**Nome dos integrantes da equipe:**

* Lucas Eduardo Faian
* Hygor Vinycius Onofre Pereira

**Tema escolhido:** contas a receber e sistema de vendas.

Contas a Receber: Registro de débitos de Clientes e suas baixas/pagamentos – pode ter baixa parcial.

Sistema de vendas: faturamento de produtos com cálculo de juros ou descontos

**Contexto do tema escolhido:**

Escolhemos esses temas porque, dentro da COAMO, existem diversos módulos integrados que conectam diretamente o processo de vendas com o controle financeiro, especialmente o que se refere às contas a receber. A proposta é simular um cenário em que a COAMO fornece produtos de seu catálogo para clientes com CNPJ, e a partir dessas transações, realiza o acompanhamento e controle da carteira de contas a receber.

Atualmente, há uma dificuldade significativa na emissão de relatórios detalhados que permitam visualizar com precisão os valores a receber por **faixa de vencimento**, **produto** ou **período específico**. Essa limitação compromete a clareza e a tomada de decisões financeiras mais assertivas. Por exemplo, sabe-se que há um total de R$ 1 bilhão a receber no mês de janeiro, mas é difícil identificar exatamente **quanto está previsto para a primeira quinzena**, ou ainda, **qual produto representa o maior volume de débitos pendentes**.

Com um sistema mais estruturado e com consultas mais detalhadas, seria possível obter respostas mais precisas e rápidas, melhorando a **visibilidade financeira**, o **planejamento de recebimentos** e a **análise estratégica** da carteira. Essa melhoria traria ganhos diretos na gestão e controle financeiro da empresa.

**Principais funcionalidades que serão desenvolvidas:**

| **Sistema de Vendas** | **Contas a Receber** |
| --- | --- |
| Registra a venda de produtos ou serviços | Gera o título financeiro correspondente à dívida do cliente |
| Aplica regras de desconto, juros e define a forma de pagamento | Define como o cliente irá pagar (prazo, número de parcelas, vencimentos) |
| Emite nota fiscal e fatura | Cria o lançamento financeiro vinculado ao cliente |
| Permite que a venda seja parcelada | Gera múltiplos lançamentos, um para cada parcela a receber |
| Pode adicionar juros por atraso no pagamento | Aplica automaticamente os juros em contas vencidas |
| Oferece desconto para pagamento antecipado | Registra a baixa da dívida com valor menor, conforme o desconto aplicado |
| Conclui o processo de faturamento | Controla o recebimento efetivo dos valores e o saldo da carteira do cliente |

**Estórias de usuário das funcionalidades principais, critérios de aceite e regras de negócio:**

**Estória 1 – Realizar uma venda**

**Como** vendedor  
**Quero** cadastrar uma nova venda para um cliente (CNPJ)  
**Para que** o sistema possa gerar automaticamente a fatura e os lançamentos financeiros.

**Critérios de Aceite:**

* O sistema deve permitir selecionar um cliente existente.
* Deve ser possível adicionar um ou mais produtos do catálogo.
* O sistema deve calcular o valor total da venda com descontos e/ou juros.
* A venda deve gerar uma fatura automaticamente.
* Deve ser possível definir se a venda será à vista ou parcelada.

**Regras de Negócio:**

* O cliente deve estar ativo no sistema.
* Se a venda for parcelada, o número máximo de parcelas é 12.

**Estória 2 – Gerar lançamentos em contas a receber**

**Como** sistema financeiro  
**Quero** gerar automaticamente os títulos a receber com base nas vendas efetuadas  
**Para que** seja possível controlar os valores a serem pagos pelos clientes.

**Critérios de Aceite:**

* Cada parcela da venda deve gerar um título de contas a receber.
* Cada título deve conter valor, vencimento, cliente e status como "Pendente" e “Liquidado”.
* As informações devem ser vinculadas à venda original.

**Regras de Negócio:**

* O vencimento de cada parcela deve obedecer ao intervalo definido na venda.
* O sistema deve permitir incluir juros futuros em caso de atraso.

