**01) Teste de Lógica**:

- Entrada de dados: S1, S2, S3 e S4.

- Saída esperada: B1 e V1 ou se os sensores estão apresentando problema (\*).

- S1, S2, S3, S4, B1 e V1 serão dados booleanos.

- Front-End – Deverá ser feito em WPF (Windows Presentation Foundation).

- Back-End - C# / .NET framework 4.5.

- Ao inserir os valores de S1, S2, S3 e S4 o sistema deverá retornar se ativa ou desativa B1 e V1 ou se os sensores estão apresentando problema.

\*Um exemplo de problema com os sensores é: S1 = 1(com água) e S2 = 0(sem água), é impossível um sensor no topo da caixa d’agua estar ativo e o do fundo estar inativo, isso indica erro no sensor.

Arquivos de Apoio:

* 01) Teste de Lógica.pdf
* 01) Teste de Lógica - Resolvido.xlsm: Exemplo de solução, a parte visual não precisa ser exatamente a mesma e não precisa ter animação na alteração dos sensores.

**02) Teste de Lógica de Programação, Front-End Web e classes**:

- Elabore um sistema web, onde na tela inicial possa ser selecionado um usuário (dentre 3). Ao selecionar o usuário, deverá ser aberta uma tela com 3 grids, com cada tipo de pagamento aparecendo em um dos grids (dependendo do usuário). Ao clicar na linha do grid, deverá abrir um modal com as informações da linha e o botão excluir, que ao clicar nele exclui a linha selecionada (Pode ser no front-end apenas).

- Back-End (C# / .NET framework 4.5) – Nele deverá estar a lógica de o que é exibido em cada grid, se baseando no usuário selecionado. No início, todos os dados deverão estar em uma lista de pagamentos, que será dividida em 3 (uma para cada grid). Um grande diferencial é a criação de uma web api, deixando o back-end completamente independente. A api deverá receber requisições contendo o nome ou id do usuário e retornar os pagamentos identificados de cada grid.

- Front-End: html, css, javascript ou bibliotecas js.

- Fonte de Dados – Os dados de pagamento deverão vir obrigatoriamente de um arquivo em excel(.xlsx) e os usuários poderão ser criados direto no código. Não utilizar banco de dados.

- Deverão ser utilizados classes para definir os atributos de ‘Usuário’ e ‘Pagamento’

Arquivos de Apoio:

* 02) Teste de Lógica de Programação, Front End Dinâmico e classes.xlsx

**Parâmetros Gerais:**

- As variáveis devem ser facilmente identificáveis.

- Será levado em consideração a organização e legibilidade do código.

- Marcar claramente códigos escritos e códigos autogerados pelo framework.

- Comentar as funções criadas com o objetivo e a saída esperada, usar preferencialmente comentários XML para o back-end.

- As exceções que não foram citadas não precisam necessariamente de tratamento.