
- Funcionários com mais de 2h frequentes
- Ranking por setor

🔒 Permissão:

- Visualização ampla (limitada ao seu setor, se aplicável)

⚙️ Ações:

- Filtrar por período
 - Exportar relatório
-

2 Tela: Relatório de Horas Extras

⌚ Objetivo: análise detalhada.

📌 Exibe:

- Funcionário
- Setor
- Data

- Horas no dia
- Regra aplicada
- Valor convertido

 Permissão:

- Visualização
- Exportação

 Ações:

- Filtrar por:
 - período
 - setor
 - funcionário
 - Exportar CSV/PDF
-

3 Tela: Aprovação / Ajuste de Horas (se aplicável)

 **Objetivo:** validar ou corrigir registros.

 Exibe:

- Horas registradas pendentes
- Histórico de ajustes

 Permissão:

- Aprovar
- Solicitar correção

- Ajustar horas



- Aprovar
 - Rejeitar
 - Editar (com justificativa)
-

4 Tela: Funcionários do Setor

⌚ Objetivo: visão individual consolidada.



- Total de horas no mês
- Total de saldo gerado
- Frequência de dias acima de 2h



- Visualização
-

TELAS — RESTAURANTE

O restaurante é **operacional e simples**.

1 Tela: Consulta de Saldo

⌚ Objetivo: verificar saldo antes de debitar.



- Nome do funcionário
- Saldo disponível

 Permissão:

- Visualização restrita ao saldo

 Ações:

- Confirmar débito
-

2 Tela: Realizar Débito

 Objetivo: registrar consumo.

 Exibe:

- Campo para valor
- Confirmação de saldo suficiente
- Botão “Confirmar”

 Permissão:

- Criar movimentação de débito

 Ações:

- Debitar valor
 - Cancelar operação
-

3 Tela: Histórico de Movimentações (Restaurante)

 Objetivo: controle interno do restaurante.



Exibe:

- Lista de débitos realizados
- Data
- Funcionário
- Valor
- Estornos (se houver)



Permissão:

- Visualização
 - Estorno (se autorizado)
-



TELAS ADMINISTRATIVAS (Sistema)

Essas telas são para controle interno da empresa.

1 Cadastro de Funcionários



Exibe:

- Nome
- Matrícula
- Setor
- Valor da hora normal
- Status



Ações:

- Criar
 - Editar
 - Desativar
-

2 Configuração de Regra de Conversão



- Exibe:
- Limite de horas (2h)
 - Percentual até limite (50%)
 - Percentual acima limite (100%)



- Ações:
- Editar regra
 - Versionar regra
-

3 Processamento Diário



- Exibe:
- Dias pendentes de processamento
 - Total de registros



- Ações:
- Executar processamento
 - Reprocessar dia
-



Estrutura Final de Telas por Perfil

Funcionário

- Dashboard
- Minhas Horas Extras
- Histórico de Consumo

Gestor

- Dashboard Gerencial
- Relatório de Horas Extras
- Funcionários do Setor
- Aprovação/Ajuste (se houver)

Restaurante

- Consulta de Saldo
- Realizar Débito
- Histórico de Movimentações

Administração

- Cadastro de Funcionários
- Configuração de Regra
- Processamento Diário

Navegação de Telas

Agora vamos organizar isso como **mapa de navegação estruturado**, separado por perfil, mostrando:

- Entrada (login)
- Menu principal
- Fluxos possíveis
- Caminhos alternativos
- Restrições por perfil

Vou estruturar como se fosse documentação de produto.

