



Plataforma Distribuída de Processamento Colaborativo de Tarefas

Disciplina: Sistemas Distribuído

Professor: Felipe Silva

Integrantes: Marcelo de Jesus, Ronaldo Correia





Objetivo do projeto



Desenvolver uma plataforma distribuida que permita

- SUBMISSÃO DE TAREFAS POR CLIENTES.
- DISTRIBUIÇÃO EFICIENTE PARA MÚLTIPLOS WORKERS.
- ACOMPANHAMENTO DO ESTADO GLOBAL DAS TAREFAS.
- RECUPERAÇÃO AUTOMÁTICA EM CASO DE FALHAS (FAILOVER).



Aplicação prática dos conceitos de sistemas distribuídos

- CONSISTÊNCIA, REPLICAÇÃO E BALANCEAMENTO.





Problema e motivação

Funcionalidades

- **Sistemas centralizados podem falhar e não escalam adequadamente;**
- **Falhas de um único nó podem comprometer todo o processamento;**
- **Motivação: implementar resiliência, tolerância a falhas e balanceamento de carga usando conceitos de sistemas distribuídos.**





Arquitetura

MODELO DISTRIBUÍDO

- O SISTEMA É COMPOSTO POR MÚLTIPLOS PROCESSOS INDEPENDENTES:
- CLIENTE: INTERFACE DE ENTRADA VIA TERMINAL
- ORQUESTRADOR PRINCIPAL: COORDENA TAREFAS E MONITORA WORKERS
- ORQUESTRADOR BACKUP: ASSUME CONTROLE EM CASO DE FALHA
- WORKERS: EXECUTAM TAREFAS PARALELAMENTE

COMUNICAÇÃO

- UTILIZA SOCKETS TCP PARA TROCA DE MENSAGENS ENTRE PROCESSOS
- HEARTBEATS ENVIADOS PERIODICAMENTE PELOS WORKERS E PELO ORQUESTRADOR PRINCIPAL
- TAREFAS E STATUS SERIALIZADOS EM JSON COM JACKSON





Arquitetura

TOLERÂNCIA A FALHAS

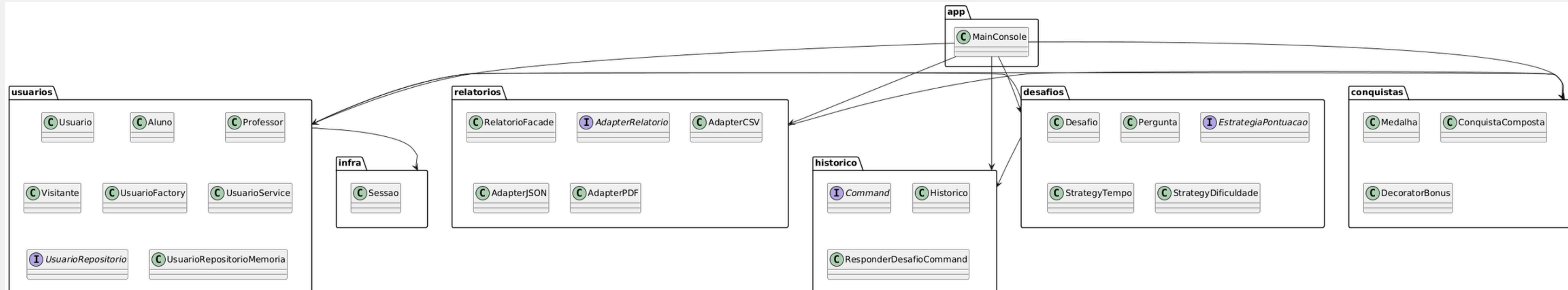
- **FALHAS DE WORKERS SÃO DETECTADAS AUTOMATICAMENTE**
- **O BACKUP ASSUME O CONTROLE SE O PRINCIPAL PARAR DE ENVIAR HEARTBEAT**
- **TAREFAS INCOMPLETAS SÃO REDISTRIBUÍDAS**

