

FIAP – Graduação em Análise e Desenvolvimento de Sistemas

**Disciplina: Computational Thinking Using Python -
Documentação de Especificação do Projeto**

Professor: Edson de Oliveira

Alunos:

Eduardo Batista Locaspi 561713

Victor Alves Lopes 561833

Sumário

1. Capa.....	1
2. Sumário.....	2
2.1. Explicação Detalhada.....	3
2.2. Justificativas e Objetivos.....	4
2.3. Link para o Vídeo Pitch.....	5
3. Código-Fonte Python (Análise).....	6

Level Up - Sistema de Requalificação e Match de Talentos

Explicação Detalhada

O projeto **Level Up** é um sistema de requalificação profissional focado em preencher a lacuna de talentos em áreas de alta demanda no mercado de trabalho (como Cloud, Cybersegurança e Data Science).

A solução é implementada em **Python** e utiliza o **Banco de Dados Oracle** como backend para gerenciar as entidades centrais do sistema: **Pessoa (T_PESSOA)**, **Empresa (T_EMPRESA)**, **Vaga (T_VAGA_EMPRESA)** e **Evento de Requalificação (T_LVUP_EVENTO)**.

Funcionalidades Principais:

- **Gestão de Pessoas (CRUD):** Cadastro (Create), Visualização/Consulta (Read), Atualização (Update) e Exclusão (Delete) completa de pessoas candidatas, incluindo seus dados de contato e áreas de foco.
- **Gestão de Empresas e Vagas (CRUD):** Operações CRUD para empresas parceiras e suas vagas disponíveis, permitindo manter o banco de vagas sempre atualizado.
- **Gestão de Eventos (CRUD):** Controle dos eventos de requalificação, registrando duração e temas abordados.
- **Consultas Estratégicas e Exportação JSON:** Implementação de 3 consultas SQL avançadas que geram relatórios estratégicos para análise e exportação no formato **JSON**, conforme solicitado.
- **Interface Amigável:** O sistema apresenta um **Menu Interativo** claro, facilitando a navegação do usuário.

A solução atende ao desafio de "**O Futuro do Trabalho**" ao fornecer uma plataforma que gerencia ativamente a requalificação e a alocação de talentos, conectando pessoas qualificadas (via eventos Level Up) diretamente às vagas das empresas parceiras. A utilização do Oracle garante a segurança e a integridade dos dados, enquanto o Python oferece a flexibilidade e o poder computacional para o processamento das informações.

Justificativas e Objetivos

O projeto **Level Up** se justifica pela crescente e urgente necessidade do mercado em requalificar a força de trabalho para acompanhar o ritmo da inovação tecnológica (IA, Cloud, Cibersegurança). A estimativa do Fórum Econômico Mundial de que quase um quarto das profissões mudará até 2027 reforça a importância de um sistema que atue diretamente na mitigação dessa mudança, transformando ameaças em oportunidades.

Objetivos Bem Definidos (Mínimo de 15 linhas):

1. **Garantir a Integridade dos Dados:** Utilizar o SGBD Oracle e a validação em Python para assegurar que todas as informações de Pessoas, Empresas e Eventos sejam consistentes e corretas. Isso é crucial para que os processos de *match* e requalificação sejam baseados em dados confiáveis.
2. **Facilitar a Gestão de Entidades:** Implementar as 4 operações CRUD (Create, Read, Update, Delete) de forma modular para as principais entidades do negócio (Pessoa, Empresa, Vaga, Evento), permitindo aos administradores gerenciar o ciclo de vida dos dados de maneira eficiente.
3. **Fornecer Informação Estratégica:** Desenvolver consultas SQL complexas que revelem padrões de engajamento, áreas de maior demanda e performance de requalificação. O objetivo é transformar os dados brutos em *insights* acionáveis para a tomada de decisão da Level Up e das empresas parceiras.
4. **Atingir a Usabilidade e Robustez:** Desenvolver uma aplicação Python com menu interativo e tratamento de exceções (`try-except`) para garantir que o sistema seja amigável e não falhe em situações inesperadas, como problemas de conexão com o banco de dados ou erros de digitação do usuário.
5. **Promover a Reutilização de Código:** Estruturar o código-fonte em funções bem definidas e documentadas, com clara passagem de parâmetros e retorno de valores, a fim de aumentar a legibilidade e facilitar futuras manutenções e expansões do sistema.
6. **Atender aos Padrões de Interoperabilidade:** Implementar a funcionalidade de exportação dos resultados das consultas para o formato JSON, um padrão de mercado, garantindo que os dados possam ser facilmente consumidos por outras aplicações, como painéis de BI ou plataformas front-end de RH.