MAPA DE NAVEGAÇÃO — SISTEMA DE HORAS EXTRAS

1. ENTRADA DO SISTEMA (COMUM A TODOS)

Tela: Login



Validação de perfil



Redirecionamento automático para Dashboard específico

Perfis:

- Funcionário
 - Gestor
 - Restaurante
 - Administrador
-

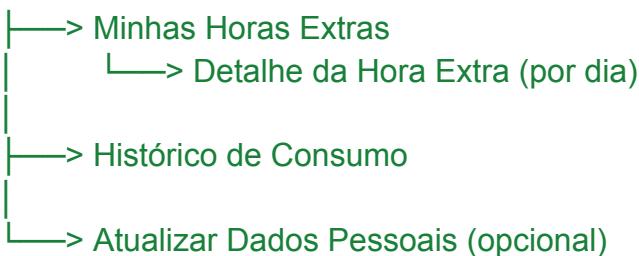
2. FLUXO DE NAVEGAÇÃO — FUNCIONÁRIO

Objetivo: Consulta e transparência

Login



Dashboard do Funcionário



Fluxo detalhado

Dashboard

Exibe:

- Saldo atual
- Total de horas no mês
- Último registro

Ações possíveis:

- Ver horas extras
- Ver consumos

Minhas Horas Extras

Lista por data.

Ação:

- Clicar em um dia → abre **Detalhe da Hora Extra**

Detalhe da Hora Extra

Exibe:

- Total de horas no dia
- Regra aplicada (50% ou 100%)
- Valor convertido

Voltar → Minhas Horas Extras

Histórico de Consumo

Lista:

- Débitos
 - Estornos
 - Saldo após cada operação
-



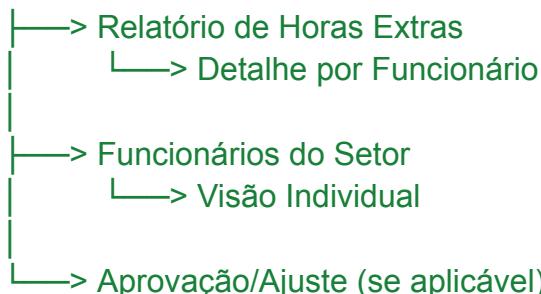
3. FLUXO DE NAVEGAÇÃO — GESTOR

⌚ Objetivo: Análise e controle

Login



Dashboard Gerencial



 **Fluxo detalhado****Dashboard Gerencial**

Exibe:

- Total de horas no período
- Total de saldo gerado
- Funcionários com mais dias > 2h

Ações:

- Filtrar período
 - Ir para relatórios
-

Relatório de Horas Extras

Lista:

- Funcionário
- Setor
- Data
- Horas
- Valor convertido

Ações:

- Filtrar
- Exportar
- Ver detalhe do funcionário

Funcionários do Setor

Lista consolidada:

- Total de horas no mês
- Total convertido

Ação:

- Clicar → Visão Individual
-

Visão Individual

Exibe:

- Histórico completo do funcionário
 - Total mensal
 - Frequência acima de 2h
-

Aprovação / Ajuste (se existir)

Lista:

- Registros pendentes
 - Botões:
 - Aprovar
 - Rejeitar
 - Ajustar (com justificativa)
-

4. FLUXO DE NAVEGAÇÃO — RESTAURANTE

⌚ Objetivo: Operação rápida e segura

Login



Tela de Débito (principal)

- Confirmar Débito
 - Histórico de Movimentações
 - Estorno (se permitido)
-

🔍 Fluxo detalhado

Tela Principal (PDV interno)

Campos:

- Buscar funcionário (matrícula/crachá)
- Exibir saldo
- Inserir valor

Ações:

- Confirmar débito
 - Cancelar
-

Confirmar Débito

Sistema:

- Valida saldo suficiente
- Realiza débito
- Atualiza saldo

Se saldo insuficiente:

- Exibe erro
 - Não permite operação
-

Histórico de Movimentações

Lista:

- Funcionário
 - Valor
 - Data
 - Tipo (débito/estorno)
-

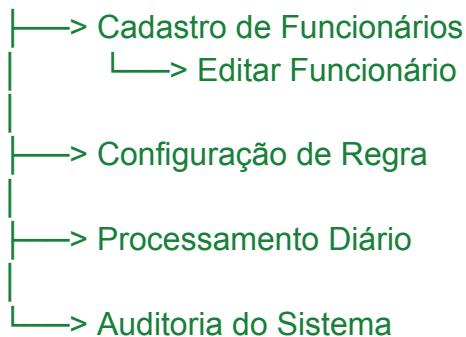
🛠 5. FLUXO DE NAVEGAÇÃO — ADMINISTRADOR

⌚ Objetivo: Configuração e governança

Login



Painel Administrativo



🔍 Fluxo detalhado

Cadastro de Funcionários

Lista:

