**Caso de Uso: Gerenciar Orçamento**

**Atores**

* **Organizador (usuário)** – cria e gerencia os orçamentos de seus eventos.
* **Sistema (Faça a Festa)** – registra, valida e controla os dados orçamentários.
* **Fornecedor (usuário aprovado)** – pode ser vinculado a itens de orçamento por meio de seus serviços.

**Objetivo**

Permitir que o organizador registre, atualize e acompanhe os **itens de orçamento** de um evento, vinculando fornecedores e serviços, acompanhando os custos e monitorando o status de pagamentos.

**Pré-condições**

* O organizador deve estar autenticado.
* Deve existir um **evento ativo** para vinculação do orçamento.
* Apenas fornecedores **aprovados** podem ser vinculados.

**Fluxo Principal**

1. O organizador acessa a opção **“Gerenciar Orçamento”** em um evento.
2. O sistema exibe a lista de orçamentos existentes vinculados ao id\_evento.
3. O organizador pode:

**a) Cadastrar item de orçamento**

* + Informa: categoria, item, custo estimado, anotações (opcional).
  + O sistema cria registro em orcamento (ativo = TRUE, status\_pagamento = 'pendente').

**b) Vincular fornecedor e serviço**

* + Seleciona fornecedor aprovado (id\_fornecedor).
  + Seleciona serviço cadastrado (id\_servico).
  + O sistema registra os vínculos.

**c) Atualizar custos**

* + Atualiza o campo custo\_real após negociação/execução.
  + O sistema mantém ambos os valores (custo\_estimado e custo\_real).

**d) Acompanhar status de pagamento**

* + Registra forma de pagamento (forma\_pagamento), valor pago, e altera status (pendente, pago, atrasado).
  + O sistema salva a atualização no status\_pagamento.

**e) Adicionar anotações**

* + O organizador pode salvar observações no campo anotacoes.

1. O organizador pode filtrar orçamentos por categoria, status ou fornecedor, ou exportar relatórios.

**Fluxos Alternativos**

* **A1 – Dados obrigatórios ausentes**: sistema bloqueia o cadastro.
* **A2 – Fornecedor não aprovado**: sistema impede vínculo ao orçamento.
* **A3 – Exclusão de item**: ao excluir, sistema faz exclusão lógica (ativo = FALSE).
* **A4 – Pagamento divergente**: se valor pago não coincide com custo real, sistema emite alerta, mas permite salvar.

**Pós-condições**

* Orçamento do evento atualizado na tabela orcamento.
* Cada item pode estar vinculado a fornecedor e serviço.
* Status de pagamento atualizado corretamente.

