

EQUILIBRA

Sistema de Gestão de Tarefas e Trabalho Híbrido

Curso de Análise e Desenvolvimento de Sistemas

Entrega: Global Solution

Integrantes da Equipe:

Raul Rezende

Pietro Donella

Ryan Vеториано

São Paulo - SP
2025

Conteúdo

1 Especificação da Solução Proposta	2
1.1 Explicação Detalhada (Item 2.1)	2
1.2 Justificativas e Objetivos (Item 2.2)	2
1.3 Link para o Vídeo Pitch (Item 2.3)	3
1.4 Modelo Físico e Scripts do Banco de Dados (Item 2.4)	4
1.4.1 Criação das Tabelas (DDL)	4
1.4.2 Inserção de Dados de Exemplo (DML)	4

1 Especificação da Solução Proposta

1.1 Explicação Detalhada (Item 2.1)

O **EQUILIBRA** é uma aplicação Desktop desenvolvida em **Python** utilizando a biblioteca gráfica **PyQt5**, projetada para auxiliar na organização de tarefas e modalidades de trabalho (Híbrido, Presencial, Home Office). O sistema utiliza um banco de dados **Oracle Database** para persistência segura das informações e integra-se com APIs externas para enriquecer a experiência do usuário.

A solução é composta pelos seguintes módulos principais, conforme verificado na análise do código fonte:

- **Gestão de Usuários:** Permite o cadastro (CRUD) de colaboradores, armazenando nome, e-mail e a modalidade de trabalho preferencial (Home Office, Presencial ou Híbrido).
- **Gestão de Tarefas:** Módulo central onde é possível criar, listar, excluir e atualizar o status de tarefas. As tarefas são categorizadas (Pessoal ou Profissional) e vinculadas a um usuário específico, com registro automático de data de criação e conclusão.
- **Relatórios Gerenciais (JSON):** O sistema possui um módulo de análise de dados (`consultas.py`) que gera exportações em formato JSON. As métricas incluem: quantidade de tarefas concluídas por usuário, tempo médio de conclusão (em dias) e percentual de tarefas por tipo.
- **Funcionalidade "Dica do Dia":** Integração via API REST (`api_integra.py`). O sistema consome a API *Advice Slip* para obter um conselho em inglês e, em seguida, utiliza a API *MyMemory* para traduzi-lo automaticamente para português, oferecendo uma mensagem motivacional ao usuário.

1.2 Justificativas e Objetivos (Item 2.2)

Justificativa: No cenário corporativo atual, a gestão do trabalho híbrido e a produtividade individual tornaram-se desafios complexos. Gestores muitas vezes carecem de ferramentas unificadas que permitam não apenas saber a modalidade de trabalho do colaborador, mas também acompanhar o desempenho de suas entregas. O uso de ferramentas dispersas ou anotações manuais resulta em perda de dados e falta de visibilidade sobre a carga de trabalho da equipe. O projeto EQUILIBRA justifica-se pela necessidade de centralizar essas informações em uma interface amigável e robusta. A escolha do Python com PyQt5 garante uma aplicação rápida e nativa, enquanto o banco de dados Oracle assegura a integridade dos dados corporativos, essencial para ambientes empresariais que demandam segurança e escalabilidade.

Objetivos:

- Implementar um sistema CRUD completo para gerenciamento de usuários e suas respectivas tarefas diárias.

- Utilizar Banco de Dados Relacional (Oracle) para manter o histórico de atividades e status das tarefas.
- Desenvolver algoritmos de consulta SQL complexos para extrair inteligência de negócio, como o cálculo do tempo médio entre a criação e a conclusão de uma tarefa.
- Integrar o sistema com serviços web externos (APIs) para demonstrar a capacidade de comunicação entre diferentes plataformas (consumo de dados JSON e tradução em tempo real).
- Fornecer uma interface gráfica moderna (GUI) que facilite a operação por usuários não técnicos.

