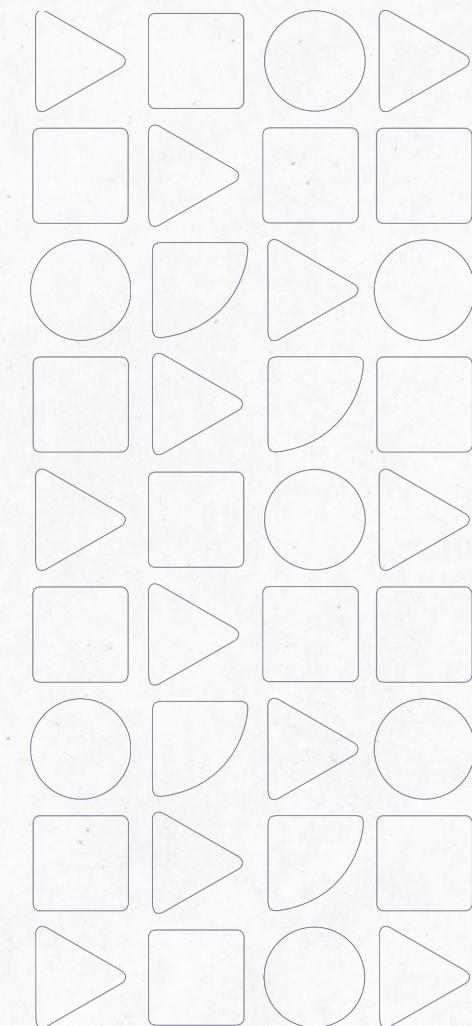


Manipulando Dados

Disciplina: Linguagem de Programação



Conteúdos:

Manipulando dados.

Habilidade(s):

- Compreender a manipulação de dados e o uso de funções e métodos;
- Aprender a coletar, armazenar e exibir dados inseridos pelos usuários, além de realizar operações com esses dados.

Bloco 1

Entendendo o que são *arrays* (vetores).

► □ ○ ▶ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □

Ao ouvir o termo “dado”, o que vem à sua mente?



Resuma em uma só
palavra!

Os dados compõem os elementos mais essenciais da linguagem de programação.

Com eles, é possível armazenar, manipular e representar informações de diversas formas.



Por que manipular dados é tão importante?

Porque isso mantém os seus dados organizados e úteis! Assim, várias ações são possíveis.

Ordenar dados: imagine, por exemplo, que você tem uma pilha de cartas e deseja organiza-las em ordem alfabética.

Filtrar: se você tem uma lista de filmes, pode filtrar apenas os de comédia, encontrando, assim, os mais engraçados.

Validar: ao preencher um formulário *on-line*, os seus dados são validados, garantindo que as informações corretas foram coletadas.

Transformar: se você tem uma lista de números, pode transformá-los em outros resultados. Fazer o cálculo da média deles é um exemplo disso.



Protegendo os seus dados

Além disso, é importante manter a **integridade**, se tratando do **armazenamento** dos seus dados.

Confira os motivos!

Recuperação de dados e criação de relatórios

É essencial para armazenar e analisar os dados coletados.

Ex.: em uma loja, um caixa recupera dados de todas as vendas do dia, criando um relatório que ajuda a entender quais foram os produtos mais vendidos.

Limpeza de dados

Significa tornar as informações mais organizadas e precisas.

Ex.: quando uma tarefa é marcada como concluída em um aplicativo de listas do que precisa ser feito, ela é removida da listagem, mantendo-a limpa e organizada.

Backup e restauração

Implica fazer uma cópia de segurança de suas informações e recuperá-las quando necessário.

Ex.: você faz o *backup* das fotos de uma viagem no computador ou na nuvem. Se o seu telefone quebrar, você pode restaurar essas imagens em um novo telefone.

Segurança de dados

Significa mantê-los em segurança.

