



PROJETOS

das Competências



- Algoritmos de Apoio à Pesquisa Operacional
- Programar em Linguagem Estruturada (Python)
- Programar em Linguagem Orientada à Objetos Básica (Java)
- Programar em Frontend
- Aplicar Fundamentos de Engenharia de Software



**ANÁLISE E
DESENVOLVIMENTO
DE SISTEMAS**



UNIFACISA
CENTRO UNIVERSITÁRIO



**SISTEMAS DE
INFORMAÇÃO**



COMPETÊNCIAS

04

Programar em Linguagem Estruturada

06

Programar em Linguagem Orientada à Objetos Básica

08

Aplicar Algoritmos de Apoio à Pesquisa Operacional

10

Programar em Frontend

12

Aplicar Fundamentos de Engenharia de Software

SOBRE A AVALIAÇÃO

A avaliação da competência nesta disciplina será baseada em um projeto prático, que compõe 60% da nota final. A entrega do projeto será dividida em duas partes, cada uma representando 30% da nota. Além disso, os 40% restantes da nota serão obtidos por meio de uma atividade individual, que pode ser uma atividade complementar relacionada ao projeto ou uma prova que avalie o conhecimento adquirido ao longo do curso.

SOBRE OS GRUPOS

A entrega do projeto deve ser realizada em grupos de 4 a 8 pessoas, sendo fundamental que todos os grupos observem as datas-limite informadas na ferramenta Canvas. Importante frisar que, apenas um integrante do grupo deve fazer o envio do projeto, devendo este garantir que, na submissão, seja incluída a lista com o nome de todos os integrantes do grupo. Lembrando que, a atividade individual, conforme se é esperado, deve ser realizada por um aluno.

LINKS ÚTEIS

Canvas para Web - <https://unifacisa.instructure.com/>

PROGRAMAR EM LINGUAGEM ESTRUTURADA (PYTHON)

SOBRE

Comumente as empresas criam um setor intitulado Ouvidoria, que acolhe manifestações dos seus clientes, que podem ser reclamações, sugestões ou, raramente, elogios. Neste sentido, a Universidade XYZ contratou um time de 6 a 8 alunos da UniFacisa para desenvolvimento de um sistema de Ouvidoria.

O sistema será detalhado a seguir.

Menu do Sistema

O sistema deve apresentar um menu com 7 opções:

- 1) Listagem das Manifestações
- 2) Listagem de Manifestações por Tipo
- 3) Criar uma nova Manifestação
- 4) Exibir quantidade de manifestações
- 5) Pesquisar uma manifestação por código
- 6) Excluir uma Manifestação pelo Código
- 7) Sair do Sistema.

A **primeira opção (Listagem das Manifestações)**, deve listar todas as manifestações. A **segunda opção (Listagem de Manifestações por Tipo)**, deve questionar ao usuário qual tipo de manifestação gostaria de listar (reclamação, elogio ou sugestão) e, na seguida, serão exibidas as manifestações pelo tipo indicado.

Na **terceira opção (Criar uma nova Manifestação)**, o usuário deve ser questionado com todos os campos necessários para criação de uma nova manifestação. Na **quarta opção (Exibir quantidade de manifestações)** deve ser exibida basicamente a quantidade de manifestações. A **quinta opção (Pesquisar uma manifestação por código)**, deve solicitar o código e informar os campos da manifestação pesquisada. A **sexta opção (Excluir uma Manifestação pelo Código)**, deve solicitar o código e proceder com a exclusão da linha no banco de dados.

PROGRAMAR EM LINGUAGEM ESTRUTURADA (PYTHON)

ENTREGA 1 (40%)

Nesta primeira entrega, devemos criar todas as funcionalidades descritas na página anterior utilizando-se de **listas** na linguagem Python.