- Nome
- Setor
- Valor da hora normal
- Status

Ações:

- Criar
 - Editar
 - Desativar
-

Configuração de Regra

Exibe:

- Limite diário (2h)
- Percentual até limite (50%)
- Percentual acima (100%)

Ação:

- Editar
 - Salvar nova versão
-

Processamento Diário

Lista:

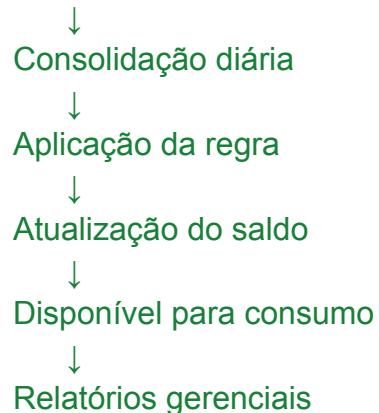
- Dias pendentes
- Total de registros

Ações:

- Processar
 - Reprocessar
-

FLUXO MACRO DO SISTEMA

Registro de Hora Extra



Estrutura Hierárquica Resumida

Funcionário

- Dashboard
- Minhas Horas
- Histórico de Consumo

Gestor

- Dashboard
- Relatórios
- Funcionários do Setor
- Aprovação/Ajuste

Restaurante

- Tela de Débito
- Histórico

Administrador

- Funcionários
 - Regra
 - Processamento
 - Auditoria
-

Resultado

O sistema fica organizado em:

- 1 fluxo de autenticação
- 4 dashboards específicos
- Navegação simples por menu lateral ou superior
- Separação clara de responsabilidades

HISTÓRIAS DE USUÁRIO

HISTÓRIAS DE USUÁRIO — FUNCIONÁRIO

História de Êxito — Funcionário com hora extra até 2h

Perfil: João, analista administrativo

Valor da hora normal: R\$ 20,00

Contexto

João trabalhou **1h30 de hora extra** em uma terça-feira para finalizar um relatório urgente.

Comportamento do sistema

- O sistema registra **1,5h** de hora extra no dia
- Aplica a regra:
 - até 2h → **50% do valor da hora**
- Cálculo:
 - $1,5h \times R\$ 20,00 \times 50\% = R\$ 15,00$
- O valor é convertido em **saldo de benefício**

Resultado percebido pelo funcionário

- João acessa o sistema
- Visualiza:
 - Data da hora extra
 - Quantidade de horas
 - Saldo atualizado

- Nenhuma informação salarial é exibida

Experiência clara, transparente e sem conflitos

História com Problema — Funcionário espera valor diferente

Perfil: Ana, assistente financeira

Valor da hora normal: R\$ 18,00

Contexto

Ana trabalhou **2h10 de hora extra** em um dia, mas acreditava que apenas os 10 minutos excedentes receberiam regra diferente.

Comportamento do sistema

- Sistema consolida o total diário: **2h10**
- Aplica a regra:
 - acima de 2h → **100% do valor da hora**
- Cálculo:
 - $2,17h \times R\$ 18,00 \times 100\% = R\$ 39,06$

Problema percebido

- Ana questiona o cálculo, pois imaginava regra progressiva

Resposta do sistema

- Histórico diário mostra:
 - total de horas
 - regra aplicada
 - valor final

- Regra clara e única por dia

⚠️ Conflito resolvido por transparência da regra

💡 Cenário Possível — Funcionário consulta apenas histórico

Perfil: Carlos, técnico de TI

Contexto

Carlos quer apenas acompanhar suas horas extras acumuladas ao longo do mês.

Comportamento do sistema

- Ele acessa:
 - lista de dias com hora extra
 - total de horas por dia
- Não vê:
 - cálculo financeiro detalhado
 - regras internas
- Apenas saldo final

📌 Sistema cumpre o papel informativo sem expor dados sensíveis



HISTÓRIAS DE USUÁRIO — GESTOR

✓ História de Êxito — Análise de impacto por setor

Perfil: Mariana, gestora do setor de Operações

Contexto

Mariana precisa justificar o aumento de horas extras do setor no último mês.