**Estória 3 – Registrar pagamento (baixa total ou parcial)**

**Como** financeiro  
**Quero** registrar o pagamento de uma parcela  
**Para que** o sistema atualize automaticamente a carteira de contas a receber.

**Critérios de Aceite:**

* O sistema deve permitir registrar uma baixa total ou parcial.
* O valor pago deve ser inferior ou igual ao valor da parcela.
* O sistema deve atualizar o status da parcela para "Paga", "Parcial" ou "Em Aberto".

**Regras de Negócio:**

* Em caso de baixa parcial, o saldo da parcela deve ser recalculado.
* Se houver atraso no pagamento, os juros devem ser aplicados no ato da baixa.

**Estória 4 – Consultar situação financeira de um cliente**

**Como** analista financeiro  
**Quero** visualizar o histórico de vendas e pagamentos de um cliente  
**Para que** eu possa analisar sua adimplência.

**Critérios de Aceite:**

* O sistema deve listar todas as vendas do cliente.
* Deve mostrar o status de cada parcela (paga, vencida, parcial, em aberto).
* Deve exibir o total em aberto e o total já recebido.

**⚙️ Regras de Negócio:**

* Clientes inativos devem continuar com o histórico visível.
* O sistema deve destacar em vermelho os títulos vencidos.

**Estória 5 – Aplicar juros automáticos em parcelas vencidas**

**Como** sistema  
**Quero** calcular e aplicar automaticamente juros sobre parcelas vencidas  
**Para que** o valor atualizado seja apresentado ao registrar a baixa.

**Critérios de Aceite:**

* O sistema deve calcular juros com base na taxa definida em configuração.
* O valor com juros deve ser exibido antes de confirmar o pagamento.

**Regras de Negócio:**

* A taxa de juros padrão é de 1% ao mês.
* O cálculo deve considerar dias de atraso com base na data de vencimento.

### **Checklist de Execução do Projeto: Sistema de Vendas e Contas a Receber**

**Fase 1: Fundação e Planejamento Técnico** *(Objetivo: Criar o alicerce do projeto antes de codificar as funcionalidades)*

* **[ ☐ ] 1. Modelagem Final do Banco de Dados (Oracle):**
  + [ ☐ ] Definir todas as tabelas necessárias: Cliente, Produto, Venda, ItemVenda (para o N:N), TituloReceber, Pagamento, Usuario, Configuracao (para juros/descontos).
  + [ ☐ ] Escrever o script SQL (CREATE TABLE...) para a criação da estrutura física no Oracle, incluindo chaves primárias, estrangeiras e constraints.
* **[ ☐ ] 2. Configuração do Ambiente de Desenvolvimento:**
  + [ ☐ ] Criar um repositório no Git para versionamento do código.
  + [ ☐ ] Configurar um ambiente virtual (venv) para o projeto Python.
  + [ ☐ ] Instalar as bibliotecas iniciais: SQLAlchemy (ORM), cx\_Oracle (driver do banco) e firebase-admin (autenticação).
* **[ ☐ ] 3. Arquitetura de Classes (POO) e ORM:**
  + [ ☐ ] Criar os arquivos .py para cada uma das classes mapeadas do banco (cliente.py, venda.py, etc.).
  + [ ☐ ] Implementar a estrutura básica de cada classe com seus atributos e métodos, aplicando **encapsulamento** (@property).
  + [ ☐ ] Configurar o **SQLAlchemy** para mapear cada classe à sua respectiva tabela no banco de dados.