Tratamentos Necessários:

- Ao listar, caso não tenha nenhuma manifestação, informar ao usuário que não existem manifestações cadastradas
- Deve ser dado feedback em todas as operações, por exemplo, ao concluir a inclusão de uma nova manifestação no banco de dados, ter a impressão do texto “Nova manifestação criada com sucesso!”

ENTREGA 2 (30%)

A segunda entrega é baseada na primeira entrega, na qual deve ser substituída a lista por um banco de dados relacional, como MySQL ou PostgreSQL.

Para acesso ao MySQL, sugere-se o uso da biblioteca criada pelo professor Daniel Abella, a OperacoesBD, disponível em : <https://github.com/daniel-abella/operacoesbd>

ATIVIDADE INDIVIDUAL (40%)

A avaliação individual será um momento para os alunos demonstrarem a compreensão do projeto de ouvidoria desenvolvido em Python. Durante essa etapa, cada estudante será questionado sobre aspectos específicos de seu trabalho, como possíveis alterações no banco de dados ou a adição de novas funcionalidades ao sistema.

PROGRAMAR EM LINGUAGEM ORIENTADA A OBJETOS BÁSICA

SOBRE

Comumente as empresas criam um setor intitulado Ouvidoria, que acolhe manifestações dos seus clientes, que podem ser reclamações, sugestões ou, raramente, elogios. Neste sentido, a Universidade XYZ contratou um time de 6 a 8 alunos da UniFacisa para desenvolvimento de um sistema de Ouvidoria.

O sistema será detalhado a seguir.

Menu do Sistema

O sistema deve apresentar um menu com 7 opções:

- 1) Listagem das Manifestações
- 2) Listagem de Manifestações por Tipo
- 3) Criar uma nova Manifestação
- 4) Exibir quantidade de manifestações
- 5) Pesquisar uma manifestação por código
- 6) Excluir uma Manifestação pelo Código
- 7) Sair do Sistema.

A **primeira opção (Listagem das Manifestações)**, deve listar todas as manifestações. A **segunda opção (Listagem de Manifestações por Tipo)**, deve questionar ao usuário qual tipo de manifestação gostaria de listar (reclamação, elogio ou sugestão) e, na seguida, serão exibidas as manifestações pelo tipo indicado.

Na **terceira opção (Criar uma nova Manifestação)**, o usuário deve ser questionado com todos os campos necessários para criação de uma nova manifestação. Na **quarta opção (Exibir quantidade de manifestações)** deve ser exibida basicamente a quantidade de manifestações. A **quinta opção (Pesquisar uma manifestação por código)**, deve solicitar o código e informar os campos da manifestação pesquisada. A **sexta opção (Excluir uma Manifestação pelo Código)**, deve solicitar o código e proceder com a exclusão da linha no banco de dados.

PROGRAMAR EM LINGUAGEM ORIENTADA A OBJETOS BÁSICA

ENTREGA 1 (30%)

Nesta primeira entrega, devemos criar todas as funcionalidades descritas na página anterior utilizando-se de **ArrayLists** na linguagem Java.

Tratamentos Necessários:

- Ao listar, caso não tenha nenhuma manifestação, informar ao usuário que não existem manifestações cadastradas
- Deve ser dado feedback em todas as operações, por exemplo, ao concluir a inclusão de uma nova manifestação no banco de dados, ter a impressão do texto “Nova manifestação criada com sucesso!”

ENTREGA 2 (30%)

A segunda entrega é baseada na primeira entrega, na qual deve ser substituído o `ArrayList` por um banco de dados relacional, como MySQL ou PostgreSQL, usando-se a API JDBC (Java Database Connectivity).

Resumo sobre JDBC: <https://daniel-abella.com/quickref/quick-ref-jdbc-v2.pdf>

Vídeo Explicativo: <https://youtu.be/RN6DMxdQEWo>

ATIVIDADE INDIVIDUAL (40%)

A avaliação individual será um momento para os alunos demonstrarem a compreensão do projeto de ouvidoria desenvolvido em Python. Durante essa etapa, cada estudante será questionado sobre aspectos específicos de seu trabalho, como possíveis alterações no banco de dados ou a adição de novas funcionalidades ao sistema.

