**Nome: Rodrigo Cinelli**
**Tarefa: Teste Caixa Branca “Cantinho da Gula”**


# Teste Caixa Branca Software “Cantinho da Gula”

## Arquivo Cardápio.html

1. **Na tabela de bebidas alcoólicas e não alcoólicas, bem como na tabela de acompanhamentos, está sendo utilizado uma linha de tabela <tr> adicional dentro de outra linha de tabela <tr>. Essa estrutura não é correta e pode causar problemas de renderização.**

```
<tr>
   <td>Heineken</td>
   <td>R$13,00</td>
   <tr style= "width: 300px; border: 1px solid #a98d0e;"></tr> <!-- Linha de tabela incorreta -->
</tr>
```

**Para corrigir esse problema, você pode remover a linha de tabela incorreta e aplicar o estilo diretamente na linha de tabela <tr> correta. Por exemplo, você pode alterar o código acima para o seguinte:**

```
<tr style="width: 300px; border: 1px solid #a98d0e;">
   <td>Heineken</td>
   <td>R$13,00</td>
</tr>
```

**Aqui está o código corrigido com as alterações aplicadas às tabelas de bebidas alcoólicas, bebidas não alcoólicas e acompanhamentos:**

```
<!-- Tabela de bebidas alcoólicas -->
<table id="Alcóolicos">
   <tr>
```

```
      <th>Alcoólicos</th>
      <th>Preço</th>
    </tr>
    <tr style="width: 300px; border: 1px solid #a98d0e;">
      <td>Heineken</td>
      <td>R$13,00</td>
    </tr>
    <!-- ... outras linhas de bebidas alcoólicas corrigidas ... -->
</table>

<!-- Tabela de bebidas não alcoólicas -->
<table id="bebidas">
    <tr>
      <th>Bebidas não Alcóolicas</th>
      <th>Preço</th>
    </tr>
    <tr style="width: 300px; border: 1px solid #a98d0e;">
      <td>Coca-Cola</td>
      <td>R$6,90</td>
    </tr>
    <!-- ... outras linhas de bebidas não alcoólicas corrigidas ... -->
</table>

<!-- Tabela de acompanhamentos -->
<center>
    <table id="Acompanhamentos">
      <tr>
        <th>Acompanhamentos</th>
        <th>Preço</th>
      </tr>
      <tr style="width: 300px; border: 1px solid #a98d0e;">
        <td>Salada Simples</td>
        <td>R$21,90</td>
      </tr>
      <!-- ... outras linhas de acompanhamentos corrigidas ... -->
    </table>
</center>
```

As linhas de tabela incorretas foram removidas e o estilo foi aplicado diretamente nas linhas de tabela `<tr>` corretas. Faça essas alterações no seu código original e o problema de renderização deve ser resolvido.

## Arquivo reservas.php

2. **O código reservas.php está funcionando corretamente. Ele faz o seguinte:**

a) **Inclui o arquivo de configuração para estabelecer uma conexão com o banco de dados.**

b) **Verifica se há algum erro de conexão.**

c) **Verifica se o método de requisição é POST.**

d) **Prepara uma instrução SQL para inserir dados na tabela de reservas.**

e) **Associa os parâmetros da instrução SQL com as variáveis que armazenam os dados do formulário.**

f) **Associa os parâmetros da instrução SQL com as variáveis que armazenam os dados do formulário.**

g) **Atribui os valores dos campos do formulário às variáveis correspondentes.**

h) **Executa a instrução SQL e verifica se a operação foi bem-sucedida.**

- **Se for bem-sucedida, exibe uma mensagem de sucesso e redireciona o usuário para a página index.html.**

