

Identificação do Plano de Ensino

| | | | | | |
|----------------------|--------------------------------------------|---------------------------------|---------|--------|-------|
| Unidade | Florianópolis | | | | |
| Curso | Ciência de Dados e Inteligência Artificial | | | | |
| Unidade Curricular | Algoritmos e Programação | | | | |
| Carga Horária da UCR | 144:00 | Ano / Semestre | 2025/01 | Turno: | Manhã |
| Turma | GR GRCD 2025/1 | N° de Situações de Aprendizagem | | | 0 |
| Docente: | Roberto Fabiano Fernandes | | | | |

Informações da Unidade Curricular e Curso

Competência Geral do Curso

O curso foi estruturado com base em competências, conforme preconiza a MSEP, cujos objetivos estão claramente definidos e implementados, considerando o perfil profissional do egresso, a estrutura curricular, o contexto educacional, as características locais e regionais e as novas práticas emergentes no campo do conhecimento relacionado ao curso. Assim, o curso tem por objetivo: Objetivo Geral: Formar profissionais capacitados para atuar de forma ética e responsável no campo de Ciência de Dados e Inteligência Artificial, desenvolvendo soluções avançadas para armazenamento, processamento, análise e transformação de grandes volumes de dados em insights e vantagens competitivas para organizações públicas e privadas. Objetivos Específicos: Para atender ao perfil proposto pelo curso e à formação dos egressos, os profissionais formados devem ser capazes de: • Criar soluções inovadoras para o armazenamento e processamento de grandes volumes de dados, garantindo a segurança, privacidade e integridade das informações, com responsabilidade social e ética. • Realizar análises de dados estratégicas, identificando padrões, tendências e insights que apoiem a tomada de decisões eficazes em diferentes setores, sempre atentos às questões éticas e de segurança dos dados. • Implementar e gerenciar modelos de inteligência artificial, transformando dados em vantagens competitivas para as organizações, observando os aspectos éticos, sociais e de segurança. • Adaptar-se às rápidas inovações tecnológicas no campo da Ciência de Dados e Inteligência Artificial, mantendo-se atualizado em relação às novas metodologias e práticas. • Atuar em equipes multidisciplinares, colaborando no desenvolvimento de soluções tecnológicas que beneficiem a sociedade como um todo.

Função

- FUNÇÃO 1: Cria soluções avançadas para o armazenamento e processamento de grandes volumes de dados, atendendo aos aspectos éticos e sociais, e de segurança dos dados para uma atuação responsável na sociedade.
- FUNÇÃO 2: Realiza análises de dados estratégicas, extraindo insights, identificando padrões, tendências e relações para tomada de decisão, atendendo aos aspectos éticos e sociais, e de segurança dos dados para uma atuação responsável na sociedade.
- FUNÇÃO 3: Implementa modelos de inteligência artificial para a transformação de dados em vantagens competitivas, atendendo aos aspectos éticos e sociais, e de segurança dos dados para uma atuação responsável na sociedade.

Objetivo Geral da Unidade Curricular

- Objetivo Geral: Propiciar o desenvolvimento de capacidades básicas e socioemocionais referente ao uso de algoritmos e programação aplicado à Ciência de Dados e IA.

Plano Aula: Aulas EAD**CH alocada: 27:00****Capacidades a serem trabalhadas**

- Compreender lógica de programação para resolução dos problemas.
- Compreender técnicas de abstração para resolução de problemas.
- Identificar estruturas de dados para construção do algoritmo.
- Compreender expressões aritméticas, relacionais e lógicas para codificação do algoritmo.
- Codificar algoritmos na resolução de problemas.
- Compreender as estruturas de controle e repetição adequadas à lógica dos algoritmos.
- Compreender padrões de nomenclatura e convenções de linguagem na codificação de algoritmos.
- Identificar padrão de nomenclatura de comentários para documentação do código fonte.
- Compreender técnica de ordenação e busca de dados para construção de algoritmos.