ALGORITMOS DE APOIO À PESQUISA OPERACIONAL

SOBRE

Os alunos desenvolverão um sistema para gerenciar um acervo de uma biblioteca, que permitirá o armazenamento, organização e busca de livros usando diferentes estruturas de dados e algoritmos.

ENTREGA 1 (30%)

Na primeira entrega, deve implementar a parte do sistema que armazena e organiza os livros, devendo utilizar uma lista sequencial ou encadeada para armazenar os dados dos livros e deve oferecer opções de ordenação por título ou autor.

Critérios de Entrega:

- Estrutura de Dados (Lista Sequencial ou Encadeada) deve ser capaz de armazenar objetos que representem os livros (título, autor, ano de publicação), não podendo ser usado Collections Framework;
- Algoritmos de Ordenação (ex.: Bubble Sort ou Quick Sort) deve ser usado para ordenar os livros por título ou autor;
- Interface Simples deve ser criada, seja em linha de comando ou algum framework de UI, desde que permita adicionar, listar, ordenar e buscar livros no acervo;
- Documentação deve ser criada com explicação sobre a escolha da estrutura de dados e dos algoritmos de ordenação utilizados (1 página).

ENTREGA 2 (30%)

Na segunda entrega, deve ser expandida a primeira entrega com a inclusão de funcionalidades mais avançadas, como a recomendação de livros com base em um critério específico (ex.: livros relacionados a um determinado autor ou tema) utilizando algoritmos avançados e estruturas como árvores ou grafos.

Critérios de Entrega:

- Deve ser implementada ao menos uma técnica avançada (Força Bruta, Algoritmos Gulosos ou Backtracking) para resolver a recomendação de livros ou a busca por um caminho de leitura sugerido;
- Deve ser implementada a estrutura de árvore (ex.: árvore binária de busca) ou grafo (ex.: grafo de relações entre autores ou temas) para organizar e realizar operações avançadas de busca e recomendação;
- Deve ter a documentação sobre a implementação das técnicas e estruturas avançadas, incluindo justificativa das escolhas e análise de complexidade dos algoritmos (até 2 páginas).

ATIVIDADE INDIVIDUAL (40%)

A avaliação individual será um momento para os alunos demonstrarem a compreensão dos algoritmos de ordenação, estruturas de dados e técnicas de algoritmos.

Comumente as empresas criam um setor intitulado Ouvidoria, que acolhe manifestações dos seus clientes, que podem ser reclamações, sugestões ou, raramente, elogios. Neste sentido, a Universidade XYZ contratou um time de 6 a 8 alunos da UniFacisa para desenvolvimento de um sistema de Ouvidoria.

O sistema será detalhado a seguir.

Menu do Sistema

O sistema deve apresentar um menu com 7 opções:

- 1) Listagem das Manifestações
- 2) Listagem de Manifestações por Tipo
- 3) Criar uma nova Manifestação
- 4) Exibir quantidade de manifestações
- 5) Pesquisar uma manifestação por código
- 6) Excluir uma Manifestação pelo Código
- 7) Sair do Sistema.

A **primeira opção (Listagem das Manifestações)**, deve listar todas as manifestações. A **segunda opção (Listagem de Manifestações por Tipo)**, deve questionar ao usuário qual tipo de manifestação gostaria de listar (reclamação, elogio ou sugestão) e, na seguida, serão exibidas as manifestações pelo tipo indicado.

Na **terceira opção (Criar uma nova Manifestação)**, o usuário deve ser questionado com todos os campos necessários para criação de uma nova manifestação. Na **quarta opção (Exibir quantidade de manifestações)** deve ser exibida basicamente a quantidade de manifestações. A **quinta opção (Pesquisar uma manifestação por código)**, deve solicitar o código e informar os campos da manifestação pesquisada. A **sexta opção (Excluir uma Manifestação pelo Código)**, deve solicitar o código e proceder com a exclusão da linha no banco de dados.