Comportamento do sistema

- Acessa relatório gerencial
- Visualiza:
 - quantidade de funcionários com hora extra
 - total de horas por setor
 - impacto financeiro estimado (saldo gerado)
- Consegue exportar relatório

Resultado

- Mariana identifica gargalos operacionais
- Usa os dados para redistribuir carga de trabalho

Sistema apoia tomada de decisão estratégica

História com Problema — Ajuste após fechamento

Perfil: Ricardo, gestor de manutenção

Contexto

Um funcionário esqueceu de registrar **30 minutos de hora extra** em um dia já processado.

Comportamento do sistema

- Ricardo solicita ajuste
- O sistema:
 - recalcula o total diário

- identifica mudança de regra (de 50% para 100%)
- gera ajuste no saldo

Problema

- O saldo do funcionário já foi parcialmente consumido

Solução do sistema

- Registra ajuste como **movimentação compensatória**
- Mantém histórico completo

 **Sistema preserva rastreabilidade e consistência**

Cenário Possível — Auditoria preventiva

Perfil: RH

Contexto

RH quer verificar padrões excessivos de hora extra.

Comportamento do sistema

- Relatório mostra:
 - funcionários com recorrência alta
 - dias consecutivos com mais de 2h
- Dados usados para ação preventiva

 **Sistema atua como ferramenta de governança**



HISTÓRIAS DE USUÁRIO — RESTAURANTE

História de Êxito — Consumo simples de saldo

Perfil: Restaurante interno

Contexto

Funcionário usa o restaurante para almoço.

Comportamento do sistema

- Atendente informa o valor: R\$ 18,00
- Sistema verifica saldo
- Debita o valor
- Confirma a operação

Resultado

- Restaurante não vê:
 - origem do saldo
 - salário
 - hora extra
- Apenas confirmação de débito

Fluxo rápido e seguro

História com Problema — Saldo insuficiente

Contexto

Funcionário tenta consumir valor maior que o saldo disponível.

Comportamento do sistema

- Sistema bloqueia operação
- Informa saldo insuficiente
- Não permite saldo negativo

 **Erro tratado sem expor dados**

Cenário Possível — Estorno de consumo

Contexto

Uma cobrança foi feita incorretamente.

Comportamento do sistema

- Restaurante solicita estorno
- Sistema:
 - registra estorno
 - devolve valor ao saldo
 - mantém histórico

 **Controle financeiro interno preservado**

Conclusão

Essas histórias demonstram que o sistema:

- Funciona em cenários reais
- Lida bem com erros e exceções
- Mantém transparência e controle

- Separa responsabilidades corretamente
- Escala para situações futuras

Arquitetura

Agora vamos estruturar isso como **arquitetura real de sistema**, pensando em:

-  Frontend (React)
-  Backend (API)
-  Banco de Dados (PostgreSQL)
-  Processamento de regra diária
-  Autenticação e perfis

Vou organizar como se fosse documento técnico de arquitetura.

1. VISÃO GERAL DA ARQUITETURA

Arquitetura sugerida:



Separação clara de responsabilidades:

- React → Interface e consumo de API
- Backend → Regras de negócio
- Banco → Persistência
- Regra de conversão → Aplicada SOMENTE no backend

2. ARQUITETURA FRONTEND (React)

Estrutura sugerida

```
src/
  └── app/
  └── modules/
    └── auth/
    └── employee/
    └── manager/
    └── restaurant/
    └── admin/
  └── services/
  └── hooks/
  └── contexts/
  └── components/
  └── routes/
```

✳️ Separação por Módulos

👤 employee

- Dashboard
- MinhasHoras
- HistoricoConsumo

👔 manager

- DashboardGerencial
- Relatorios
- FuncionariosSetor
- Aprovacoes

⌚ restaurant

- TelaDebito
- HistoricoMovimentacoes



- Funcionarios
 - Regras
 - Processamento
-

Controle de Rotas (Role Based Access)

Exemplo:

```
/employee/*  
/manager/*  
/restaurant/*  
/admin/*
```

Middleware de rota:

- Verifica JWT
 - Verifica role
-

Comunicação com Backend

Todos os dados vêm da API:

```
GET /employees/me  
GET /overtime/my  
GET /balance/my  
POST /restaurant/debit  
GET /manager/report  
POST /admin/process-day
```

Frontend nunca calcula regra.