Ex.: um *site* pede uma senha para acessar a sua conta, protegendo as suas informações pessoais de tentativas não autorizadas de acesso.

Integração de dados

Refere-se ao processo de combinar informações de diferentes fontes ou sistemas para criar uma visão unificada e coesa dos dados.

Um exemplo disso é a famosa **planilha** de dados no Excel, um banco de dados e dados de estoque que, juntos, podem integrar-se.



Mineração de dados

É o processo de explorar grandes conjuntos de dados para descobrir padrões, tendências e informações ocultas.

Um *site* de comércio eletrônico deseja entender o comportamento de compra de seus clientes. Usando técnicas de mineração de dados, eles podem descobrir que os clientes tendem a comprar produtos relacionados, como um telefone e uma capa protetora, ao mesmo tempo.



Manipulação de datas

Envolve trabalhar com informações de datas e horários,
como dia, mês, ano, hora, minuto e segundo.





Você sabia?

Em JavaScript, a data é um objeto que nos mostra a data atual em milissegundos percorridos desde 1º de janeiro de 1970.

Conceitos-chave sobre a manipulação de datas

Representação de datas e horários

Datas e horários são representados como objetos que contêm informações específicas, como dia, mês, ano, hora, minuto e segundo. Essas informações são armazenadas em uma estrutura de dados que permite a manipulação.

Formatos de data e hora

As datas podem ser representadas em diversos formatos, como "dd/mm/aaaa", "aaaa-mm-dd" ou "mm/dd/aaaa". Além disso, os horários podem ser representados em formatos de 12 ou 24 horas, com ou sem fuso horário.

Operações básicas

A manipulação de datas também envolve operações básicas, como adição ou subtração. Elas são utilizadas nos dias, meses e anos, bem como na formatação de datas em um formato específico e cálculos de diferenças entre duas datas.



Observando na prática

Imagine que você está criando um *site* que faz a contagem regressiva para o Ano Novo. O seu objetivo, então, é mostrar aos usuários quanto tempo falta para a meia-noite e a chegada do próximo ano. Para isso, é preciso manipular datas.

Para obter a data atual, você poderia utilizar `const dataAtual = new Date()`

Para definir a data do Ano Novo do ano em questão, seria possível usar `const anoNovo = new Date(dataAtual.getFullYear() + 1, 0, 1)`

Para calcular a diferença entre as datas, seria necessário utilizar `const diferenca = anoNovo - dataAtual`

Analisando o exemplo anterior

Com esse código, você visualiza a data atual, podendo ver quanto tempo falta para o próximo ano. O programa também mostrará uma mensagem divertida, informando quantos dias, horas, minutos e segundos faltam para esse momento específico.



Bloco 2

Explorando os fusos horários, análise de séries
temporais, validação e formatação, tarefas
recorrentes e *callback*.

Adedonha

Vamos recapitular o que vimos até aqui?



Para realizar essa dinâmica, o(a) professor(a) precisará sortear **uma letra** do alfabeto. A partir do sorteio, os alunos terão que dizer os primeiros sistemas, *sites* ou programas que vierem à mente. É importante lembrar que os nomes precisam começar com a letra sorteada e, além disso, devem estar relacionados à manipulação de datas.

Tratamento de fusos horários

Em contexto global, é importante considerar os fusos horários ao manipular datas.