PROGRAMAR PARA FRONTEND

ENTREGA 1 (30%)

Na primeira entrega deve ser implementado o backend utilizando Flask em Python para criar um sistema CRUD (*Create, Read, Update, Delete*) para reclamações e usuários.

Critérios de Entrega:

- Backend (Flask): Implementação do CRUD completo para as entidades Reclamação e Usuário, devendo ter integração com BD MySQL.

ENTREGA 2 (30%)

A segunda entrega é focada no frontend com Angular 17 e integração com o backend da entrega 1.

Critérios de Entrega:

- Frontend Básico (HTML e Angular 17) integrado ao Backend com Flask

ATIVIDADE INDIVIDUAL (40%)

A avaliação individual será um momento para os alunos demonstrarem a compreensão do projeto de ouvidoria desenvolvido. Durante essa etapa, cada estudante será questionado sobre aspectos específicos de seu trabalho, como possíveis alterações no banco de dados ou a adição de novas funcionalidades ao sistema.

SOBRE

Comumente as empresas criam um setor intitulado Ouvidoria, que acolhe manifestações dos seus clientes, que podem ser reclamações, sugestões ou, raramente, elogios. Neste sentido, a Universidade XYZ contratou um time de 6 a 8 alunos da UniFacisa para desenvolvimento de um sistema de Ouvidoria.

O sistema será detalhado a seguir.

Menu do Sistema

O sistema deve apresentar um menu com 7 opções:

- 1) Listagem das Manifestações
- 2) Listagem de Manifestações por Tipo
- 3) Criar uma nova Manifestação
- 4) Exibir quantidade de manifestações
- 5) Pesquisar uma manifestação por código
- 6) Excluir uma Manifestação pelo Código
- 7) Sair do Sistema.

A **primeira opção (Listagem das Manifestações)**, deve listar todas as manifestações. A **segunda opção (Listagem de Manifestações por Tipo)**, deve questionar ao usuário qual tipo de manifestação gostaria de listar (reclamação, elogio ou sugestão) e, na seguida, serão exibidas as manifestações pelo tipo indicado.

Na **terceira opção (Criar uma nova Manifestação)**, o usuário deve ser questionado com todos os campos necessários para criação de uma nova manifestação. Na **quarta opção (Exibir quantidade de manifestações)** deve ser exibida basicamente a quantidade de manifestações. A **quinta opção (Pesquisar uma manifestação por código)**, deve solicitar o código e informar os campos da manifestação pesquisada. A **sexta opção (Excluir uma Manifestação pelo Código)**, deve solicitar o código e proceder com a exclusão da linha no banco de dados.

APLICAR FUNDAMENTOS EM ENGENHARIA DE SOFTWARE

ENTREGA 1 (30%)

Na primeira entrega está relacionada à elicitação de requisitos do sistema de Ouvidoria.

Critérios de Entrega:

- Documento de Requisitos: Elaborar um documento detalhado que capture os principais requisitos funcionais do sistema de ouvidoria.
- Protótipo de Alta Fidelidade: Utilizar a ferramenta Figma para criar um protótipo de alta fidelidade que represente a interface do usuário do sistema de ouvidoria.
- User Stories: Escrever User Stories para o sistema, cada uma acompanhada por critérios de aceitação bem definidos.

ENTREGA 2 (30%)

A segunda entrega é focada em Testes de Software e Requisitos Não Funcionais.

Critérios de Entrega:

- Atualização dos Documentos de Requisitos com Inclusão dos Não Funcionais
- Testes End-to-End com Cypress
- Testes de API com Cypress

ATIVIDADE INDIVIDUAL (40%)

A avaliação individual será um momento para os alunos demonstrarem a compreensão dos artefatos criados sobre o projeto de ouvidoria.



UNIFACISA
CENTRO UNIVERSITÁRIO

