**31/05/2022**

Bootcamp Unimed BH, de 12/06 a 26/07

**Vídeo de apresentação**

Vídeo de apresentação

**Mentoria 0: Aula inaugural – Geração Tech Unimed-BH / Fullstack**

Quando: Dia 26/05, 19:00, 2hs, acessado dia 08/06/2022

Quem: Camila Cavalcanti, Venilton FalvoJr, Renan Oliveira (DIO)

Assuntos:

* Sobre a DIO
* Sobre a Unimed-BH
* Quem ministra o bootcamp
* Quem são os mentores
* Sobre o programa
  + Micro learning: aulas curtas
  + Stack tecnológica: front-end, back-end, api e suas tecnologias
* O que é um bootcamp: campo de treinamento, inspirado nos campos de treinamento de militares americanos, se popularizou a partir do Vale do Silício
* Metodologia Bootcamp:
  + Desenvolvimento de projetos
  + Experiência de aprendizagem gamificada
  + Sair da zona de conforto
  + Produções reais e imersivas (prazos reais)
  + Competição e desafios práticos de código
  + Comunidade colaborativa
  + “Confie no processo!”

**Mentoria #1: Carreiras Tech na Unimed BH**

Quando: 30/05, 14:00, 2horas, acessado dia 08/06/2022

Quem: Caio (DIO), Paulo Henrique Amorim Rodrigues, Cesar Magdalena, Rafael Paolinelli (Unimed-BH)

- Apresentação da Unimed e das equipes de TI, dos projetos, dos desafios.

- Dica de ouro: pensar ágil e não tem medo de errar, agilidade é transparência, inspeção e adaptação; tenha confiança, acredita no seu potencial, valoriza sua história pessoal.

**Mentoria #2: Introdução ao banco de dados Oracle**

Quando: 14/06, 14:00, 2 horas, acessado ao vivo

Quem: Denilson Bonatti (Digital Innovation One)

O que é banco de dados relacional .

Oracle é um SGBD (sistema gerenciador de banco de dados) é um dos mais antigos bancos de dados, surgiu no fim dos anos 70. É o primeiro banco de dados relacionais comercializado no mundo.

O MySql foi comprada pela Oracle

Fonte: <https://developpaper.com/why-mongodb-ranks-in-the-top> [...]

Características:

* Restauração e recuperação
* Alto desempenho
* Presença no mercado
* Atualizações

A linguagem SQL tem os mesmos comandos básicos, independente do banco de dados.

Desenvolvedor Oracle ou DBA

* Desenvolvimento
  + Oracle SQL
  + PlSql
  + Modelagem de dados

Diferenças entre o SQL e o PlSql

* SQL: linguagem padrão
* Pl/sql: linguagem proprietária da Oracle

Instalação para uso pessoal: versão Oracle 21c XE (Express Edition). Esta versão possui limitação do tamanho do banco de dados.

* Instalação Windows: padrão Windows (next, next, finish).
  + Cuidado na definição de usuário e senha do SYS, pois não tem como recuperar a senha
  + É uma aplicação pesada então exige uma configuração de máquina mínima.
  + Para gerenciar o banco de dados é necessário instalar o JDK 11 (X64).
  + Instalar o SQL Developer 21. Copiar o arquivo zipado e descompactar. Utilizar o “exe”. Aqui é onde é feita a conexão com o banco de dados.
    - Criar a conexão com o usuário e senha definido para o banco de dados

Exemplos de comando SQL: create table, select, insert

Compartilhamento do banco de dados, em um servidor disponível na internet:

* Servidores proprietários: infraestrutura própria e falta de elasticidade
* Servidores sob demanda : comprar ou alugar serviço em nuvem – a Oracle oferece este serviço gratuitamente, até um limite de tamanho.

Oracle Cloud Infrastructure (provisionar banco de dados ADW)

* Permite a utilização do serviço de nuvem, de armazenamento de dados
* Fazer o cadastro e informar um cartão de crédito
* Criar autonomous database, configurar banco “allways free”, para limitar o banco e não gerar cobrança, com usuário ADMIN e senha, tipo de rede (privada ou de qualquer local), email de contato.
* Após criado o banco de dados, a Oracle fornece os dados detalhados do banco de dados criado.
* Visualização de métricas para acompanhar a utilização do banco de dados.
* É fornecido o IP para acesso ao banco de dados (procurando o IP do banco de dados na juvem)
* Criar Wallet para acesso para não ter que ficar logando, baixar o arquivo do wallet (zipado).
* No Oracle instalado local, criar a conexão usando o wallet baixado, com o usuário e senha criado na nuvem.