O sistema reflete o escopo do projeto ao modelar as relações entre as entidades de requalificação (T_LVUP_EVENTO) e o mercado de trabalho (T_VAGA_EMPRESA), cumprindo a função de um hub de talentos orientado por dados.

Link para o Vídeo Pitch

Pitch proposta: https://www.youtube.com/shorts/_mLTC6II3CQ

Pitch python: <https://www.youtube.com/watch?v=KwZ7Ok4hV8M>

Código-Fonte Python (Análise)

O código-fonte (`main.py`) demonstra a aplicação dos conceitos de Computational Thinking e a conexão com o Oracle, cumprindo os requisitos de 60 pontos.

3.1. Menu Interativo e Estruturas de Controle

- Menu Amigável: O sistema é estruturado em torno da função `main()`, que apresenta um menu principal para a navegação (Gerenciar Pessoas, Gerenciar Empresas, Consultas, etc.).
- Estruturas de Decisão (`if/elif/else`): Utilizadas em todo o código, especialmente dentro dos laços de repetição (`while True`) para direcionar o fluxo do programa com base na escolha do usuário.
- Estruturas de Repetição (`while True`): Usadas para manter o menu principal e os submenus ativos até que o usuário escolha sair. As estruturas `for` são usadas para iteração em listas, como na exibição de resultados de consultas e na exportação JSON.

3.2. Validação de Entradas e Tratamento de Exceções

- Validação de Entradas: Funções dedicadas (como `obter_int`, `obter_string_validada`, `obter_email_validado`) são usadas para garantir a integridade dos dados. Por exemplo:
 - `obter_int` garante que a entrada seja um número.
 - `obter_email_validado` usa expressões regulares (`re`) para validar o formato do e-mail.
- Tratamento de Exceções (`try-except`):
 - O código implementa blocos `try-except` na inicialização do sistema para gerenciar falhas de conexão com o banco de dados Oracle (`oracledb.Error`).
 - Blocos `try-except` são aplicados em todas as operações CRUD (Criação, Leitura, Atualização, Exclusão) para capturar e gerenciar erros de banco (como violações de *Unique Key* ou *Foreign Key*) e entradas inválidas do usuário, garantindo que o programa não seja interrompido abruptamente e fornecendo mensagens amigáveis ao usuário.

3.3. Modularização em Funções

O código é altamente modularizado, facilitando a legibilidade e a manutenção:

- Funções de Exibição: `limpar_terminal()`, `exibir_titulo_centralizado()`.
- Funções de Validação: `obter_int()`, `obter_sim_nao()`, `obter_string_validada()`.
- Funções de Conexão com o BD: `executar_select()`, `executar_dml()`, encapsulando a lógica de conexão e `commit/rollback`.

- Funções CRUD: `inserir_pessoa()`, `consultar_pessoa_por_id()`, `atualizar_pessoa()`, `excluir_pessoa_por_id()`. O uso dessas funções demonstra a passagem de parâmetros (ex: `conn`, `id_pessoa`) e retorno de valores (ex: `ok`, `mensagem_resultado`).

3.4. Consultas e Exportação em Arquivos JSON

A seção de Consultas é implementada para realizar as 3 consultas exigidas, com a opção de exportar o resultado para JSON.

Consultas Implementadas (Exemplos, a serem confirmados no `main.py`):

1. Consulta 1: Pessoas com Maior Histórico de Conclusões por Área de Foco.
 - *Objetivo:* Identificar os candidatos mais engajados e com maior histórico de sucesso.
2. Consulta 2: Vagas Disponíveis e Suas Respectivas Empresas em um Setor Específico.
 - *Objetivo:* Mapear a demanda por vagas em um setor-chave (ex: "Tecnologia").
3. Consulta 3: Eventos de Maior Duração com Maior Taxa de Sucesso na Conclusão.
 - *Objetivo:* Avaliar a eficácia dos eventos de requalificação.

Exportação JSON:

- A função `exportar_para_json()` é chamada após cada consulta.
- Ela recebe o nome do arquivo e os dados do resultado.
- Utiliza a biblioteca `json` do Python para escrever os resultados (`dados_consulta`) no formato JSON em um arquivo, cumprindo o requisito de interoperabilidade de dados.