Conhecimentos Relacionados

- Introdução a Programação com Python.
- Instalação Anaconda, Jupyter e Google Colab.
- Tipos primitivos e variáveis.
- Operadores lógicos e aritméticos.
- Estruturas condicionais e repetições.
- Funções.
- Programação funcional.
- Manipulação de Strings.
- Estruturas de dados homogêneas: Vetores. Matrizes.
- Exceções.
- Arquivos e E/S.
- Modularização, indentação e comentários de código: Teste de mesa.
- Módulos, pacotes e PIP.
- Visão geral de pacotes populares. Pandas, numpy, statsmodels, scikit-learn, matplotlib, seaborn.
- Organização do Trabalho: Planejamento de atividades. Hierarquia de atividades. Organização de atividades. Controle de atividades.
- Trabalho em equipe: Diversidade e convivência em grupos sociais. Cooperação e colaboração. Divisão de papéis e responsabilidades. Compromisso com objetivos e metas. Relações com o líder. Comunicação e Gestão de Conflitos.
- Iniciativa e pró-atividade: Formas de demonstrar iniciativa. Consequências favoráveis e desfavoráveis. Colaboração e diversidade de perspectivas. Autonomia. Resolução de problemas.
- Ética: Ética e cidadania. Princípios e valores éticos das organizações.
- Ética nas áreas Funcionais.

| | |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Estratégias de Ensino | - Atividade Prática |
| Recursos e Ambientes Pedagógicos | Uso dos materiais complementares em sala de aula e perguntas criadas pelo professor nas avaliações formativas (podendo ser múltipla escolha com descrição da justificativa) |
| Critérios de Avaliação (como vou avaliar) | Avaliação individual; uso das leituras complementares em sala de aula para complementar o conhecimento. |
| Instrumentos de Avaliação da Aprendizagem | Trabalhos postados no ambiente AVA com leituras complementares, conjunto de exercícios. Todos serão sem avaliação. Haverá Avaliações formativas pontuais nas seguintes datas: EAD1 - Avaliação Formativa - Questionários e quizzes; Diálogos e feedbacks; Autoavaliação e avaliação entre pares; Análises de código e resolução de problemas; Data 03/04; Peso: 0,05 EAD2: Avaliação Formativa - Questionários e quizzes; Diálogos e feedbacks; Autoavaliação e avaliação entre pares; Análises de código e resolução de problemas; Data: 08/05; Peso: 0,05 EAD3: Avaliação Formativa - Questionários e quizzes; Diálogos e feedbacks; Autoavaliação e avaliação entre pares; Análises de código e resolução de problemas; Data: 05/06; Peso: 0,05 |

Plano Aula: Módulo 1: Fundamentos da Programação e Lógica Computacional**CH alocada: 14:00**

| | |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Capacidades a serem trabalhadas | <ul style="list-style-type: none">- Compreender lógica de programação para resolução dos problemas.- Compreender técnicas de abstração para resolução de problemas.- Interpretar a simbologia das representações gráficas para definição do fluxo do algoritmo.- Compreender expressões aritméticas, relacionais e lógicas para codificação do algoritmo.- Codificar algoritmos na resolução de problemas.- Compreender padrões de nomenclatura e convenções de linguagem na codificação de algoritmos. |
| Conhecimentos Relacionados | <ul style="list-style-type: none">- Introdução a Programação com Python.- Instalação Anaconda, Jupyter e Google Colab.- Tipos primitivos e variáveis.- Operadores lógicos e aritméticos.- Estruturas condicionais e repetições.- Funções.- Programação funcional.- Manipulação de Strings.- Estruturas de dados homogêneas: Vetores. Matrizes. |

| | |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none"> - Exceções. - Arquivos e E/S. - Modularização, indentação e comentários de código: Teste de mesa. - Módulos, pacotes e PIP. - Visão geral de pacotes populares. Pandas, numpy, statsmodels, scikit-learn, matplotlib, seaborn. - Organização do Trabalho: Planejamento de atividades. Hierarquia de atividades. Organização de atividades. Controle de atividades. - Trabalho em equipe: Diversidade e convivência em grupos sociais. Cooperação e colaboração. Divisão de papéis e responsabilidades. Compromisso com objetivos e metas. Relações com o líder. Comunicação e Gestão de Conflitos. - Iniciativa e pró-atividade: Formas de demonstrar iniciativa. Consequências favoráveis e desfavoráveis. Colaboração e diversidade de perspectivas. Autonomia. Resolução de problemas. - Ética: Ética e cidadania. Princípios e valores éticos das organizações. - Ética nas áreas Funcionais. |
| Estratégias de Ensino | <ul style="list-style-type: none"> - Dinâmica de Grupo - Trabalho em Grupo - Atividade Prática - Exposição Dialogada |
| Recursos e Ambientes Pedagógicos | Em sala de aula. Perguntas criadas pelo professor e eleitas aleatoriamente pelo software: https://wordwall.net/pt/create/picktemplate , ZAP das Perguntas sobre TI e Usar a Aprendizagem Baseada em Projetos |
| Critérios de Avaliação (como vou avaliar) | Uso da roleta aleatória e do ZAP de Perguntas com perguntas sobre os assuntos tratados em aula. Sem pontuação. Com intuito de lembrar o conteúdo apresentado. Fonte: https://wordwall.net/pt/create/picktemplate , Prova de Múltipla Escolha e Discursiva |
| Instrumentos de Avaliação da Aprendizagem | Autoavaliações em todas as aulas e Provas Objetivas de Revisão - e Prova de Múltipla Escolha e Discursiva (AV1 - 03/04/2025) |