ESCALABILIDADE

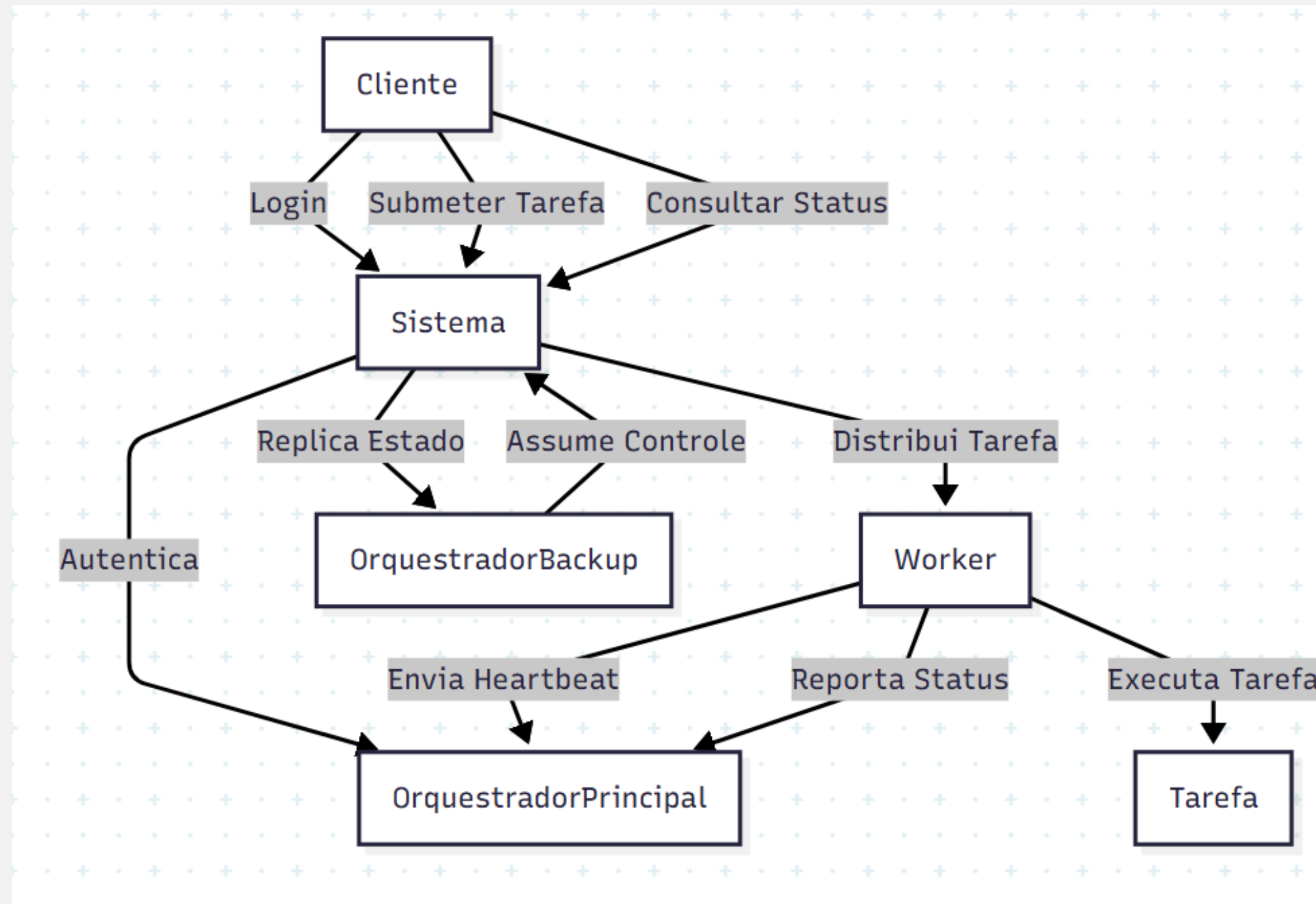
- **SUORTE A MÚLTIPLOS WORKERS**
- **LOADBALANCER DISTRIBUI TAREFAS DE FORMA EFICIENTE**
- **ARQUITETURA MODULAR PERMITE EXPANSÃO FUTURA (EX: INTERFACE GRÁFICA, BANCO DE DADOS)**