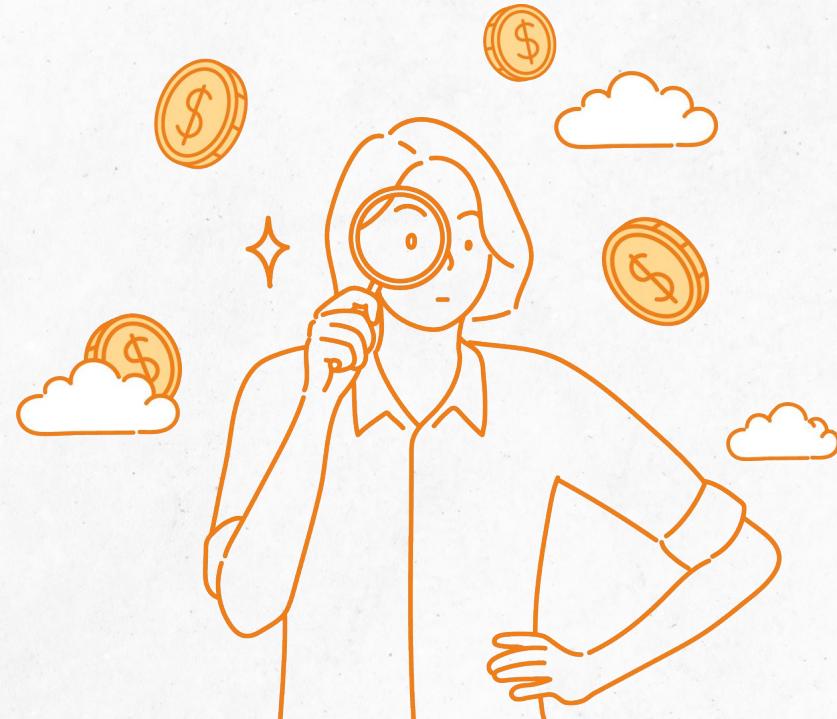
Plataformas de reservas de viagens, como Expedia e Booking.com, ajustam automaticamente os horários de chegada e partida de voos e hotéis, considerando os fusos horários relevantes. Isso ajuda os viajantes a planejarem suas viagens de acordo com os fusos horários locais.



Análise de séries temporais

É o estudo de dados que variam ao longo do tempo. Isso envolve a análise de padrões, tendências e flutuações em dados temporais para fazer previsões ou extrair *insights*.

Um exemplo disso é a análise do histórico de preços de ações, prevendo o seu comportamento futuro.



Validação e formatação

Inclui a validação de datas para garantir que estejam em um formato correto. Aqui, também é necessário formatar as datas para exibi-las em formato legível aos usuários.

Um exemplo é validar se uma data inserida por um usuário está no formato correto (por exemplo, "dd/mm/aaaa") e, em seguida, formatá-la para "**dd de mês de aaaa**".



Tarefas recorrentes

Muitas aplicações lidam com tarefas que ocorrem em datas e horários específicos, como agendamento de eventos, lembretes e notificações. A manipulação de datas é essencial para o gerenciamento dessas tarefas, garantindo que ocorram no momento correto.

Agendar um alarme para tocar em determinada hora do dia ou enviar um *e-mail* de lembrete em uma data específica.



Callback

É uma função passada como argumento para outra função. A função de *callback* será invocada dentro da função externa em um momento específico, geralmente para completar uma ação ou executar uma rotina específica.

Em JavaScript, ao usar **setTimeout** para agendar uma ação que deve ser executada após um certo período de tempo, você pode passar uma função de *callback* para especificar o que deve acontecer quando o tempo expirar.



Bloco 3

Vamos praticar?



Dê *play* no conhecimento!



Antes, vamos descobrir como elaborar um **mapa mental!**

Vamos explorar?

Nesta atividade, vocês devem explorar sistemas ou aplicativos do mundo real que envolvem aspectos da manipulação de datas, como análise de séries temporais, tratamento de fusos horários e tarefas recorrentes em datas. Depois, devem pesquisar e criar mapas mentais em uma cartolina.

Dividam-se em cinco equipes e sintam-se livres para escolher qualquer um dos temas a seguir:

análise de séries temporais

tarefas recorrentes em datas

tratamento de fusos horários

Bloco 4

Conhecendo as bibliotecas.

Para facilitar a manipulação de datas em programação, muitas linguagens de programação oferecem bibliotecas ou módulos sobre esse tema.