Plano Aula: Módulo 2: Trabalhando com Dados e Variáveis

CH alocada: 14:00

| | |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Capacidades a serem trabalhadas | <ul style="list-style-type: none"> - Compreender lógica de programação para resolução dos problemas. - Compreender técnicas de abstração para resolução de problemas. - Interpretar a simbologia das representações gráficas para definição do fluxo do algoritmo. - Compreender expressões aritméticas, relacionais e lógicas para codificação do algoritmo. - Codificar algoritmos na resolução de problemas. - Compreender padrões de nomenclatura e convenções de linguagem na codificação de algoritmos. |
| | |

| | |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Conhecimentos Relacionados | <ul style="list-style-type: none"> - Introdução a Programação com Python. - Instalação Anaconda, Jupyter e Google Colab. - Tipos primitivos e variáveis. - Operadores lógicos e aritméticos. - Estruturas condicionais e repetições. - Funções. - Programação funcional. - Manipulação de Strings. - Estruturas de dados homogêneas: Vetores. Matrizes. - Exceções. - Arquivos e E/S. - Modularização, indentação e comentários de código: Teste de mesa. - Módulos, pacotes e PIP. - Visão geral de pacotes populares. Pandas, numpy, statsmodels, scikit-learn, matplotlib, seaborn. - Organização do Trabalho: Planejamento de atividades. Hierarquia de atividades. Organização de atividades. Controle de atividades. - Trabalho em equipe: Diversidade e convivência em grupos sociais. Cooperação e colaboração. Divisão de papéis e responsabilidades. Compromisso com objetivos e metas. Relações com o líder. Comunicação e Gestão de Conflitos. - Iniciativa e pró-atividade: Formas de demonstrar iniciativa. Consequências favoráveis e desfavoráveis. Colaboração e diversidade de perspectivas. Autonomia. Resolução de problemas. - Ética: Ética e cidadania. Princípios e valores éticos das organizações. - Ética nas áreas Funcionais. |
| Estratégias de Ensino | <ul style="list-style-type: none"> - Dinâmica de Grupo - Trabalho em Grupo - Atividade Prática - Exposição Dialogada |
| Recursos e Ambientes Pedagógicos | Em sala de aula. Perguntas criadas pelo professor e eleitas aleatoriamente pelo software: https://wordwall.net/pt/create/picktemplate , ZAP das Perguntas sobre TI e Usar a Aprendizagem Baseada em Projetos |
| Critérios de Avaliação (como vou avaliar) | Uso da roleta aleatória e do ZAP de Perguntas com perguntas sobre os assuntos tratados em aula. Sem pontuação. Com intuito de relembrar o conteúdo apresentado. Fonte: https://wordwall.net/pt/create/picktemplate , Prova de Múltipla Escolha e Discursiva |
| Instrumentos de Avaliação da Aprendizagem | Autoavaliações em todas as aulas e Provas Objetivas de Revisão - e Prova de Múltipla Escolha e Discursiva (AV1 - 03/04/2025) |

