**✅ Parte 1: Entidades Principais (Modelos Django)**

**🧱 Modelos baseados nos módulos que você criou:**

**1. User (Usuário)**

Responsável por vincular todos os dados a cada pessoa que usa o app.

python

CopiarEditar

from django.contrib.auth.models import AbstractUser

class User(AbstractUser):

email = models.EmailField(unique=True)

# Outros campos personalizados futuros

**2. Categoria**

Usada para classificar receitas, despesas e vendas.

python

CopiarEditar

class Category(models.Model):

name = models.CharField(max\_length=100)

TYPE\_CHOICES = [

('income', 'Receita'),

('expense', 'Despesa'),

('sale', 'Venda'),

]

type = models.CharField(max\_length=10, choices=TYPE\_CHOICES)

user = models.ForeignKey(User, on\_delete=models.CASCADE)

def \_\_str\_\_(self):

return f"{self.name} ({self.get\_type\_display()})"

**3. Receita (Income)**

python

CopiarEditar

class Income(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

category = models.ForeignKey(Category, limit\_choices\_to={'type': 'income'}, on\_delete=models.SET\_NULL, null=True, blank=True)

amount = models.DecimalField(max\_digits=12, decimal\_places=2)

description = models.CharField(max\_length=255, blank=True, null=True)

received\_at = models.DateField()

created\_at = models.DateTimeField(auto\_now\_add=True)

**4. Despesa (Expense)**

python

CopiarEditar

class Expense(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

category = models.ForeignKey(Category, limit\_choices\_to={'type': 'expense'}, on\_delete=models.SET\_NULL, null=True, blank=True)

amount = models.DecimalField(max\_digits=12, decimal\_places=2)

description = models.CharField(max\_length=255, blank=True, null=True)

paid\_at = models.DateField()

payment\_method = models.CharField(max\_length=50, choices=[('cash', 'Dinheiro'), ('credit\_card', 'Cartão de Crédito'), ('pix', 'PIX'), ('transfer', 'Transferência')])

is\_installment = models.BooleanField(default=False)

installment\_number = models.PositiveIntegerField(null=True, blank=True)

total\_installments = models.PositiveIntegerField(null=True, blank=True)

created\_at = models.DateTimeField(auto\_now\_add=True)

**5. Venda (Sale)**

python

CopiarEditar

class Sale(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

product\_or\_service = models.CharField(max\_length=255)

client = models.CharField(max\_length=255, blank=True, null=True)

amount = models.DecimalField(max\_digits=12, decimal\_places=2)

received\_at = models.DateField()

payment\_method = models.CharField(max\_length=50, choices=[('cash', 'Dinheiro'), ('pix', 'PIX'), ('transfer', 'Transferência')])

created\_at = models.DateTimeField(auto\_now\_add=True)

**6. Cartão de Crédito (CreditCard)**

python

CopiarEditar

class CreditCard(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

name = models.CharField(max\_length=100)

limit = models.DecimalField(max\_digits=12, decimal\_places=2)

closing\_day = models.PositiveSmallIntegerField()

due\_day = models.PositiveSmallIntegerField()

**7. Transações no Cartão de Crédito (CreditCardTransaction)**

python

CopiarEditar

class CreditCardTransaction(models.Model):

card = models.ForeignKey(CreditCard, on\_delete=models.CASCADE, related\_name='transactions')

category = models.ForeignKey(Category, limit\_choices\_to={'type': 'expense'}, on\_delete=models.SET\_NULL, null=True, blank=True)

amount = models.DecimalField(max\_digits=12, decimal\_places=2)

description = models.CharField(max\_length=255)

transaction\_date = models.DateField()

is\_installment = models.BooleanField(default=False)

installment\_number = models.PositiveIntegerField(null=True, blank=True)

total\_installments = models.PositiveIntegerField(null=True, blank=True)

**8. Investimentos (Investment)**

python

CopiarEditar

class Investment(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

name = models.CharField(max\_length=100)

type = models.CharField(max\_length=50, choices=[('renda\_fixa', 'Renda Fixa'), ('renda\_variavel', 'Renda Variável'), ('cripto', 'Cripto'), ('outros', 'Outros')])

amount = models.DecimalField(max\_digits=12, decimal\_places=2)

applied\_at = models.DateField()

current\_value = models.DecimalField(max\_digits=12, decimal\_places=2, null=True, blank=True)

monthly\_return\_percentage = models.DecimalField(max\_digits=5, decimal\_places=2, null=True, blank=True)

**9. Orçamento (BudgetPlan)**

python

CopiarEditar

class BudgetPlan(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

total\_income\_base = models.DecimalField(max\_digits=12, decimal\_places=2)

month\_reference = models.DateField()

created\_at = models.DateTimeField(auto\_now\_add=True)

**10. Orçamento por Categoria (BudgetCategoryLimit)**

python

CopiarEditar

class BudgetCategoryLimit(models.Model):

budget\_plan = models.ForeignKey(BudgetPlan, on\_delete=models.CASCADE, related\_name='categories')

category = models.ForeignKey(Category, on\_delete=models.CASCADE)

percentage = models.DecimalField(max\_digits=5, decimal\_places=2)

amount\_limit = models.DecimalField(max\_digits=12, decimal\_places=2)