Acessando uma biblioteca

Na plataforma **code.org**, podemos acessar a biblioteca de dados através do botão localizado no canto superior esquerdo da tela.

The screenshot shows the code.org interface with the 'Dados' tab selected in the top navigation bar. On the left, there's a preview area for a screen named 'screen1' with an 'Executar' button at the bottom. The main area is divided into two sections: 'Biblioteca de dados' and 'Navegador de dados'. The 'Biblioteca de dados' section lists various categories with their respective table counts:

Categoria	Quantidade de tabelas
Animals	6 tables
Art	3 tables
Economics	4 tables
Geography	12 tables
Language and Literature	5 tables
Culture & Entertainment	14 tables
Miscellaneous	10 tables
NASA	6 tables
Politics	7 tables
Science	15 tables
Sports	14 tables
Spotify Charts	4 tables
Transportation	4 tables

The 'Navegador de dados' section contains fields for creating new tables: 'Nome da tabela' and 'Adicionar'. Below these fields is a table header with columns 'Nome da tabela' and 'Ações'.

Passo 1: formate a planilha

Em um editor de planilha de sua escolha, certifique-se de que os dados estejam formatados.

Como exemplo, vamos utilizar uma tabela com nomes, idades e as comidas favoritas correspondentes de cada pessoa.

Nome	Idade	Comida favorita
Fernanda	18	Macarrão
Tiago	20	Pizza
Roberto	21	Cachorro-quente

Passo 2: salve a planilha

Salve o arquivo como **.csv** e baixe-o em seu computador.



Um arquivo separado por vírgula, ou arquivo .csv, é um formato simples de salvamento de arquivos. Além disso, a maioria dos editores de planilhas suporta esse formato.



Passo 3: adicionando o arquivo

Importe o arquivo para o armazenamento de dados da tabela do App Lab.

Clique no botão “visualizar dados” no App Lab enquanto visualiza seu código no “modo de código”.

Clique no nome de uma tabela existente ou adicione uma nova tabela ao seu aplicativo.

Clique no botão “importar” e selecione o arquivo .csv da etapa 2.

Clique em “substituir e importar” para que os dados da planilha substituam os dados atuais da tabela.

Passo 4: revisando os dados

Se os seus dados forem importados com sucesso, revise a tabela para ter certeza de que tudo está correto.

Para criar, ler, atualizar e excluir registros de tabela do seu aplicativo, use as seguintes funções do App Lab Data.

Para criar uma linha em uma tabela: `createRecord()`

Para ler uma ou várias linhas em uma tabela: `readRecords()`

Para atualizar uma linha existente em uma tabela:
`updateRecord()`

Para excluir uma linha existente de uma tabela: `deleteRecord()`





Navegue pelo conhecimento!



Hora de visualizar tudo isso na prática!

Bloco 5

Coletando dados.

Coletando dados

No exemplo do bloco anterior, é possível visualizar três espaços para os dados: **nome**, **idade** e **comida favorita**.

Para coletar dados, alguns passos são realizados.

Dentro da função de clique, um objeto chamado favFoodData é criado. Esse objeto será usado para armazenar os dados inseridos pelo usuário.

Os valores dos campos de entrada de texto são obtidos usando as funções `getText()` e `getNumber()` e são atribuídos às propriedades do objeto favFoodData.

A função `createRecord()` é usada para enviar os dados do objeto favFoodData para uma tabela chamada “`fav_foods`”.