Plano Aula: Módulo 3: Estruturas de Controle e Repetição**CH alocada: 26:00**

| | |
|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Capacidades a serem trabalhadas | <ul style="list-style-type: none">- Compreender lógica de programação para resolução dos problemas.- Compreender técnicas de abstração para resolução de problemas.- Identificar estruturas de dados para construção do algoritmo.- Compreender expressões aritméticas, relacionais e lógicas para codificação do algoritmo.- Codificar algoritmos na resolução de problemas.- Compreender as estruturas de controle e repetição adequadas à lógica dos algoritmos.- Compreender padrões de nomenclatura e convenções de linguagem na codificação de algoritmos. |
| Conhecimentos Relacionados | <ul style="list-style-type: none">- Tipos primitivos e variáveis.- Operadores lógicos e aritméticos.- Estruturas condicionais e repetições.- Funções.- Programação funcional.- Manipulação de Strings.- Estruturas de dados homogêneas: Vetores. Matrizes.- Modularização, indentação e comentários de código: Teste de mesa.- Módulos, pacotes e PIP.- Visão geral de pacotes populares. Pandas, numpy, statsmodels, scikit-learn, matplotlib, seaborn.- Organização do Trabalho: Planejamento de atividades. Hierarquia de atividades. Organização de atividades. Controle de atividades. |
| Estratégias de Ensino | <ul style="list-style-type: none">- Dinâmica de Grupo- Trabalho em Grupo- Atividade Prática- Exposição Dialogada |
| Recursos e Ambientes Pedagógicos | Em sala de aula. Perguntas criadas pelo professor e eleitas aleatoriamente pelo software: https://wordwall.net/pt/create/picktemplate , ZAP das Perguntas sobre TI e Usar a Aprendizagem Baseada em Projetos |
| Critérios de Avaliação (como vou avaliar) | Uso da roleta aleatória e do ZAP de Perguntas com perguntas sobre os assuntos tratados em aula. Sem pontuação. Com intuito de relembrar o conteúdo apresentado. Fonte: https://wordwall.net/pt/create/picktemplate , Prova de Múltipla Escolha e Discursiva |
| Instrumentos de Avaliação da Aprendizagem | Autoavaliações em todas as aulas e Provas Objetivas de Revisão - e Prova de Múltipla Escolha e Discursiva (AV2 - 07/05/2025) |

Plano Aula: Módulo 4: Modularização e Organização do Código**CH alocada: 11:00****Capacidades a serem trabalhadas**

- Compreender lógica de programação para resolução dos problemas.
- Compreender técnicas de abstração para resolução de problemas.
- Interpretar a simbologia das representações gráficas para definição do fluxo do algoritmo.
- Identificar estruturas de dados para construção do algoritmo.
- Compreender expressões aritméticas, relacionais e lógicas para codificação do algoritmo.
- Codificar algoritmos na resolução de problemas.
- Compreender as estruturas de controle e repetição adequadas à lógica dos algoritmos.
- Compreender padrões de nomenclatura e convenções de linguagem na codificação de algoritmos.
- Identificar padrão de nomenclatura de comentários para documentação do código fonte.
- Compreender técnica de ordenação e busca de dados para construção de algoritmos.

Conhecimentos Relacionados

- Introdução a Programação com Python.
- Instalação Anaconda, Jupyter e Google Colab.
- Tipos primitivos e variáveis.
- Operadores lógicos e aritméticos.
- Estruturas condicionais e repetições.
- Funções.
- Programação funcional.
- Manipulação de Strings.
- Estruturas de dados homogêneas: Vetores. Matrizes.
- Exceções.
- Arquivos e E/S.
- Modularização, indentação e comentários de código: Teste de mesa.
- Módulos, pacotes e PIP.
- Visão geral de pacotes populares. Pandas, numpy, statsmodels, scikit-learn, matplotlib, seaborn.
- Organização do Trabalho: Planejamento de atividades. Hierarquia de atividades. Organização de atividades. Controle de atividades.
- Trabalho em equipe: Diversidade e convivência em grupos sociais. Cooperação e colaboração. Divisão de papéis e responsabilidades. Compromisso com objetivos e metas. Relações com o líder. Comunicação e Gestão de Conflitos.
- Iniciativa e pró-atividade: Formas de demonstrar iniciativa. Consequências favoráveis e desfavoráveis. Colaboração e diversidade de perspectivas. Autonomia. Resolução de problemas.
- Ética: Ética e cidadania. Princípios e valores éticos das organizações.
- Ética nas áreas Funcionais.

Estratégias de Ensino

- Design Thinking
- Sala de Aula Invertida
- Gamificação
- Painel Temático

| | |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none">- Seminário- Workshop- Ensaio Tecnológico- Visita Técnica- Dinâmica de Grupo- Trabalho em Grupo- Atividade Prática- Exposição Dialogada |
| Recursos e Ambientes Pedagógicos | Em sala de aula. Perguntas criadas pelo professor e eleitas aleatoriamente pelo software: https://wordwall.net/pt/create/picktemplate , ZAP das Perguntas sobre TI e Usar a Aprendizagem Baseada em Projetos |
| Critérios de Avaliação (como vou avaliar) | Uso da roleta aleatória e do ZAP de Perguntas com perguntas sobre os assuntos tratados em aula. Sem pontuação. Com intuito de lembrar o conteúdo apresentado. Fonte: https://wordwall.net/pt/create/picktemplate , Prova de Múltipla Escolha e Discursiva |
| Instrumentos de Avaliação da Aprendizagem | Autoavaliações em todas as aulas e Provas Objetivas de Revisão - e Prova de Múltipla Escolha e Discursiva (AV2 - 07/05/2025) |