1.3 Link para o Vídeo Pitch (Item 2.3)

Conforme requisito da entrega, o vídeo demonstrativo da solução (Pitch) encontra-se disponível no link abaixo:

Link do Vídeo (YouTube):

<https://youtu.be/C3XvlbjwR0>

1.4 Modelo Físico e Scripts do Banco de Dados (Item 2.4)

A estrutura do banco de dados foi extraída diretamente do arquivo `script_banco.sql` e do código Python de conexão. O modelo consiste em duas tabelas principais: **USUARIOS** e **TAREFAS**, com relacionamento 1:N (Um usuário possui muitas tarefas).

1.4.1 Criação das Tabelas (DDL)

```
1 -- Tabela para cadastro dos colaboradores e modalidade de trabalho
2 CREATE TABLE USUARIOS (
3     ID_USUARIO NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY KEY,
4     NOME VARCHAR2(100) NOT NULL,
5     EMAIL VARCHAR2(100) UNIQUE NOT NULL,
6     TIPO_TRABALHO VARCHAR2(20) CHECK (TIPO_TRABALHO IN ('Home Office', 'Presencial', 'Híbrido'))
7 );
8
9 -- Tabela para controle das atividades/tarefas
10 CREATE TABLE TAREFAS (
11     ID_TAREFA NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY KEY,
12     ID_USUARIO NUMBER NOT NULL,
13     TITULO VARCHAR2(100) NOT NULL,
14     DESCRICAO VARCHAR2(255),
15     TIPO VARCHAR2(20) CHECK (TIPO IN ('Pessoal', 'Profissional')),
16     DATA_CRIACAO DATE DEFAULT SYSDATE,
17     DATA_CONCLUSAO DATE,
18     STATUS VARCHAR2(20) DEFAULT 'Pendente' CHECK (STATUS IN ('Pendente', 'Concluída')),
19     FOREIGN KEY (ID_USUARIO) REFERENCES USUARIOS(ID_USUARIO)
20 );
```

Listing 1: Estrutura das Tabelas (Baseado no `script_banco.sql`)

1.4.2 Inserção de Dados de Exemplo (DML)

Abaixo seguem comandos de inserção compatíveis com a estrutura definida acima, para fins de teste e validação dos relatórios.

```
1 -- Inserindo Usuários
2 INSERT INTO USUARIOS (NOME, EMAIL, TIPO_TRABALHO)
3 VALUES ('Raul Silva', 'raul@equilibra.com', 'Híbrido');
4
5 INSERT INTO USUARIOS (NOME, EMAIL, TIPO_TRABALHO)
6 VALUES ('Ana Souza', 'ana@equilibra.com', 'Home Office');
7
8 INSERT INTO USUARIOS (NOME, EMAIL, TIPO_TRABALHO)
9 VALUES ('Carlos Pereira', 'carlos@equilibra.com', 'Presencial');
10
11 -- Inserindo Tarefas (Simulando tarefas criadas e concluídas)
12 -- Tarefa 1: Concluída (Para testar cálculo de tempo médio)
13 INSERT INTO TAREFAS (ID_USUARIO, TITULO, DESCRICAO, TIPO, DATA_CRIACAO,
14 DATA_CONCLUSAO, STATUS)
15 VALUES (1, 'Relatório Mensal', 'Finalizar fechamento', 'Profissional',
    SYSDATE-5, SYSDATE-1, 'Concluída');
```

```
16 -- Tarefa 2: Pendente
17 INSERT INTO TAREFAS (ID_USUARIO, TITULO, DESCRICAO, TIPO, DATA_CRIACAO,
18 STATUS)
19 VALUES (1, 'Arrumar Setup', 'Organizar cabos', 'Pessoal', SYSDATE, '
20 Pendente');
21
22 -- Tarefa 3: Concluída Rápida
23 INSERT INTO TAREFAS (ID_USUARIO, TITULO, DESCRICAO, TIPO, DATA_CRIACAO,
24 DATA_CONCLUSAO, STATUS)
25 VALUES (2, 'Reunião Daily', 'Alinhamento diário', 'Profissional', SYSDATE
-1, SYSDATE, 'Concluída');
26
27 COMMIT;
```

Listing 2: Script de Carga de Dados