

# Lista de exercícios Promises

## 1. Simulação de Cadastro de Usuário:

- Crie uma função `cadastrarUsuario(nome, email)` que simula o cadastro de um usuário.
- A função deve receber o nome e o email do usuário.
- Simule uma verificação assíncrona de banco de dados para verificar se o email já está em uso.
- Se o email estiver disponível, resolva a Promise com uma mensagem de sucesso.
- Se o email já estiver em uso, rejeite a Promise com uma mensagem de erro.

## 2. Gerador de Números Aleatórios:

- Crie uma função `gerarNumeroAleatorio(min, max)` que gera um número aleatório dentro de um intervalo.
- A função deve receber os valores mínimo e máximo do intervalo como parâmetros.
- Use `Math.random()` para gerar o número aleatório.
- Resolva a Promise com o número gerado.

## 3. Consulta de Dados de Usuário:

- Crie uma função `consultarUsuario(id)` que simula a consulta de dados de um usuário em um banco de dados.
- A função deve receber o ID do usuário como parâmetro.
- Simule uma consulta assíncrona ao banco de dados.
- Resolva a Promise com os dados do usuário (por exemplo, nome, email, idade) se o usuário for encontrado.
- Se o usuário não for encontrado, rejeite a Promise com uma mensagem de erro.

## 4. Autenticação de Usuário:

- Crie uma função `autenticarUsuario(email, senha)` que simula o processo de autenticação de um usuário.
- A função deve receber o email e a senha do usuário como parâmetros.
- Simule uma verificação assíncrona de banco de dados para verificar se as credenciais são válidas.
- Resolva a Promise com um token de autenticação se as credenciais forem válidas.
- Se as credenciais forem inválidas, rejeite a Promise com uma mensagem de erro.

## 5. Consulta de Previsão do Tempo:

- Crie uma função `consultarPrevisaoTempo(cidade)` que simula a consulta da previsão do tempo de uma cidade.
- A função deve receber o nome da cidade como parâmetro.

- Simule uma consulta assíncrona a uma API de previsão do tempo.
- Resolva a Promise com os dados da previsão do tempo se a consulta for bem-sucedida.
- Se a consulta falhar, rejeite a Promise com uma mensagem de erro.

#### 6. Envio de E-mail Assíncrono:

- Crie uma função `enviarEmail(destinatario, assunto, corpo)` que simula o envio de um e-mail.
- A função deve receber o destinatário, o assunto e o corpo do e-mail como parâmetros.
- Simule o envio assíncrono do e-mail.
- Resolva a Promise com uma confirmação de que o e-mail foi enviado com sucesso.
- Se ocorrer um erro durante o envio, rejeite a Promise com uma mensagem de erro. (o erro deve acontecer 20% das vezes)
  - Caso o envio falhe 1 vez, tente enviar novamente