3. ARQUITETURA BACKEND

Estrutura sugerida (Node + Clean Architecture)

```
src/
  └── modules/
      ├── auth/
      ├── employee/
      ├── overtime/
      ├── balance/
      ├── restaurant/
      ├── reports/
      ├── rules/
      └── common/
          └── config/
              └── database/
```

CAMADAS

1 Controllers

Recebem requisição HTTP

2 Services

Contêm regra de negócio

3 Repositories

Comunicação com PostgreSQL

LÓGICA CRÍTICA: PROCESSAMENTO DIÁRIO

Service principal:

`processDailyOvertime(date)`

Fluxo interno:

1. Buscar horas extras daquele dia
 2. Agrupar por funcionário
 3. Calcular total diário
 4. Aplicar regra:
if total <= 2h → 50%
else → 100%
 5. Calcular:
horas * valorHoraNormal * percentual
 6. Criar movimentação de crédito
 7. Atualizar saldo
-

█ 4. MODELAGEM DO BANCO (POSTGRESQL)

Agora estrutura consistente com tudo definido.

🔗 Tabela: users

Controle de login e perfil.

id (uuid)
email
password_hash
role (EMPLOYEE, MANAGER, RESTAURANT, ADMIN)
created_at

🔗 Tabela: employees

id (uuid)
user_id (fk users)
name
matricula
sector

hourly_rate
status
created_at

Tabela: overtime

Registro bruto.

id (uuid)
employee_id (fk)
date
hours
status (PENDING, APPROVED, PROCESSED)
created_at

Tabela: rule_versions

Permite versionamento futuro.

id (uuid)
max_hours_limit (2)
percentage_until_limit (0.5)
percentage_above_limit (1.0)
created_at
active (boolean)

Tabela: balance

Saldo atual consolidado.

id (uuid)
employee_id (fk)
current_balance
updated_at

Tabela: balance_movements

Histórico completo (crédito e débito).

id (uuid)
employee_id (fk)
type (CREDIT, DEBIT, ADJUSTMENT)
amount
reference_date
description
created_at

📌 Tabela: restaurant_transactions

(opcional se quiser separar domínio)

id (uuid)
employee_id
amount
created_at

🔄 5. PROCESSAMENTO AUTOMÁTICO

Duas opções:

Opção A — Manual

Admin clica "Processar Dia"

Opção B — Automático

- Cron Job diário (23:59)
- Worker processa horas pendentes

Recomendado:

- 👉 Começar manual (MVP)
 - 👉 Evoluir para automático
-

🔒 6. SEGURANÇA

- JWT com role
 - Middleware de autorização
 - Nunca retornar:
 - salário mensal
 - regra detalhada
 - cálculo interno para restaurante
-

7. RELATÓRIOS (QUERY STRATEGY)

Para performance:

- Índice em:
 - overtime.date
 - employees.sector
 - balance_movements.reference_date
- Usar agregações SQL:

```
SELECT sector, SUM(hours)
FROM overtime
GROUP BY sector;
```

8. SEPARAÇÃO DE DOMÍNIO (IMPORTANTE)

Domínios isolados:

- Overtime → registro bruto
- Rule Engine → cálculo

- Balance → financeiro interno
- Restaurant → apenas débito
- Reports → leitura agregada

Isso evita acoplamento.

⌚ 9. VISÃO MACRO FINAL

[React]



[API Backend]



[Rule Engine Service]



[PostgreSQL]

Regra aplicada apenas no backend.

Restaurante só acessa saldo.

Funcionário só consulta.

Gestor só analisa.

❤️ Resultado

Você agora tem:

- Arquitetura de frontend modular
- Backend organizado por domínio
- Banco normalizado
- Regra isolada
- Sistema escalável