**Fase 2: Desenvolvimento do Core de Vendas e Contas a Receber** *(Objetivo: Implementar as funcionalidades centrais que geram e movimentam dados)*

* **[ ☐ ] 4. Implementar Cadastros Base (CRUDs):**
  + [ ☐ ] Desenvolver as funcionalidades para Cadastrar, Ler, Atualizar e Deletar Cliente.
  + [ ☐ ] Desenvolver as funcionalidades para Cadastrar, Ler, Atualizar e Deletar Produto (incluindo controle de estoque, conforme regra de negócio).
* **[ ☐ ] 5. Implementar o Processo de Venda (Estória 1):**
  + [ ☐ ] Criar a lógica para iniciar uma Venda, selecionar um Cliente e adicionar Produtos (criando os ItemVenda).
  + [ ☐ ] Implementar o cálculo do valor total, aplicando descontos/juros da venda.
  + [ ☐ ] Implementar a validação das regras de negócio (cliente ativo, estoque disponível).
* **[ ☐ ] 6. Implementar a Geração de Títulos (Estória 2):**
  + [ ☐ ] Desenvolver a função que, ao confirmar a venda, gera automaticamente os objetos TituloReceber, um para cada parcela, com os valores e vencimentos corretos.
* **[ ☐ ] 7. Implementar a Baixa de Pagamentos (Estória 3):**
  + [ ☐ ] Criar a funcionalidade para registrar um Pagamento vinculado a um TituloReceber.
  + [ ☐ ] Garantir que a lógica trate corretamente **baixas parciais** (atualizando o saldo devedor) e **baixas totais** (mudando o status para "Pago").

**Fase 3: Desenvolvimento das Funcionalidades de Suporte e Automação** *(Objetivo: Construir as regras automáticas e as ferramentas de análise)*

* **[ ☐ ] 8. Implementar Regras de Juros e Descontos (Estória 5):**
  + [ ☐ ] Desenvolver a rotina automática que verifica os TituloReceber com data\_vencimento < hoje e status "Pendente" para calcular e aplicar os juros.
  + [ ☐ ] Adicionar a lógica de aplicar descontos por antecipação no momento da baixa do pagamento.
* **[ ☐ ] 9. Implementar Consultas e Relatórios (Estória 4 e 10):**
  + [ ☐ ] Desenvolver a tela/função "Situação Financeira do Cliente", exibindo o histórico e o saldo.
  + [ ☐ ] Implementar as 10 consultas SQL exigidas pelo projeto (usando SQLAlchemy), com foco nas de alta complexidade que resolvem o problema do contexto (ex: "total a receber por faixa de vencimento").
* **[ ☐ ] 10. Desenvolver Item de Criatividade:**
  + [ ☐ ] **Sugestão:** Criar um "Dashboard Financeiro" que usa as consultas complexas para exibir gráficos (com matplotlib ou plotly) do total a receber por período, produtos mais vendidos, etc.

**Fase 4: Finalização, Segurança e Entrega** *(Objetivo: Integrar as partes, proteger o sistema e preparar para a apresentação)*

* **[ ☐ ] 11. Implementar Módulo de Autenticação (Segurança):**
  + [ ☐ ] Configurar o projeto no console do **Firebase** e ativar a autenticação por E-mail/Senha.
  + [ ☐ ] Desenvolver as funções de login e cadastro na aplicação Python que se comunicam com o Firebase.
  + [ ☐ ] Proteger as funcionalidades críticas, permitindo o acesso apenas a usuários autenticados.
* **[ ☐ ] 12. Implementar Funcionalidades Adicionais:**
  + [ ☐ ] Desenvolver a emissão da Fatura com base nos dados da Venda (pode ser um simples arquivo de texto formatado ou um PDF).
* **[ ☐ ] 13. Testes e Validação:**
  + [ ☐ ] Realizar testes para cada Estória de Usuário, verificando todos os Critérios de Aceite e Regras de Negócio.
  + [ ☐ ] Testar cenários de exceção (ex: tentar vender para cliente inativo, pagar mais que o devido, etc.).
* **[ ☐ ] 14. Preparação Final:**
  + [ ☐ ] Revisar e refatorar o código, garantindo que os conceitos de POO foram bem aplicados.
  + [ ☐ ] Preparar a documentação final e a apresentação do projeto.