Oracle DB Developer VM – Oracle VM VirtualBox

* Utilizado para ensino e aprendizagem, não é recomendado para a produção.
* Recomendado: Database App Development VM (é uma aplicação pesada)
* Importar Appliance Virtual (consome memória RAM)
* Após a instalação, ele abre um terminal Linux já com o Oracle instalado que permite execução de scripts em SQL

Para um DBA Oracle é importante saber Linux

**Mentoria #3: Melhorias práticas com API Rest e Restful**

- 30/06, 16:00, 2 horas

- Participantes: a definir (Digital Innovation One)

**Mentoria #4: Como ser o próximo contratado pela Unimed BH**

- 12/07, 16:00, 2 horas

- Participantes: Marco Antônio (Unimed-BH)

**Módulo I – Princípios de Desenvolvimento de Software**

**Introdução à programação e pensamento computacional**

Instrutora: Juliana Mascarenhas

**Pensamento computacional**

Apresentação

Apresentação da instrutora

Objetivo geral

Percurso

Aula 1 Pensamento computacional

Aula 2 Introdução à lógica da programação

Aula 3 Fundamentos de algoritmos

Aula 4 Linguagens de programação

Aula 5 Primeiro contato com a programação

Introdução

Aula 1 – pensamento computacional

O que é pensamento computacional:

* processo de pensamento envolvido na expressão de soluções em passos computacionais ou algoritmos
* Sistemático e eficiente: formulação e resolução de problemas
* Humanos e máquinas devem ser capazes de resolver
* Habilidade generalista: matemática, leitura, escrita

Quatro pilares do pensamento computacional

* Decomposição: dividir um problema complexo em subproblemas
* Reconhecimento de padrões
* Abstração: do particular para o genérico, modelagem do fenômeno
* Design de algoritmos: automatização da resolução de problemas, passo a passo da solução
  + Input -> operador -> output
  + Processo contínuo: definir solução, testar, aperfeiçoar / análise, refinamento e teste
  + Círculo virtuoso
  + Habilidades humanas x recursos computacionais:
    1. Abstração: resolução de problemas
    2. Automatização: expressão de soluções
    3. Análise: execução de solução e avaliação (refinamento)
  + Competências:
    1. Pensamento sistemático
    2. Colaboração dentro da equipe
    3. Criatividade e design
    4. Facilitador

Habilidades complementares

Raciocínio lógico:

* Forma de pensamento estruturado que permite determinar a resolução de um problema.
* Habilidade em treinamento
* Classificação:
  + Indução: a partir de um fenômeno observado gerando leis e teorias
  + Dedução: a partir de leis e teorias, se deduz previsões e explicações
  + Abdução : a partir de uma conclusão, se supõe uma premissa, muito utilizado em processos investigativos e diagnósticos
* Raciocínio lógico
  + Sintética
    - Abdução
    - Indução
  + Analítica
    - Dedução
* Aperfeiçoamento
  + Melhoramento
  + Aprimoramento
  + Refinamento
  + Ato de aperfeiçoar
    - Melhor uso de recursos
      * Encontrar solução eficiente
      * Otimizar processos
    - Melhores códigos e algoritmos
      * Simplificar linhas de código
      * Funções bem definidas
  + Determinar pontos de melhoria e refinamento

Pilares: Decomposição

Identificar o problema e dividi-lo em problemas menores, mais fáceis de resolver e de gerenciar

Estratégia

* Análise:
  + Quebrar o problema em problemas maiores
  + Estudar/explorar
  + Realizar exame detalhado
  + Decompor em elementos constituintes
* Síntese:
  + Combinar os elementos recompondo o problema original
  + Consiste em reunir elementos distintos em um único elemento
  + Processo de reconstrução
  + Fundir elementos de maneira coerente
* Ordem de execução
  + Sequencial: existe dependência e devem ser executadas em fila
  + Paralela: tarefas podem ser executadas concomitantemente e posteriormente agregadas

Decomposição

* Habilidade generalista em treinamento
* Existem maneiras distintas de resolver um problema
  + Custo x benefício
* Como decompor:
  + Identificar etapas ou coletar os dados
  + Agregar os dados
  + Funcionalidade
* Etapas:
  + Identificar as atividades
  + Determinar etapas, em paralelo ou sequencial
  + Executar cada etapa
  + Agregar as atividades, recompondo com coerência
* Exemplo: movimentos de um avatar
  + Ações
    - Padrão
      * Virar
        + Esquerda
        + Direita
      * Mover
        + Cima
        + Baixo
    - Movimento
      * Andar
      * Correr
        + Mover a perna
        + Impulsionar
      * Sentar
      * Levantar

