

A Arte do Recebível

Estratégias para o Controle Supremo



Denise Kurosky



E-Book: Código do Recebível

Dominando os Processos de Contas a Receber

As contas a receber representam o coração financeiro de qualquer negócio. Elas garantem que o fluxo de caixa esteja saudável, assegurando que sua empresa possa operar e crescer. Mas como tornar esse processo eficiente e menos suscetível a erros? A resposta está na estruturação de processos claros e no uso de automações simples, muitas vezes representadas por códigos ou soluções tecnológicas.

Neste e-book, vamos explorar os principais processos de contas a receber de forma prática, com exemplos de código que podem ser aplicados no dia a dia.

01

CADASTRO DE CLIENTES E CRÉDITOS

O que é? Cadastrar clientes significa armazenar informações essenciais para facilitar o acompanhamento de pagamentos.

Cadastro de Clientes

%

```
// clientes = []

def cadastrar_cliente(nome, email, limite_credito):
    cliente = {
        "nome": nome,
        "email": email,
        "limite_credito": limite_credito,
        "saldo_devedor": 0
    }
    clientes.append(cliente)
    return f"Cliente {nome} cadastrado com sucesso!"

print(cadastrar_cliente("Maria Silva", "maria@email.com", 5000))
put your code here
```

snappify.com



02

EMISSÃO DE FATURAS

O que é? A emissão de faturas formaliza o compromisso financeiro entre o cliente e o negócio.

Emissão de Faturas

%

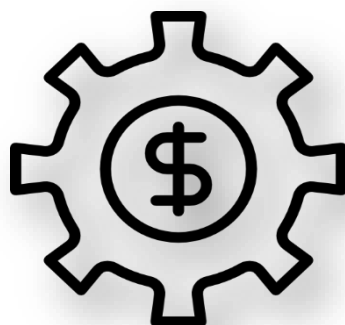
```
from datetime import datetime

def emitir_fatura(cliente, valor):
    if cliente["saldo_devedor"] + valor > cliente["limite_credito"]:
        return "Erro: Limite de crédito excedido."

    fatura = {
        "cliente": cliente["nome"],
        "valor": valor,
        "data_emissao": datetime.now().strftime("%d/%m/%Y"),
        "status": "Pendente"
    }
    cliente["saldo_devedor"] += valor
    return f"Fatura emitida: {fatura}"

cliente_exemplo = clientes[0]
print(emitir_fatura(cliente_exemplo, 2000))
```

snappify.com



03

COBRANÇA DE PAGAMENTOS



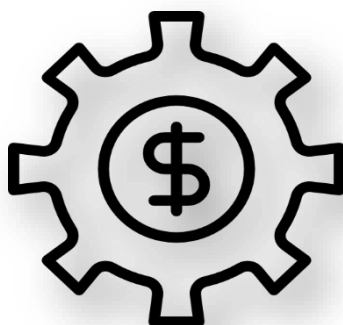
O que é? O processo de cobrança garante que os pagamentos sejam feitos no prazo, preservando o fluxo de caixa.

Cobrança de Pagamentos

%

```
def registrar_pagamento(cliente, valor):  
    if valor > cliente["saldo_devedor"]:  
        return "Erro: Pagamento excede o saldo devedor."  
  
    cliente["saldo_devedor"] -= valor  
    return f"Pagamento de R${valor:.2f} registrado com sucesso para {cliente['nome']}."  
  
print(registrar_pagamento(cliente_exemplo, 1500))
```

snappify.com



04

RELATÓRIOS DE CONTAS A RECEBER



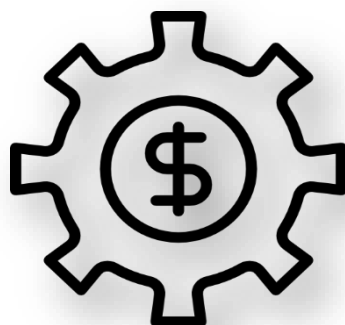
O que é? Relatórios ajudam a visualizar o status das contas a receber, permitindo decisões mais informadas.

Relatório de Contas a Receber

%

```
def gerar_relatorio():  
    print("Relatório de Contas a Receber:")  
    for cliente in clientes:  
        print(f"Cliente: {cliente['nome']}, Saldo Devedor: R${cliente['saldo_devedor']:.2f}")  
  
gerar_relatorio()
```

snappify.com



05

AUTOMAÇÃO DE LEMBRETES DE PAGAMENTO



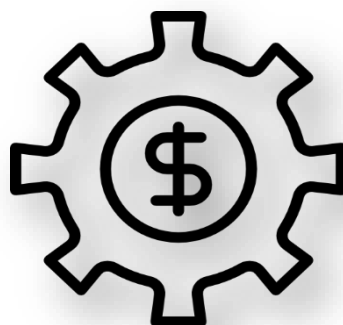
O que é? Automatizar lembretes reduz atrasos e melhora o relacionamento com o cliente.

Automação de Lembretes de Pagamento

%

```
def enviar_lembrete(cliente):  
    if cliente["saldo_devedor"] > 0:  
        return f"Lembrete enviado para {cliente['email']}: Você tem um saldo devedor de R${cliente['saldo_devedor']:.2f}."  
  
print(enviar_lembrete(cliente_exemplo))
```

snappify.com



AGRADECIMENTO



Obrigada, por ler até aqui!



%

A você, leitor(a), meu mais sincero agradecimento por dedicar seu tempo a este e-book.

Sua busca por conhecimento e melhoria contínua é inspiradora e, sem dúvida, essencial para o sucesso no mundo das finanças.

Este conteúdo foi gerado com fins didáticos de aprendizagem, não foi realizado uma validação humana no conteúdo e pode conter erros gerados por uma IA.

Com gratidão,