Plano Aula: Módulo 5: Programação Funcional, Estruturas de Dados , Tratamento de Exceção e Arquivos de E/S**CH alocada: 23:00**

| | |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Capacidades a serem trabalhadas | <ul style="list-style-type: none">- Compreender lógica de programação para resolução dos problemas.- Compreender técnicas de abstração para resolução de problemas.- Interpretar a simbologia das representações gráficas para definição do fluxo do algoritmo.- Identificar estruturas de dados para construção do algoritmo.- Compreender expressões aritméticas, relacionais e lógicas para codificação do algoritmo.- Codificar algoritmos na resolução de problemas.- Compreender as estruturas de controle e repetição adequadas à lógica dos algoritmos.- Compreender padrões de nomenclatura e convenções de linguagem na codificação de algoritmos.- Identificar padrão de nomenclatura de comentários para documentação do código fonte.- Compreender técnica de ordenação e busca de dados para construção de algoritmos. |
| Conhecimentos Relacionados | <ul style="list-style-type: none">- Operadores lógicos e aritméticos.- Estruturas condicionais e repetições.- Funções.- Programação funcional.- Manipulação de Strings.- Estruturas de dados homogêneas: Vetores. Matrizes.- Exceções.- Arquivos e E/S.- Modularização, indentação e comentários de código: Teste de mesa. |

| | |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none"> - Módulos, pacotes e PIP. - Visão geral de pacotes populares. Pandas, numpy, statsmodels, scikit-learn, matplotlib, seaborn. - Organização do Trabalho: Planejamento de atividades. Hierarquia de atividades. Organização de atividades. Controle de atividades. - Trabalho em equipe: Diversidade e convivência em grupos sociais. Cooperação e colaboração. Divisão de papéis e responsabilidades. Compromisso com objetivos e metas. Relações com o líder. Comunicação e Gestão de Conflitos. - Iniciativa e pró-atividade: Formas de demonstrar iniciativa. Consequências favoráveis e desfavoráveis. Colaboração e diversidade de perspectivas. Autonomia. Resolução de problemas. - Ética: Ética e cidadania. Princípios e valores éticos das organizações. - Ética nas áreas Funcionais. |
| Estratégias de Ensino | <ul style="list-style-type: none"> - Dinâmica de Grupo - Trabalho em Grupo - Atividade Prática - Exposição Dialogada |
| Recursos e Ambientes Pedagógicos | Em sala de aula. Perguntas criadas pelo professor e eleitas aleatoriamente pelo software: https://wordwall.net/pt/create/picktemplate , ZAP das Perguntas sobre TI e Usar a Aprendizagem Baseada em Projetos |
| Critérios de Avaliação (como vou avaliar) | Uso da roleta aleatória e do ZAP de Perguntas com perguntas sobre os assuntos tratados em aula. Sem pontuação. Com intuito de lembrar o conteúdo apresentado. Fonte: https://wordwall.net/pt/create/picktemplate , Prova de Múltipla Escolha e Discursiva |
| Instrumentos de Avaliação da Aprendizagem | Autoavaliações em todas as aulas e Provas Objetivas de Revisão - e Prova Discursiva (AV3 - 26/06/2025) |