- **Se não for bem-sucedida, exibe uma mensagem de erro e redireciona o usuário para a página index.html.**

i) **Fecha a instrução e a conexão com o banco de dados.**

## Arquivo buscar_reserva.php

3. **O código buscar_reserva.php está funcionando corretamente. Ele faz o seguinte:**

a) **Inclui o arquivo de configuração para estabelecer uma conexão com o banco de dados.**

b) **Verifica se há algum erro de conexão.**

c) **Recupera os valores de e-mail e senha enviados pelo formulário.**

d) **Consulta as reservas do usuário com base no e-mail e senha informados.**

e) **Verifica se há resultados.**

- **Se não houver resultados, exibe um alerta informando que o e-mail ou senha são inválidos e retorna à página anterior.**
- **Se houver resultados, exibe as reservas encontradas em um formulário de edição e um formulário de exclusão.**

f) **Fecha a conexão com o banco de dados.**

## Arquivo cancelar.php

4. **O código cancelar.php está funcionando corretamente. Ele faz o seguinte:**

a) **Configura a conexão com o banco de dados.**

b) **Cria uma conexão com o banco de dados usando a biblioteca MySQLi.**

c) **Verifica se há algum erro de conexão.**

d) **Verifica se o formulário foi enviado.**

e) **Prepara e executa uma instrução SQL para excluir a reserva com base no e-mail fornecido.**

f) **Verifica se a execução foi bem-sucedida e exibe uma mensagem adequada.**

g) **Fecha a instrução preparada e a conexão com o banco de dados.**

**No entanto, existe uma consideração a ser feita:**

- **O arquivo não inclui o arquivo config.php como nos outros exemplos. Isso não é um problema, mas é uma boa prática manter a consistência no código. Para isso, é importante substituir as linhas de configuração do banco de dados (1-4) pela inclusão do arquivo config.php:**

  **include_once('config.php');**

## Arquivo config.php

5. **O arquivo config.php é um arquivo de configuração simples e correto. Ele realiza as seguintes ações:**

a) **Define as variáveis de configuração do banco de dados: servidor, nome de usuário, senha e nome do banco de dados.**

b) **Cria uma conexão com o banco de dados usando a biblioteca MySQLi e armazena a conexão na variável $conn.**

**Este arquivo é útil porque permite que o usuário mantenha as informações de configuração do banco de dados em um único local. Isso facilita a manutenção e evita a duplicação de código em vários arquivos PHP que interagem com o banco de dados.**

**A configuração atual  está correta e funcionará bem com os outros arquivos. Caso precise modificar as credenciais do banco de dados no futuro, basta atualizar este arquivo.**

## Arquivo editar.php

6. **O arquivo editar.php é um script PHP que lida com a busca de uma reserva no banco de dados usando o endereço de e-mail fornecido no formulário. Aqui está uma análise detalhada do código:**

a) **Ele define as variáveis de configuração do banco de dados e cria uma conexão com o banco de dados usando a biblioteca MySQLi.**

b) **Verifica se a conexão com o banco de dados foi bem-sucedida, caso contrário, exibe uma mensagem de erro e encerra o script.**

c) **Verifica se o formulário foi enviado usando o método POST.**

d) **Prepara e executa a instrução SQL para selecionar a reserva com base no endereço de e-mail fornecido no formulário.**

e) **Verifica se a instrução SQL foi executada com sucesso.**

- **Se a reserva for encontrada, o usuário é redirecionado para a página de edição da reserva (editar-reserva-form.php), com o ID da reserva passado como um parâmetro GET.**

- **Se a reserva não for encontrada, exibe uma mensagem de erro informando que a reserva não foi encontrada.**

- **Caso ocorra um erro ao buscar a reserva, exibe uma mensagem de erro relacionada ao problema.**

f) Fecha o objeto da instrução e a conexão com o banco de dados.

O código  está correto e funcionará conforme o esperado, desde que haja um arquivo chamado editar-reserva-form.php que lida com a exibição e edição da reserva. Se não tiver um arquivo chamado editar-reserva-form.php, crie um e implemente a funcionalidade de edição da reserva.

Além disso, recomendo substituir as primeiras linhas de configuração do banco de dados e criação da conexão com o banco de dados pelo arquivo config.php que já existe, assim estará utilizando uma única fonte de configuração para todos os arquivos que interagem com o banco de dados.

Para fazer isso, substitua as linhas 1 a 9 por:

```
include_once('config.php');
```

Isso fará com que o arquivo config.php seja incluído e a conexão com o banco de dados seja estabelecida.