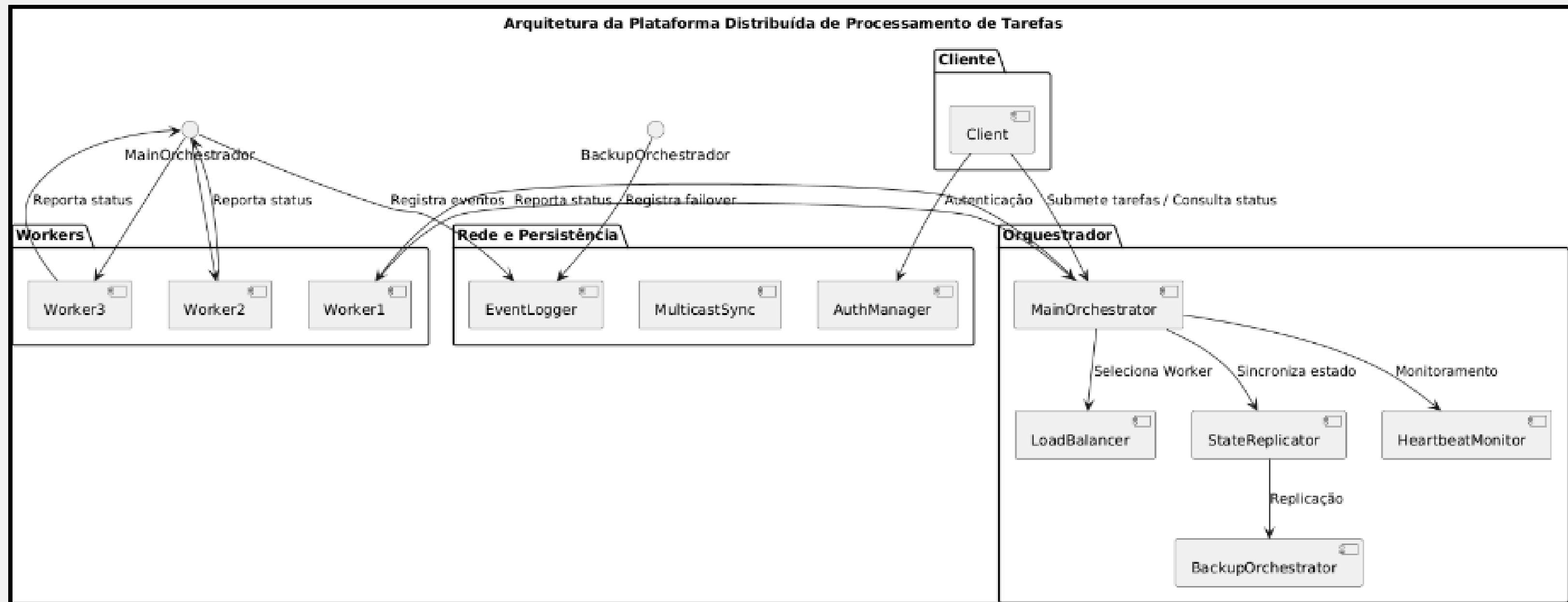
UML Arquitetura Geral



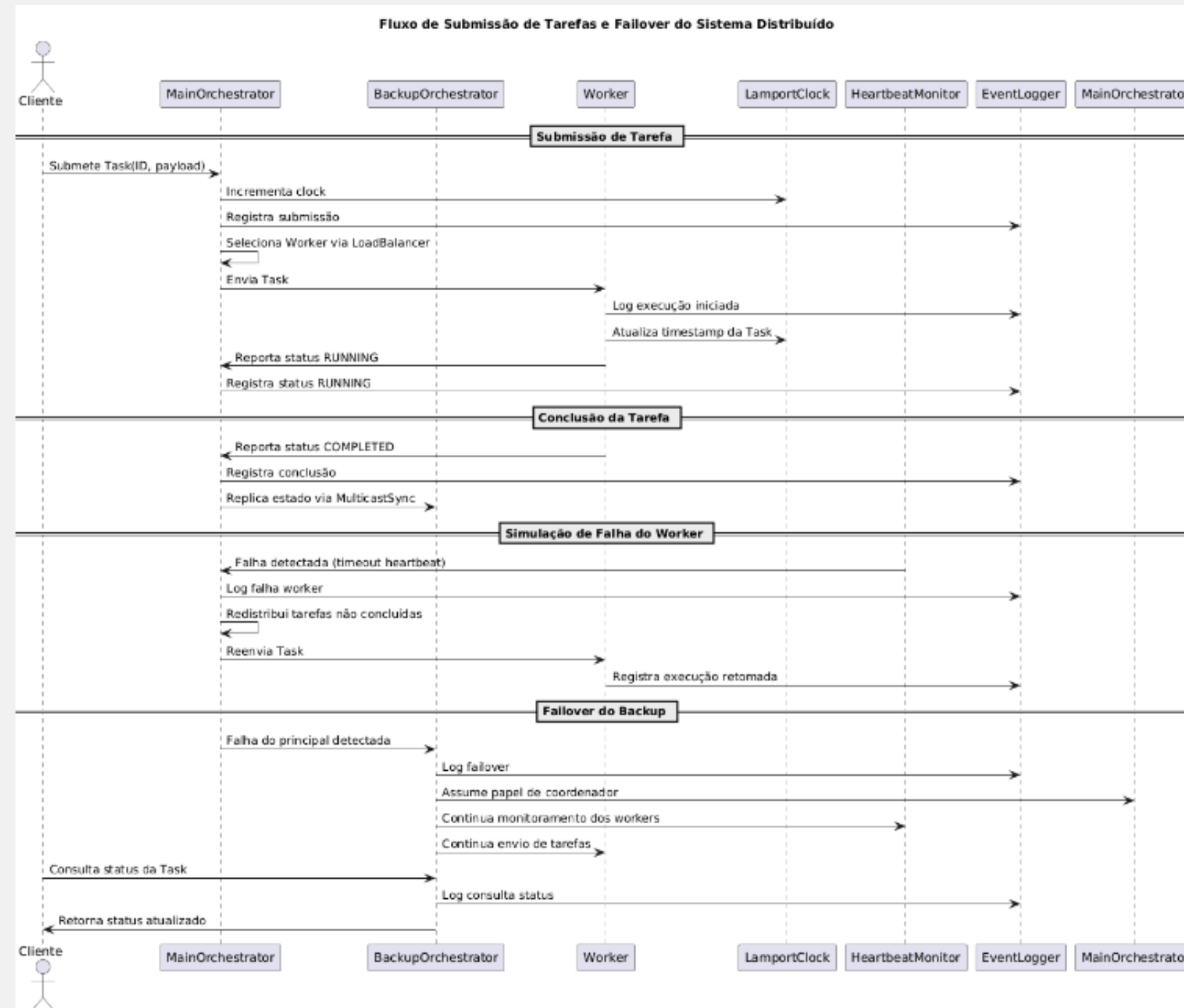
UML Diagrama de casos de uso



UML Diagrama de componentes- Arquitetura Geral