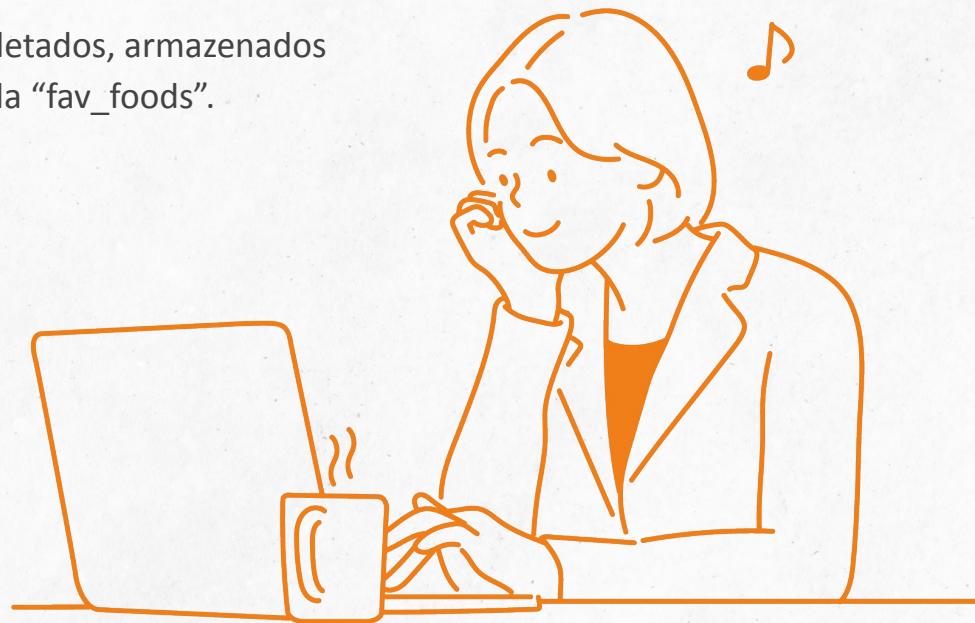
A função `createRecord()` aceita uma função de retorno de chamada (callback) que será executada após os dados serem enviados com sucesso para a tabela.

Dentro da função de retorno de chamada, as informações do registro recém-criado são exibidas no console do navegador, incluindo o `id` do registro, o nome, a idade e a comida favorita.

Interface

Esse código cria uma interface na qual os usuários podem inserir nome, idade e comida favorita.

Quando o botão “done” é clicado, os dados são coletados, armazenados em um objeto e enviados para uma tabela chamada “fav_foods”.





Navegue pelo conhecimento!



Hora de visualizar tudo isso na prática!

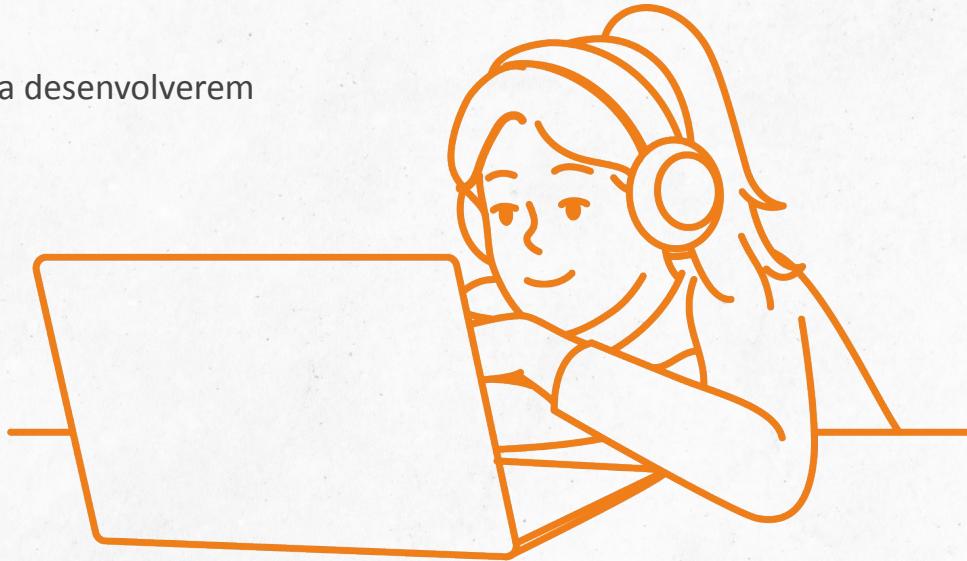
Bloco 6

Hora de praticar!