Pilares: Padrões

Reconhecimento de padrões:

* Modelo base (referências)
* Estrutura invariante
* Repetição

Detecção de similaridades e diferenças

* Armazenamento de fotos: primeiro compressão e posterior armazenamento
* Reconhecimento de padrões é intuitivo na vida cotidiana

Determinando padrões:

* Generalizar para obter resolução de problemas diferentes
* Grau de similaridade
* Grupos conhecidos x objeto desconhecido
* Reconhecimento de padrão é feito por comparação

Padrões – como simular o comportamento humano

* Representação de atributos
* Aprendizado – conceito associado ao objeto
* Armazenamento de dados
* Regras de decisão

Abordagem:

* Extração de características
* Classificação de dados
* Diferentes métodos de reconhecimento de padrão

Aplicações:

* Classificação de dados
* Reconhecimento de imagem
* Reconhecimento de fala
* Análise de cenas
* Classificação de documentos

Pilares: Abstração

Conceitos:

* Generalização: tornar mais amplo, extenso, geral – permite definir uma classe geral
* Abstração: extrapolar um objeto da realidade

Como classificar os dados

* Características: identificação
* Pontos essenciais: ignorar características que não são relevantes
* Generalizar x detalhar: definir que características ignorar ou detalhar

Exemplos de abstrações:

* Árvores
* Listas
* Grafos
* Máquinas de estados finitos
* Linguagens de programação
* Arquiteturas de TI

Pilares: Algoritmos

O computador não opera sozinho: precisa de instruções detalhadas do que e de como processar.

Instruções – o entendimento deve ser possível por humanos e máquinas:

* O que precisa ser feito
* Qual a ordem de execução

Desenvolvimento de programas

* Análise: estudo e definição de entradas e saídas
* Algoritmo: descreve o problema por meio de ferramentas
* Codificação: utiliza linguagem de programação

Algoritmos

* Sequência de passos com objetivo definido
* Execução de tarefas específicas
* Conjunto de operações que resultam em uma sucessão finita de ações

Como construir um algoritmo

* Compreensão do problema
* Definição de dados de entrada
* Definir processamento
* Definir dados de saída
* Utilizar método de construção
* Teste e diagnóstico

Construção algoritmo

* Narrativa: linguagem natural
* Fluxograma: utilização de símbolos pré-definidos
* Pseudocódigo: portugol

Como aprimorar essa habilidade

* Buscando explicações para cada decisão em seu exercícios de pensamento computacional

Estudo de caso: Perdido

Caso: existe uma pessoa perdida na floresta

Objetivo: como maximizar a sua sobrevivência

Método: decomposição, reconhecimento de padrões e abstração

Detalhamento:

* Água:
  + Chuva
  + Nascente
    - Fogo
* Comida:
  + Coletar
  + Caçar
    - Fogo
* Abrigo
  + Localização
    - Mapa
  + Proteção
    - Fogo
    - Lança
  + Quente e seco
    - Fogo

Estudo de caso: Soma de um intervalo

Caso: soma de um número de 1 a 200

Objetivo: Fazer o cálculo de modo simples, utilizando um padrão (função)

Método: decomposição, reconhecimento de padrões e abstração

A definição do padrão cria uma função matemática

Estudo de caso: Adivinhe um número

Caso: Adivinhe um número

Objetivo: Determinar o número escolhido por uma pessoa dentro de um intervalo, a partir de perguntas com respostas do tipo sim ou não. Não existe limite de perguntas,

Método: decomposição, reconhecimento de padrões e abstração

A definição do padrão de perguntas busca limites superior e inferior, como por exemplo “o número é maior do que 50?”

**Introdução à lógica de programação**

O que é lógica

Conceito: forma de raciocínio, ordenada, que segue convenções, de forma coesa.

Definição: organização e planejamento de instruções, assertivas, em um algoritmo, a fim de viabilizar a implantação de um programa.

Pressupõem a previsão de comportamentos.

Lógica ensina a pensar.

Técnicas de lógica de programação

Técnicas de lógica:

* Técnica linear:
  + Modelo tradicional
  + Execução sequenciada
  + Unidimensional
  + Recursos limitados
* Técnica estruturada (hierarquia)
  + Escrita
  + Entendimento
  + Validação
  + Manutenção
* Técnica modular
  + Partes independentes, controladas por um conjunto de regras
  + Modelo:
    - Dados de entrada
    - Processo de transformação
    - Dados de saída
  + Objetivos (metas):
    - Simplificação
    - Decomposição de problemas
    - Verificação por módulo