UML Diagrama de Sequência



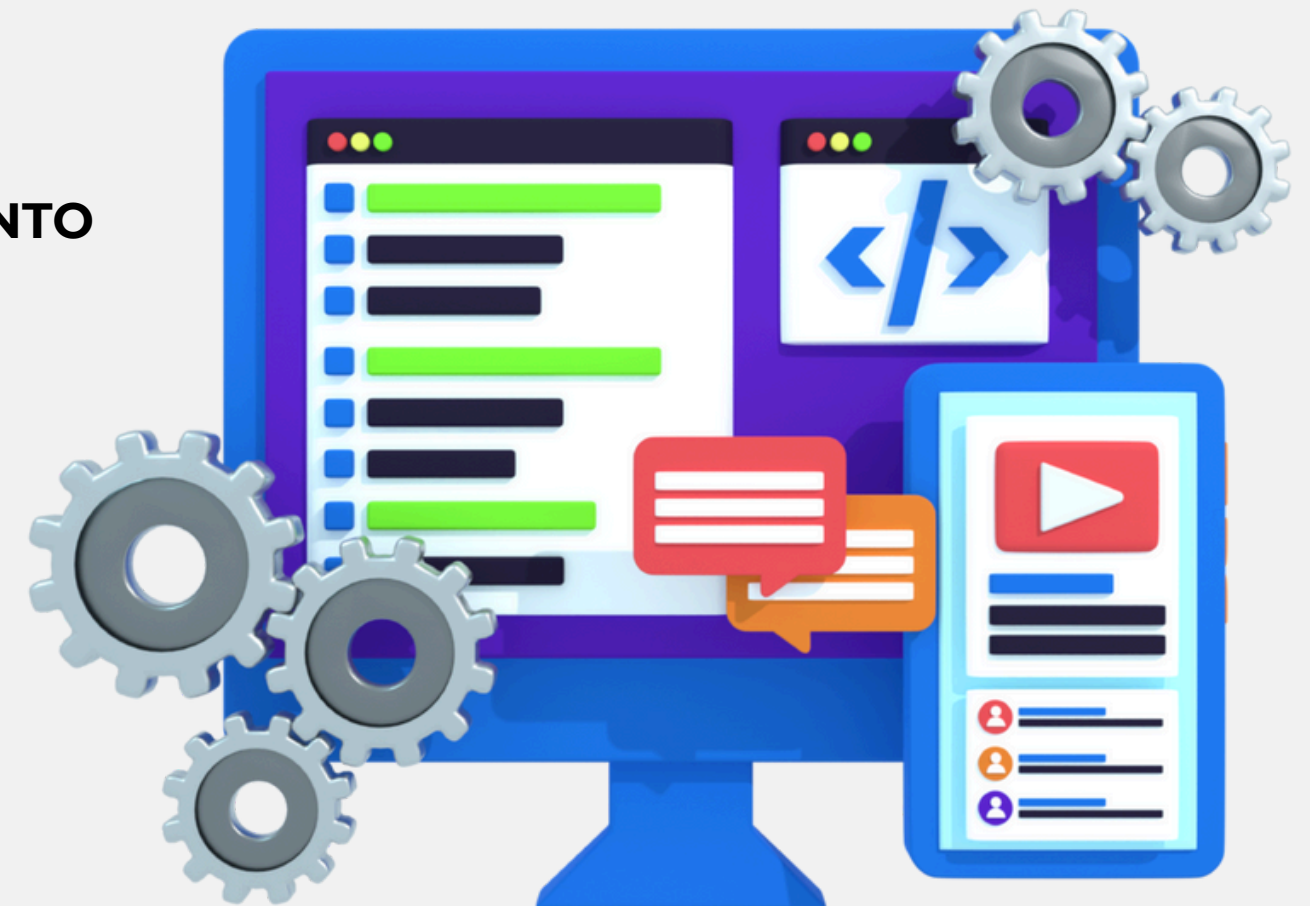
Implementação

TECNOLOGIAS UTILIZADAS

- JAVA 21 — LINGUAGEM PRINCIPAL
- MAVEN — GERENCIAMENTO DE DEPENDÊNCIAS E BUILD
- SOCKETS TCP — COMUNICAÇÃO ENTRE PROCESSOS
- MULTITHREADING — EXECUÇÃO PARALELA DE WORKERS E MONITORAMENTO
- JACKSON — SERIALIZAÇÃO/DESERIALIZAÇÃO DE TAREFAS (JSON)