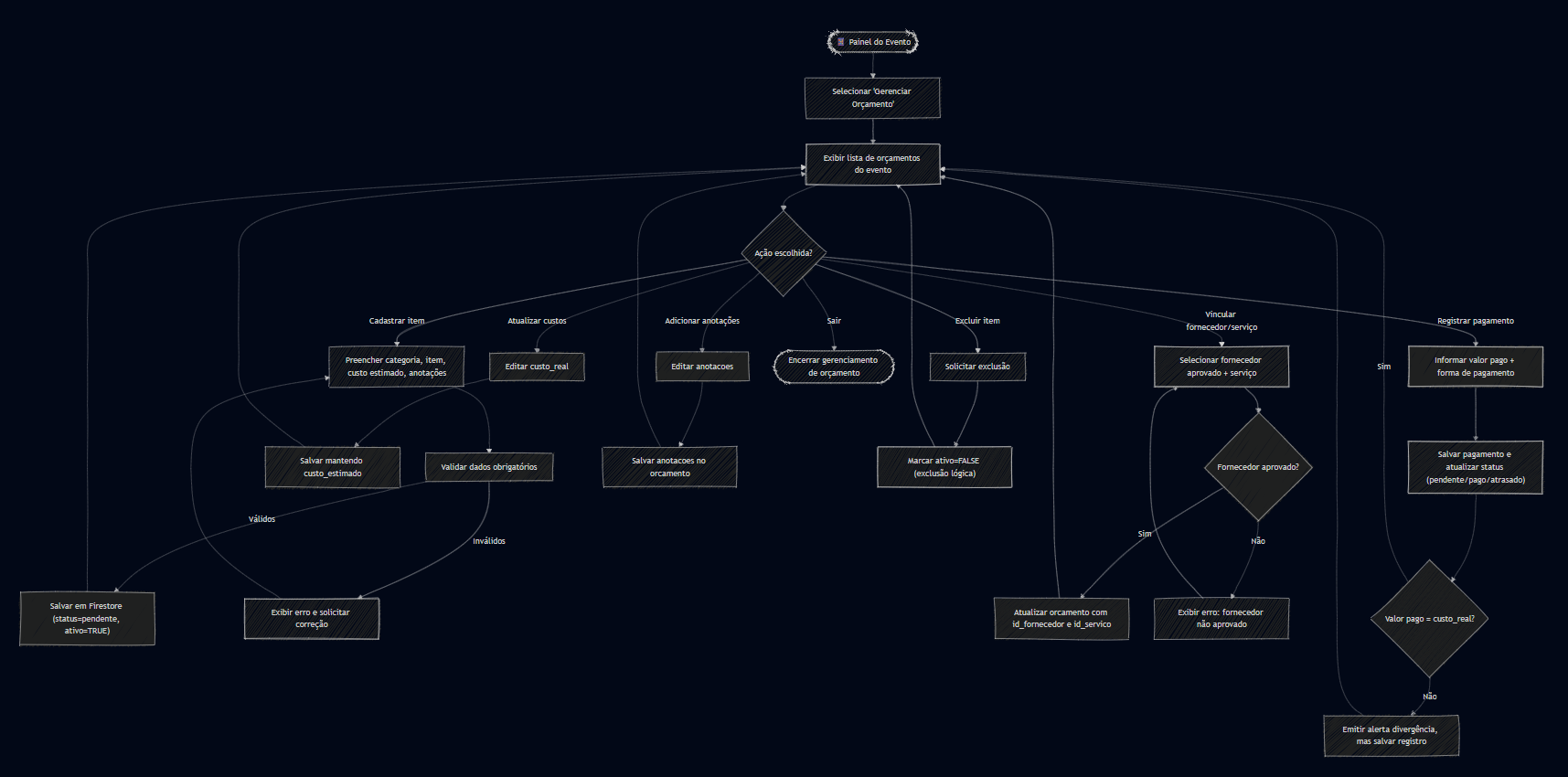
**Regras de Negócio**

* Cada orçamento deve estar obrigatoriamente vinculado a um **evento** (id\_evento).
* categoria, item e custo\_estimado são obrigatórios.
* Apenas fornecedores **aprovados** podem ser vinculados.
* O status de pagamento deve refletir a situação atual (pendente, pago, atrasado).
* Exclusões devem ser lógicas (ativo = FALSE) para manter histórico.

Imagem/Link 001 – Diagrama de sequência

<https://github.com/rrs-sistema/projeto-tcc-faca-a-festa/blob/main/documentos/casos-de-usos/007.1%20-%20Caso%20de%20Uso%20-%20Gerenciar%20Or%C3%A7amento%20-%20Diagrama%20-%20Sequencia.pdf>

Imagem/Link 002 – Diagrama de atividade



Códigos dos diagramas:

001 – Sequência:

sequenceDiagram

    autonumber

    actor Org as Organizador

    participant UI as App Flutter

    participant API as OrcamentoController

    participant SVC as OrcamentoService

    participant VAL as ValidadorDados

    participant DB as Firestore

    participant Forn as Fornecedor (aprovado)

    %% Fluxo inicial

    Org->>UI: Acessa "Gerenciar Orçamento"

    UI->>API: GET /eventos/{id}/orcamentos

    API->>SVC: listarOrcamentos(eventoId)

    SVC->>DB: buscar orçamentos

    DB-->>SVC: lista de orçamentos

    SVC-->>API: retorna orçamentos

    API-->>UI: Exibe lista de orçamentos

    alt Cadastrar item

        Org->>UI: Preenche categoria, item, custo estimado, anotacoes

        UI->>API: POST /orcamentos

        API->>SVC: criarOrcamento(payload)

        SVC->>VAL: validarCamposObrigatorios

        VAL-->>SVC: ok|erro

        alt Dados válidos

            SVC->>DB: salvar orcamento (status=pendente, ativo=TRUE)

            DB-->>SVC: id\_orcamento

            SVC-->>API: orcamento criado

            API-->>UI: Exibe item criado

        else Dados inválidos

            SVC-->>API: 422 erro validação

            API-->>UI: Mostrar erro

        end

    end

    alt Vincular fornecedor/serviço

        Org->>UI: Seleciona fornecedor aprovado + serviço

        UI->>API: PUT /orcamentos/{id}/vincular

        API->>SVC: vincularFornecedorServico

        SVC->>DB: atualizar orcamento (id\_fornecedor, id\_servico)

        DB-->>SVC: ok

        SVC-->>API: vínculo registrado

        API-->>UI: Exibe atualização

    end

    alt Atualizar custos

        Org->>UI: Edita custo\_real

        UI->>API: PUT /orcamentos/{id}/custos

        API->>SVC: atualizarCustos

        SVC->>DB: salvar custo\_real mantendo custo\_estimado

        DB-->>SVC: ok

        SVC-->>API: custos atualizados

        API-->>UI: Exibe valores atualizados

    end

    alt Acompanhar pagamento

        Org->>UI: Registra pagamento (valor, forma)

        UI->>API: POST /pagamentos

        API->>SVC: registrarPagamento

        SVC->>DB: salvar pagamento vinculado ao orcamento

        DB-->>SVC: ok

        SVC-->>API: status\_pagamento atualizado

        API-->>UI: Exibe status atualizado

    end

    alt Adicionar anotações

        Org->>UI: Edita anotacoes

        UI->>API: PUT /orcamentos/{id}/anotacoes

        API->>SVC: atualizarAnotacoes

        SVC->>DB: salvar anotacoes

        DB-->>SVC: ok

        SVC-->>API: anotacoes atualizadas

        API-->>UI: Exibe observacoes

    end

    opt Fluxos Alternativos

        SVC-->>API: Bloqueio se fornecedor não aprovado

        SVC-->>API: Exclusão lógica (ativo=FALSE) ao excluir item

        SVC-->>API: Alerta divergência pagamento

    end

001 – Atividade: