### **EDUARDO RODRIGUES DA SILVA**

### **1)**

### **a) Existe possibilidade de manipular Strings associadas a Arrays? Como isso funciona?**

R: Sim, no JS as strings podem ser tratadas de forma parecida aos arrays, pois as strings podem ser acessadas por índice. Também é possível transformar uma string em array e também converter esse array de volta em string. Isso é bom para fazer modificações como substituição e separações.

**b) Diferença entre Math.floor(), Math.ceil() e Math.round():**

R: O Math.floor() arredonda um número **para baixo**, sempre para o número inteiro menor.  
Já no Math.ceil**()** arredonda um número **para cima**, sempre para o número inteiro maior.  
No Math.round() arredonda para o **inteiro mais próximo**, levando em conta a parte decimal, se for 0.5 ou mais, arredonda para cima; se for menor que 0.5, já vai para baixo.

**c) Como objetos do tipo data são criados em JS?**

R: Objetos do tipo data em JavaScript são criados usando o construtor Date. Ao usar esse construtor sem parâmetros, ele cria um objeto com a data e hora atuais. Esse objeto pode ser usado para obter o dia, mês, ano, hora, minuto e segundo do momento em que foi criado.

### **2)**

### **a) O que é uma expressão regular em JS? Qual seu propósito?**

R: Uma expressão regular em JS é um padrão usado para identificar, buscar ou validar partes de uma string. Tendo o propósito de atuar em tarefas como verificar formatos de dados como e-mails e senhas e extrair partes específicas de textos.

**b) Para que serve o método .test?**

R: O método .test() é usado com expressões regulares para verificar se uma determinada string segue o padrão definido. Ele retorna true se a string corresponde ao padrão e false caso contrário, sendo utilizado em validação de entradas dos usuários.

### **3)**

**a) O que é e como funciona a Programação Assíncrona em JS?**

R: A programação assíncrona permite que operações de requisições a servidores ou leitura de arquivos, sejam realizadas em segundo plano, sem parar o restante do código, melhorando o desempenho do usuário.

**b) Como o conceito de Single-Threaded afeta o fluxo de tarefas no JS?**

R: JScript é um single-threaded, que só executa uma instrução por vez, podendo causar travamentos caso tarefas demoradas forem rodadas de forma síncrona, por isso, o uso de tarefas assíncronas é necessário para manter a fluidez do código.

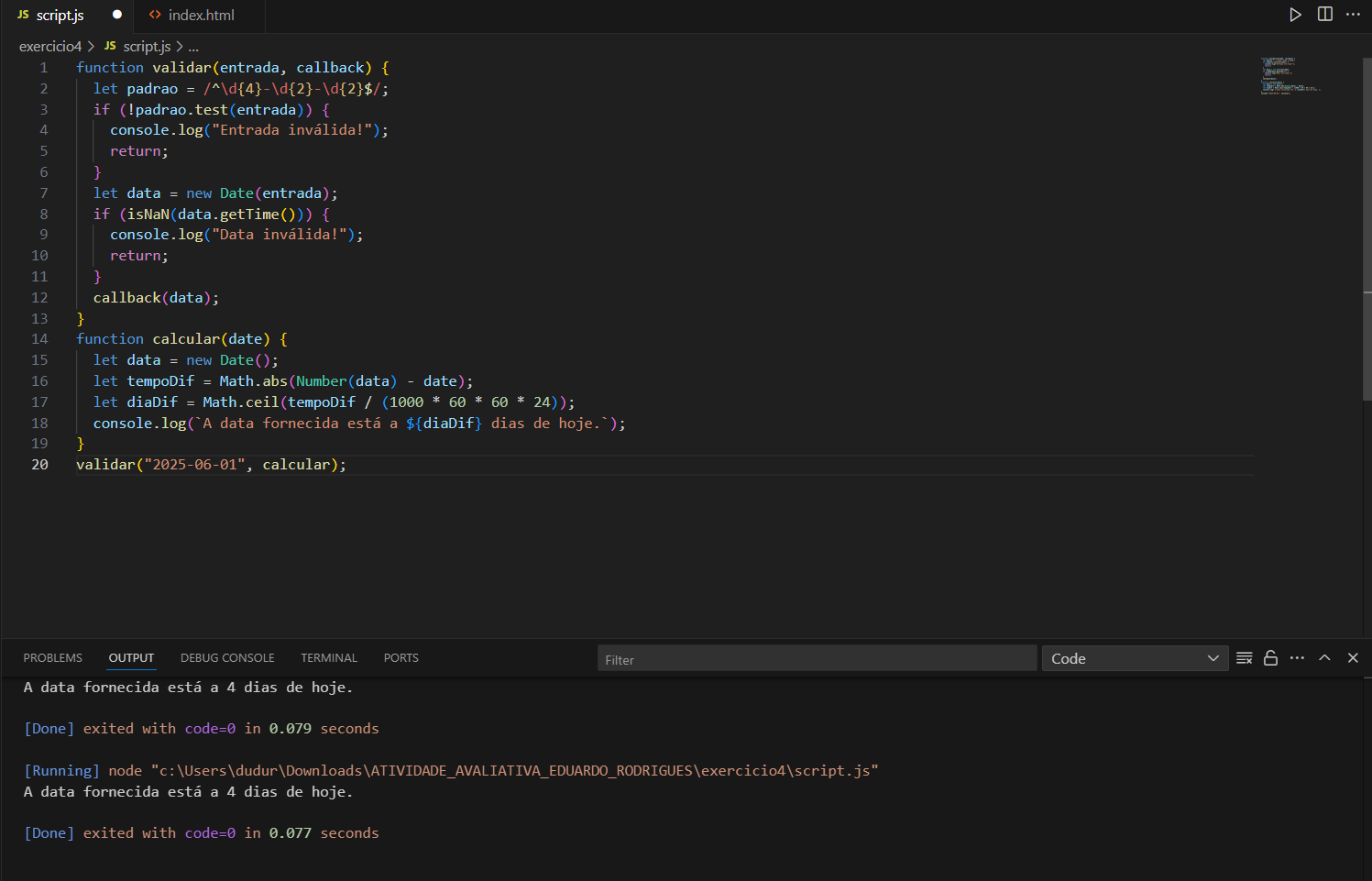
**c) Qual a diferença entre promises e async/await? Existe alguma relação com callbacks?**

R: Os Promises são uma forma de lidar com tarefas assíncronas, oferecendo um jeito mais organizado do que callbacks.  
O async/await é uma forma mais moderna e legível de lidar com Promises.

Os callbacks, Promises e async/await servem para lidar com ações que não acontecem imediatamente, mas async/await facilita a leitura e escrita do código em comparação com os outros.

**d) No contexto de JavaScript, o que é event loop?**

R: O event loop é o mecanismo que controla a execução de tarefas assíncronas. Ele monitora a fila de tarefas e a pilha de execução, decidindo a ordem correta para executar cada função. Isso garante que, mesmo com código assíncrono, tudo aconteça no momento apropriado sem travar o programa.

4 - O resultado foi “A data fornecida está a 4 dias de hoje”.  


5 - O JS faz primeiro as coisas que estão normais no código, tipo os console.log, depois ele faz a função async function(), e por último ele faz o SetTimeOut.