COMPONENTES IMPLEMENTADOS

- MAINORCHESTRATOR — RECEBE TAREFAS, DISTRIBUI, MONITORA
- BACKUPORCHESTRATOR — FAILOVER AUTOMÁTICO
- WORKER — EXECUTA TAREFAS E ENVIA STATUS
- CLIENT — INTERFACE CLI PARA SUBMISSÃO E CONSULTA
- HEARTBEATMONITOR — DETECTA FALHAS DE WORKERS
- STATEREPLICATOR — SINCRONIZA ESTADO COM O BACKUP
- AUTHMANAGER — AUTENTICAÇÃO PERSISTENTE VIA USERS.TXT
- EVENTLOGGER — LOG DE EVENTOS COM TIMESTAMP



Testes

TESTES REALIZADOS

- **SUBMISSÃO DE TAREFAS: CLIENTE ENVIA E RECEBE STATUS CORRETAMENTE**
- **EXECUÇÃO PARALELA: MÚLTIPLOS WORKERS PROCESSANDO TAREFAS SIMULTANEAMENTE**
- **SIMULAÇÃO DE FALHA: INTERRUPÇÃO DE HEARTBEAT DETECTADA**
- **FAILOVER AUTOMÁTICO: BACKUP ASSUME CONTROLE APÓS QUEDA DO PRINCIPAL**
- **REDISTRIBUIÇÃO DE TAREFAS: TAREFAS SÃO REATRIBUÍDAS APÓS FALHA**
- **LOGS E MONITORAMENTO: EVENTOS REGISTRADOS COM PRECISÃO**





Resultados

RESULTADOS OBTIDOS

- O SISTEMA FOI CAPAZ DE:
- PROCESSAR TAREFAS DISTRIBUÍDAS ENTRE MÚLTIPLOS WORKERS.
- DETECTAR FALHAS DE WORKERS VIA HEARTBEAT.
- REALIZAR FAILOVER AUTOMÁTICO COM O BACKUPORCHESTRATOR.
- MANTER O ESTADO SINCRONIZADO ENTRE ORQUESTRADORES.
- REGISTRAR EVENTOS E STATUS COM PRECISÃO VIA EVENTLOGGER.
- AUTENTICAR USUÁRIOS COM PERSISTÊNCIA LOCAL (USERS.TXT).
- TESTES MOSTRARAM QUE O SISTEMA É TOLERANTE A FALHAS E ESCALÁVEL.

DRIVES
DECISIONS—AND
DESTINY.





Conclusões

- A ARQUITETURA DISTRIBUÍDA IMPLEMENTADA GARANTE CONFIABILIDADE MESMO EM CENÁRIOS DE FALHA.
- O USO DE MODULARIZAÇÃO FACILITOU A MANUTENÇÃO E EVOLUÇÃO DO CÓDIGO.
- A COMUNICAÇÃO VIA SOCKETS TCP E O USO DE MULTITHREADING FORAM ESSENCIAIS PARA O DESEMPENHO.
- O PROJETO DEMONSTROU NA PRÁTICA CONCEITOS COMO:
 1. BALANCEAMENTO DE CARGA
 2. REPLICAÇÃO DE ESTADO
 3. DETECÇÃO DE FALHAS
 4. FAILOVER AUTOMÁTICO
- O SISTEMA ESTÁ PRONTO PARA SER EXPANDIDO COM NOVAS FUNCIONALIDADES (EX: INTERFACE GRÁFICA, PERSISTÊNCIA EM BANCO DE DADOS).

DATA
DECISION
DESTINY