Plano Aula: Módulo 6: Visão Geral de Pacotes Populares para Ciência de Dados

CH alocada: 9:00

| | |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Capacidades a serem trabalhadas | <ul style="list-style-type: none"> - Compreender lógica de programação para resolução dos problemas. - Compreender técnicas de abstração para resolução de problemas. - Interpretar a simbologia das representações gráficas para definição do fluxo do algoritmo. - Identificar estruturas de dados para construção do algoritmo. - Compreender expressões aritméticas, relacionais e lógicas para codificação do algoritmo. - Codificar algoritmos na resolução de problemas. - Compreender as estruturas de controle e repetição adequadas à lógica dos algoritmos. - Compreender padrões de nomenclatura e convenções de linguagem na codificação de algoritmos. - Identificar padrão de nomenclatura de comentários para documentação do código fonte. - Compreender técnica de ordenação e busca de dados para |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| | |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | construção de algoritmos. |
| Conhecimentos Relacionados | <ul style="list-style-type: none">- Introdução a Programação com Python.- Instalação Anaconda, Jupyter e Google Colab.- Tipos primitivos e variáveis.- Operadores lógicos e aritméticos.- Estruturas condicionais e repetições.- Funções.- Programação funcional.- Manipulação de Strings.- Estruturas de dados homogêneas: Vetores. Matrizes.- Exceções.- Arquivos e E/S.- Modularização, indentação e comentários de código: Teste de mesa.- Módulos, pacotes e PIP.- Visão geral de pacotes populares. Pandas, numpy, statsmodels, scikit-learn, matplotlib, seaborn.- Organização do Trabalho: Planejamento de atividades. Hierarquia de atividades. Organização de atividades. Controle de atividades.- Trabalho em equipe: Diversidade e convivência em grupos sociais. Cooperação e colaboração. Divisão de papéis e responsabilidades. Compromisso com objetivos e metas. Relações com o líder. Comunicação e Gestão de Conflitos.- Iniciativa e pró-atividade: Formas de demonstrar iniciativa. Consequências favoráveis e desfavoráveis. Colaboração e diversidade de perspectivas. Autonomia. Resolução de problemas.- Ética: Ética e cidadania. Princípios e valores éticos das organizações.- Ética nas áreas Funcionais. |
| Estratégias de Ensino | <ul style="list-style-type: none">- Design Thinking- Sala de Aula Invertida- Gamificação- Painel Temático- Seminário- Workshop- Ensaio Tecnológico- Visita Técnica- Dinâmica de Grupo- Trabalho em Grupo- Atividade Prática- Exposição Dialogada |

| | |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Recursos e Ambientes Pedagógicos | Em sala de aula. Perguntas criadas pelo professor e eleitas aleatoriamente pelo software: https://wordwall.net/pt/create/picktemplate , ZAP das Perguntas sobre TI e Usar a Aprendizagem Baseada em Projetos |
| Critérios de Avaliação (como vou avaliar) | Uso da roleta aleatória e do ZAP de Perguntas com perguntas sobre os assuntos tratados em aula. Sem pontuação. Com intuito de lembrar o conteúdo apresentado. Fonte: https://wordwall.net/pt/create/picktemplate , Prova de Múltipla Escolha e Discursiva |
| Instrumentos de Avaliação da Aprendizagem | Autoavaliações em todas as aulas e Provas Objetivas de Revisão - Prova Discursiva (AV3 - 26/06/2025) |

Plano Aula: Projeto Final de um Pequeno Sistema de Análise de Dados

CH alocada: 11:00

| | |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Capacidades a serem trabalhadas | <ul style="list-style-type: none"> - Compreender lógica de programação para resolução dos problemas. - Compreender técnicas de abstração para resolução de problemas. - Interpretar a simbologia das representações gráficas para definição do fluxo do algoritmo. - Identificar estruturas de dados para construção do algoritmo. - Compreender expressões aritméticas, relacionais e lógicas para codificação do algoritmo. - Codificar algoritmos na resolução de problemas. - Compreender as estruturas de controle e repetição adequadas à lógica dos algoritmos. - Compreender padrões de nomenclatura e convenções de linguagem na codificação de algoritmos. - Identificar padrão de nomenclatura de comentários para documentação do código fonte. - Compreender técnica de ordenação e busca de dados para construção de algoritmos. |
| Conhecimentos Relacionados | <ul style="list-style-type: none"> - Tipos primitivos e variáveis. - Operadores lógicos e aritméticos. - Estruturas condicionais e repetições. - Funções. - Programação funcional. - Manipulação de Strings. - Estruturas de dados homogêneas: Vetores. Matrizes. - Exceções. - Arquivos e E/S. - Modularização, indentação e comentários de código: Teste de mesa. - Módulos, pacotes e PIP. - Visão geral de pacotes populares. Pandas, numpy, statsmodels, scikit-learn, matplotlib, seaborn. - Organização do Trabalho: Planejamento de atividades. Hierarquia de atividades. Organização de atividades. Controle de atividades. - Trabalho em equipe: Diversidade e convivência em grupos sociais. Cooperação e colaboração. Divisão de papéis e responsabilidades. Compromisso com objetivos e metas. Relações com o líder. Comunicação e Gestão de Conflitos. |

| | |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| | |
| Estratégias de Ensino | - Trabalho em Grupo - Atividade Prática |
| Recursos e Ambientes Pedagógicos | Em sala de aula. Perguntas criadas pelo professor e eleitas aleatoriamente; Aprendizagem Baseada em Projetos |
| Critérios de Avaliação (como vou avaliar) | Apresentação do trabalho final de forma individual com questionamentos feito pelo professor e análise subjetiva da aprendizagem |
| Instrumentos de Avaliação da Aprendizagem | Autoavaliações em todas as aulas e Provas Objetivas de Revisão - Prova Discursiva (AV3 - 26/06/2025) |