Meu filme preferido

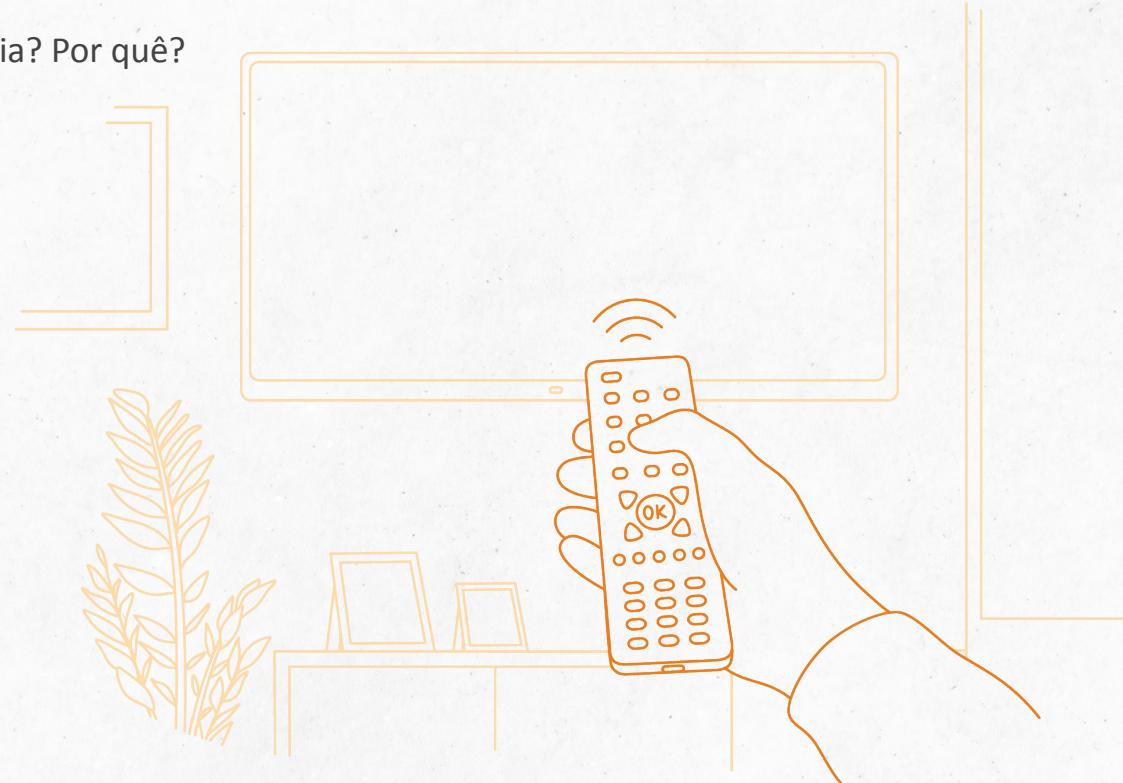
Dividam-se em cinco grupos e acessem um *notebook* ou computador disponível. A partir do que foi apresentado, coletam os nomes, os gêneros de filmes favoritos e os filmes preferidos de cada um, reunindo-os em uma tabela.

Em seguida, acessem a plataforma do code.org para desenvolverem um sistema de coleta e armazenamento de dados.



Fechamento

Se a aula de hoje fosse um filme, qual seria? Por quê?



Referências Bibliográficas

PROZ EDUCAÇÃO. *Apostila de Desenvolvimento para Dispositivos Móveis I.* 2023.

CNBLOGS. *Travessia de DOM*. 10 fev. 2017. Disponível em:

<https://www.cnblogs.com/xiaohuochai/p/6387570.html>. Acesso: 11 out. 2023.