FIAP ADJ8 Feedback Platform

Visão Geral do Projeto

O **FIAP ADJ8 Feedback Platform** é uma plataforma modular e escalável para coleta, gestão e monitoramento de feedbacks de alunos. Ela engloba três subprojetos principais, cada um com responsabilidades claras, permitindo automação de notificações e geração de relatórios semanais.

O projeto segue **arquitetura hexagonal**, garantindo separação de responsabilidades entre domínio, aplicação e infraestrutura, facilitando manutenção, testes e evolução futura.

Objetivos Principais

- Centralizar o gerenciamento de feedbacks de alunos
- Notificar administradores sobre feedbacks urgentes
- Gerar relatórios semanais automatizados
- Fornecer APIs REST para consulta e integração
- Garantir segurança via roles e autenticação HTTP Basic

Tecnologias e Ferramentas

- **Linguagem:** Java 17 / 21
- **Frameworks:** Spring Boot 3, Google Cloud Functions Framework
- **Banco de Dados:** PostgreSQL 16 (Cloud SQL)
- **Serviços GCP:** App Engine, Cloud Functions, Pub/Sub, Cloud Scheduler, Artifact Registry
- **Gerenciamento de Builds:** Maven Multi-módulo
- **Outros:** Shell scripts para gerenciamento de Service Accounts e infraestrutura, Flyway para versionamento de banco

---

Arquitetura Geral

Arquitetura Principal

A arquitetura apresentada segue um modelo **event-driven** com componentes **serverless** e mensageria **Pub/Sub** para integração entre funções e aplicações. A sequência de operação é a seguinte:

Cloud Scheduler

- Um job é agendado para execução semanal, especificamente todo domingo às 00:00.
- O Scheduler dispara eventos de forma programática e confiável, sem necessidade de servidores dedicados.

Pub/Sub: `weekly-feedback-reports`

- O evento disparado pelo Cloud Scheduler é publicado em um tópico Pub/Sub chamado `weekly-feedback-reports`.
- Este tópico atua como um **broker de mensagens**, desacoplando o disparo do evento da execução da função que processa o relatório semanal.

Weekly Report Function

- Uma **Cloud Function** que consome mensagens do tópico `weekly-feedback-reports`.
- Essa função gera ou agrega relatórios semanais baseados nos dados de feedback recebidos.
- Após o processamento, a função disponibiliza os resultados para consumo do **Feedback App** via **REST API**.

Feedback App

- Aplicação central que armazena e gerencia todos os feedbacks dos usuários.
- Consome os relatórios gerados pela **Weekly Report Function** via chamadas REST.
- Identifica feedbacks urgentes ou críticos que precisam de atenção imediata.

Pub/Sub: `feedback-alerts`

- Quando o **Feedback App** detecta feedbacks urgentes, ele publica eventos em um tópico Pub/Sub chamado `feedback-alerts`.
- Esse tópico serve para propagar **alertas críticos** de forma assíncrona e confiável.

Notify Admin Function

- Função **serverless** que consome mensagens do tópico `feedback-alerts`.
- Responsável por enviar notificações ou alertas imediatos para os administradores, garantindo resposta rápida a feedbacks críticos.

Características e Benefícios

- **Arquitetura desacoplada:** Uso de Pub/Sub evita acoplamento direto entre produtores e consumidores.
- **Serverless:** Funções e agendamento não exigem infraestrutura dedicada, garantindo escalabilidade automática.
- **Event-driven:** Todo o fluxo é baseado em eventos (Scheduler → Pub/Sub → Functions → App → Alerts).
- **Segurança e permissões:** Cada função e serviço interage via **Service Accounts** específicas, garantindo **least privilege** e rastreabilidade.
- **Automatização de relatórios e alertas:** O sistema permite gerar relatórios periódicos e disparar alertas críticos sem intervenção manual.

Subprojetos

1. Feedback App

O **Feedback App** é uma aplicação Spring Boot responsável pelo envio, consulta e gestão de feedbacks. Ele disponibiliza endpoints REST para integração e publicação de mensagens em tópicos Pub/Sub para feedbacks urgentes.

Principais Funcionalidades:

- CRUD de feedbacks via REST
- Publicação de mensagens em Pub/Sub para feedbacks urgentes
- Fornecimento de dados para relatórios semanais

- Controle de acesso via roles STUDENT e ADMIN
- Integração com Cloud SQL para persistência de dados

Fluxo de Interações:

- Feedbacks marcados como urgentes são enviados para Pub/Sub (`feedback-alerts`) e consumidos pela função Notify Admin.
- A função Weekly Report consulta dados via REST para gerar relatórios semanais.