Ambientes Pedagógicos

Sala de aula e Ambiente Virtual de Aprendizagem (AVA)

IDE para construção do Código:

<https://colab.google/>

Material na WEB do Professor Fabiano,:

<https://charming-tick-981.notion.site/03-Conte-do-a-ser-Estudado-d6072fcd4f554c5aafd63564f8a816e8?pvs=4>

GPT do Professor Fabiano sobre a linguagem Python:

<https://chatgpt.com/g/g-NHX7XWDW4-logica-computacional-e-algoritmos-com-python>

Referências Bibliográficas**Básica**

1. FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estrutura de dados. 3ª ed. São Paulo (SP): Pearson Prentice Hall, 2005. 218 p. ISBN 9788576050247. (Disponibilidade: 6 unidades)
2. SOUZA, Marco Antonio de et al. Algoritmos e lógica de programação: um texto introdutório para engenharia. 2ª ed. São Paulo (SP): Cengage Learning, 2011. 234 p. ISBN 9788522111299. (Disponibilidade: 3 unidades)
3. VILARIM, Gilvan de Oliveira. Algoritmos: programação para iniciantes. 2ª ed. Rio de Janeiro (RJ): Ciência Moderna, 2004. 270 p. ISBN 857393316X. (Disponibilidade: 6 unidades)

Complementar

4. FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estruturas de dados com aplicações em Python. 4ª ed. São Paulo: Grupo A, 2022. Disponível em: Plataforma BVirtual. Acesso em: 20 jun. 2023.

5. SOUZA, Sérgio Guedes de (org.). Lógica de programação algorítmica. 1ª ed. São Paulo, SP: Pearson, 2014. Disponível em: Plataforma BVirtual. Acesso em: 20 jun. 2023.
6. ARAÚJO, Sandro de. Linguagem de programação (ADS). 1ª ed. São Paulo: Contentus, 2020. Disponível em: Plataforma BVirtual. Acesso em: 20 jun. 2023.
7. CORRÊA, Ana Grasielle Dionísio (org.). Programação I. 1ª ed. São Paulo, SP: Pearson, 2017. Disponível em: Plataforma BVirtual. Acesso em: 20 jun. 2023.
8. KOLIVER, Cristian et al. Introdução à construção de algoritmos: notas de aula. Caxias do Sul: Educus, 2009. 224 p. Disponível em: Plataforma BVirtual. Acesso em: 01 ago. 2024.
9. ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. Fundamentos de programação de computadores: algoritmos, Pascal, C/C++ (padrão ANSI) e Java. 3ª ed. São Paulo: Pearson Prentice Hall, 2012. 588 p. Disponível em: Plataforma BVirtual. Acesso em: 01 ago. 2024.

Outros Instrumentos de Avaliação

AV1- Módulos 1 e 2 - Avaliação teórica - Múltipla Escolha e Discursiva, Construção de Script - Data - 03/04/2025 - Peso: 0,10
AV2 - Módulo 3 e 4 - Avaliação Formativa com Construção de Script de Programação - Data: 07/05/2025 - Peso - 0,15
AV3 IDEA- 11/06 - 0,10;
AV4 - Módulo 5, 6 e Projeto Final - Avaliação Formativa com Construção de Script de Programação - Data: 25/06/2025 - Peso:0,50
EAD1 - Avaliação Formativa - Questionários e quizzes; Diálogos e feedbacks; Autoavaliação e avaliação entre pares; Análises de código e resolução de problemas; Data 03/04; Peso: 0,05
EAD2: Avaliação Formativa - Questionários e quizzes; Diálogos e feedbacks; Autoavaliação e avaliação entre pares; Análises de código e resolução de problemas; Data: 08/05; Peso: 0,05
EAD3: Avaliação Formativa - Questionários e quizzes; Diálogos e feedbacks; Autoavaliação e avaliação entre pares; Análises de código e resolução de problemas; Data:05/06; Peso: 0,05

Observações

Obs.: A Aprendizagem Baseada em Projetos (ABP) desenvolve várias competências pedagógicas importantes e algumas das práticas pedagógicas para a construção do conhecimento são:

1. Resolução de Problemas: Os alunos aprendem a identificar, analisar e resolver problemas complexos de maneira criativa e eficaz.
2. Trabalho em Equipe: ABP incentiva a colaboração, comunicação e trabalho em grupo, ajudando os alunos a desenvolverem habilidades sociais e de cooperação.
3. Pensamento Crítico: Os alunos são estimulados a questionar, avaliar informações e argumentos, e tomar decisões informadas.
4. Autonomia e Gestão do Tempo: A ABP promove a autonomia dos alunos na gestão do próprio aprendizado e no cumprimento de prazos e metas.
5. Comunicação: Os alunos desenvolvem habilidades de comunicação oral e escrita, importantes para apresentar e defender suas ideias e projetos.
6. Aprendizagem Ativa: Os alunos são incentivados a serem ativos no processo de aprendizado, participando ativamente da construção do conhecimento.
7. Iniciativa e Empreendedorismo: A ABP incentiva a proatividade e a capacidade de iniciar e desenvolver

projetos, muitas vezes com um enfoque empreendedor.

8. Uso de Tecnologias: Os alunos aprendem a utilizar ferramentas e tecnologias para pesquisar, criar e apresentar seus projetos.

Critérios de Avaliação

AV1: Peso 0,20 - Múltipla Escolha e Discursiva

1. Pontualidade da Entrega

Entrega no Prazo: Recebe 100% da nota possível para a atividade.

Entrega com Atraso de até 1 Dia: Dedução de 10% da nota total.

Entrega com Atraso de 2 a 3 Dias: Dedução de 20% da nota total.

Entrega com Atraso de 4 a 7 Dias: Dedução de 30% da nota total.

Entrega com Atraso Superior a 7 Dias: Dedução de 50% da nota total.

2. Questões de Múltipla Escolha e Discursivas (100%)

Cada questão correta: 1 pontos.

Cada questão incorreta: 0 pontos.

As respostas seguem o padrão:

Excelente (1 ponto) Resposta completa, clara e bem fundamentada, justificada com os materiais de aula.

Satisfatório (0,5 ponto) Resposta parcialmente correta, mas falta clareza ou detalhes.

Insatisfatório (0 ponto): Resposta incompleta ou incorreta.

AV2: Peso 0,30 - Avaliação Formativa com Construção de Script de Programação

1. Pontualidade da Entrega

Entrega no Prazo: Recebe 100% da nota possível para a atividade.

Entrega com Atraso de até 1 Dia: Dedução de 10% da nota total.

Entrega com Atraso de 2 a 3 Dias: Dedução de 20% da nota total.

Entrega com Atraso de 4 a 7 Dias: Dedução de 30% da nota total.

Entrega com Atraso Superior a 7 Dias: Dedução de 50% da nota total.

2. Funcionalidade do Script (40%)

Excelente (4 pontos) Script funciona perfeitamente, sem erros.

Satisfatório (2 pontos): Script funciona parcialmente, mas com erros significativos.

Insatisfatório (0 pnto): Script não funciona ou tem muitos erros.

3. Qualidade do Código (30%)

Excelente (3 pontos) Código bem organizado, legível, e bem comentado.

Satisfatório (1,5): Código legível, mas desorganizado ou com falta de comentários.

Insatisfatório (0): Código desorganizado e difícil de entender.

4. Originalidade e Criatividade (20%)

Excelente (3): Script inovador, soluções criativas para problemas, sem uso de GPTs.

Satisfatório (1): Script básico, com poucas ideias originais.

Insatisfatório (0): Script sem originalidade, códigos de GPTs

AV3: Peso 0,40 - Avaliação Formativa com Construção de Script de Programação e entrega Individual com questionamento e análise do Professor

Programação

1. Pontualidade da Entrega

Entrega no Prazo: Recebe 100% da nota possível para a atividade.

Entrega com Atraso de até 1 Dia: Dedução de 10% da nota total.

Entrega com Atraso de 2 a 3 Dias: Dedução de 20% da nota total.

Entrega com Atraso de 4 a 7 Dias: Dedução de 30% da nota total.

Entrega com Atraso Superior a 7 Dias: Dedução de 50% da nota total.

2. Funcionalidade do Script (40%)

Excelente (4 pontos) Script funciona perfeitamente, sem erros

Satisfatório (2 pontos): Script funciona parcialmente, mas com erros significativos.

Insatisfatório (0 pnto): Script não funciona ou tem muitos erros.

3. Qualidade do Código (30%)

Excelente (3 pontos) Código bem organizado, legível, e bem comentado.

Satisfatório (1,5): Código legível, mas desorganizado ou com falta de comentários.

Insatisfatório (0): Código desorganizado e difícil de entender.

4. Originalidade e Criatividade (20%)

Excelente (3): Script inovador, soluções criativas para problemas, sem uso de GPTs.

Satisfatório (1): Script básico, com poucas ideias originais.

Insatisfatório (0): Script sem originalidade, códigos de GPTs