Feedback App

---

2. Notify Admin Function

A **Notify Admin Function** é uma função do GCP responsável por enviar notificações de feedbacks urgentes aos administradores.

Principais Características:

- Acionada por mensagens Pub/Sub (`feedback-alerts`)
- Decodifica mensagens e converte avaliações em formato adequado
- Envia email aos administradores com notificações imediatas
- Permissões via Service Accounts dedicadas para deploy e runtime

Fluxo Principal:

1. Feedback App publica mensagem urgente no Pub/Sub
2. Pub/Sub aciona a função Notify Admin
3. Função envia email para administradores

Notify Admin Function

---

3. Weekly Report Function

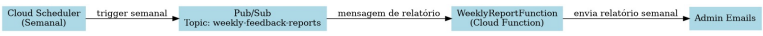
A **Weekly Report Function** gera relatórios semanais de feedbacks para administradores cadastrados.

Principais Características:

- Acionada automaticamente pelo **Cloud Scheduler**
- Recebe trigger via Pub/Sub (`weekly-feedback-reports`)
- Consulta feedbacks da semana via REST do Feedback App
- Gera relatório em formato HTML e envia por email
- Permissões via Service Accounts dedicadas

Fluxo Principal:

1. Cloud Scheduler dispara evento semanal
2. Mensagem publicada em Pub/Sub (`weekly-feedback-reports`)
3. Função consome mensagem, consulta Feedback App e gera relatório
4. Relatório enviado para administradores



Weekly Report Function

Fluxos de Interações

- **Feedback Urgente:**
  Feedback App → Pub/Sub (`feedback-alerts`) → Notify Admin Function

- **Relatório Semanal:**
  Cloud Scheduler → Pub/Sub (`weekly-feedback-reports`) → Weekly Report Function → Feedback App (REST)

Segurança

- Autenticação via **HTTP Basic**
- Roles:
  - STUDENT: acesso a /feedback/**
  - ADMIN: acesso a /admin/** e monitoramento geral
- Usuários pré-configurados para teste em memória
- Senhas codificadas via BCrypt
- Controle de permissões de funções e deploy via Service Accounts GCP

Estrutura Geral de Projetos

fiap-adj8-feedback-platform/ ├─ pom.xml # POM pai (Maven multi-módulo) ├─ fiap-adj8-feedback-app/ # Aplicação principal Spring Boot ├─ functions/ │ ├─ fiap-adj8-feedback-notify-admin-function/ │ └─ fiap-adj8-feedback-weekly-report-function/

yaml Copy code

Objetivos Arquiteturais

- Modularidade e separação de responsabilidades
- Escalabilidade horizontal dos serviços
- Automação de notificações e relatórios
- Governança centralizada via projeto pai Maven
- Facilidade para evolução e manutenção futura

Service Accounts, Provisionamento e Deploy no GCP

Tabela de Service Accounts e papéis

SA	Função / Propósito	Roles Principais	Obse
`sa-infra`	Provisionamento de infraestrutura (Cloud SQL, App Engine, Artifact Registry, Compute, etc.)	cloudsql.admin, artifactregistry.admin, pubsub.admin, cloudscheduler.admin, iam.serviceAccountUser, resourcemanager.projectIamAdmin, compute.networkAdmin, iam.securityAdmin, editor	Usad criar e con perm outro
`sa-`		appengine.deployer, artifactregistry.reader, appengine.serviceAdmin,	Neces para

`deploy-feedback-app`	Deploy da aplicação Feedback App	storage.admin, logging.viewer/logWriter, serviceusage.serviceUsageViewer, cloudbuild.builds.editor, iam.serviceAccountTokenCreator, cloudsql.client, secretmanager.secretAccessor	push Dock deplo Engi acess segre
`sa-deploy-notify-admin`	Deploy da função Cloud Function `notify-admin`	cloudfunctions.developer, pubsub.admin, logging.viewer, storage.admin	Deplo funçã envia por e preci criar/ tópic Pub/S
`sa-deploy-weekly-report`	Deploy da função Cloud Function `weekly-report`	cloudfunctions.developer, pubsub.admin, cloudscheduler.admin, logging.viewer, storage.admin	Deplo funçã geraç relat sema preci tópic Pub/S jobs Sched
`sa-runtime-feedback-app`	Runtime Feedback App	cloudsql.client, pubsub.publisher, logging.logWriter	Execu aplic produ publi mens em tó Pub/S conec Cloud
`sa-runtime-notify-admin`	Runtime Notify Admin	pubsub.subscriber, logging.logWriter	Funç receb mens alert envia consc tópic Pub/S
`sa-runtime-weekly-report`	Runtime Weekly Report	pubsub.publisher, logging.logWriter	Funç publi mens relat sema consc Cloud Sched envia

Sessão de Provisionamento e Interação com GCP

Etapa	Propósito	Ferramenta / Comando	Motivação Visão

Autenticação com SA	Garantir que scripts e deploys usem credenciais corretas	`gcloud auth activate-service-account --key-file=KEY.json`	Evita usar credenciais de usuário e gara automação segura.
Habilitar APIs necessárias	Disponibilizar serviços GCP para o projeto	`gcloud services enable appengine.googleapis.com ...`	Sem APIs habilitadas, recursos com Cloud SQL, Cl Functions e Artifact Regis não funciona
Criar Cloud SQL	Banco de dados PostgreSQL	`gcloud sql instances create ...`	Armazename persistente de dados do Feedback App
Criar Artifact Registry	Repositório de imagens Docker	`gcloud artifacts repositories create ...`	Necessário pa push/pull de imagens Dock usadas pelo A Engine.
Criar App Engine	Hospedar a aplicação	`gcloud app create --region ...`	Ambiente gerenciado pa executar a aplicação e gerenciar escalabilidade automaticam
Build & Push Docker	Preparar imagem da aplicação	`docker build`, `docker tag`, `docker push`	Centraliza a aplicação em container versionado pa deploy no GCI
Deploy App Engine	Colocar a aplicação em produção	`gcloud app deploy`	Publicação fir da aplicação c variáveis de ambiente configuradas.
Criar Tópicos Pub/Sub	Comunicação assíncrona	`gcloud pubsub topics create`	Permite desacoplar funções (Notif Admin, Week Report) do ap principal.
Deploy Cloud Functions	Funções serverless	`gcloud functions deploy`	Automação de alertas e relatórios; funções pode escalar automaticam
Criar Cloud Scheduler Jobs	Agendamento de tarefas	`gcloud scheduler jobs create pubsub`	Garante execu semanal da função de relatórios.
EnvVars / **.env**	Configuração central	`envsubst` / `env.yaml`	Mantém segre e URLs de serviços separados do código-fonte.

Validação final	Teste do deploy	`gcloud pubsub topics publish`, `gcloud functions logs read`	Confirma que comunicação entre serviços funções está funcionando.

Motivo geral da arquitetura e automação

1. **Segurança e segregação de responsabilidades:** cada SA tem permissões mínimas necessárias (princípio do least privilege).
2. **Automação e reprodutibilidade:** scripts permitem provisionar infra, criar roles, SA, deploys e funções de forma repetível.
3. **Desacoplamento e escalabilidade:** Pub/Sub + Cloud Functions + Scheduler permite que cada módulo (alertas, relatórios, app principal) funcione de forma independente.
4. **Observabilidade:** roles de logging e roles de execução permitem monitorar ações de cada SA.

Observação

Os detalhes de cada comando, criação de SAs, provisionamento de infra e deploy estão nos arquivos:

- `manage-sa.sh` (root)
- `infra.sh` (root)
- `deploy.sh` em cada projeto correspondente

PARA DETALHES MAIS TÉCNICOS, visualize o README.md de cada projeto, partindo de: README.md

Clonagem do Repositório e Atualização com Submodules

Para clonar o projeto `fiap-adj8-feedback-platform`, você pode usar **SSH** ou **HTTPS**, dependendo de como suas chaves/credenciais estão configuradas:

Clonando via SSH

```
        git clone --recurse-submodules git@github.com:gabriel-dears/fiap-adj8-feedback-platform.git
```

Clonando via HTTPS

```
        git clone --recurse-submodules https://github.com/gabriel-dears/fiap-adj8-feedback-platform.git
```

⚠ **O parâmetro –recurse-submodules garante que todos os submodules do projeto também sejam clonados automaticamente.**

Claro! Aqui está o conteúdo formatado em Markdown:

Atualizando o Repositório com Submodules

Depois de já ter o repositório clonado, para atualizar o código principal e todos os submodules, use:

```
        git pull --recurse-submodules
```

```
git submodule update --remote
```

- **git pull –recurse-submodules:** Atualiza o branch principal e verifica se os submodules têm commits novos.
- **git submodule update –remote:** Sincroniza os submodules com os commits mais recentes dos repositórios remotos.