## Arquivo process-newsletter.php

7. O código process-newsletter.php é um script PHP que lida com o envio de um e-mail de boas-vindas após um usuário se inscrever na newsletter. O script recebe os dados do formulário (nome e e-mail) e envia um e-mail para o endereço fornecido. Aqui está uma análise detalhada do código:

a) Verifica se o formulário foi enviado usando o método POST.

b) Recupera os valores de "name" e "email" do formulário e armazena-os nas variáveis $name e $email, respectivamente.

c) Define as variáveis $to, $subject e $message para o envio do e-mail.

- $to: armazena o endereço de e-mail do usuário.

- **$subject: define o assunto do e-mail como "Bem-vindo à nossa newsletter!".**

- **$message: define a mensagem do e-mail agradecendo ao usuário por se inscrever na newsletter.**

d) **Define os cabeçalhos do e-mail para incluir informações como o endereço "From" (De), "Reply-To" (Responder para) e o tipo de conteúdo (Content-Type).**

e) **Usa a função mail() do PHP para enviar o e-mail com as informações fornecidas.**

- **Se o e-mail for enviado com sucesso, exibe a mensagem "E-mail enviado com sucesso!".**

- **Se houver algum erro ao enviar o e-mail, exibe a mensagem "Erro ao enviar e-mail!".**

**O código está correto e funcionará conforme o esperado, mas precisa substituir seuemail@seudominio.com nos cabeçalhos pelo endereço de e-mail real que se deseja usar como remetente e para respostas. Além disso, é necessário verificar se a função mail() está corretamente configurada no servidor onde está hospedado o site, pois a função pode ser desabilitada ou exigir configurações adicionais para funcionar corretamente.**

## Arquivo processar_edicao.php

8. **O código processar_edicao.php é um script PHP responsável por atualizar informações de uma reserva existente no banco de dados. O script recebe os dados do formulário, como ID, nome completo, e-mail, telefone, data e hora e número de pessoas, e atualiza a entrada correspondente na tabela de reservas. Aqui está uma análise detalhada do código:**

a) **Inclui o arquivo config.php para estabelecer conexão com o banco de dados.**

b) **Verifica se a conexão com o banco de dados foi bem-sucedida, caso contrário, exibe uma mensagem de erro.**

c) **Recupera e escapa os valores dos campos do formulário usando mysqli_real_escape_string() para evitar ataques de injeção SQL.**

d) **Prepara uma instrução SQL para atualizar a entrada correspondente na tabela de reservas com os novos valores fornecidos.**

e) **Executa a instrução SQL usando mysqli_query(). Se a atualização for bem-sucedida, exibe um alerta informando que a reserva foi atualizada com sucesso e redireciona o usuário para index.html. Caso contrário, exibe um alerta com a mensagem de erro.**

**O código está correto e funcionará conforme o esperado, desde que os nomes dos campos no formulário correspondam aos nomes esperados no script. Certifique-se de que o arquivo config.php e a tabela de reservas estejam configurados corretamente, e que a estrutura da tabela seja compatível com as colunas mencionadas no script.**

## Arquivo processar_exclusao.php

9. **O código processar_exclusao.php é um script PHP responsável por excluir uma reserva existente no banco de dados. O script recebe o ID da reserva a ser excluída e executa uma instrução SQL para excluir a entrada correspondente na tabela de reservas. Aqui está uma análise detalhada do código:**

a) **Inclui o arquivo config.php para estabelecer conexão com o banco de dados.**

b) **Recupera e escapa o valor do ID da reserva usando mysqli_real_escape_string() para evitar ataques de injeção SQL.**

c) **Prepara uma instrução SQL para excluir a entrada correspondente na tabela de reservas usando o ID fornecido.**

d) **Executa a instrução SQL usando mysqli_query(). Se a exclusão for bem-sucedida, exibe um alerta informando que a reserva foi excluída com sucesso e redireciona o usuário para index.html. Caso contrário, exibe um alerta com a mensagem de erro.**

e) **Fecha a conexão com o banco de dados usando mysqli_close().**

**O código está correto e funcionará conforme o esperado, desde que o arquivo config.php e a tabela de reservas estejam configurados corretamente, e que a estrutura da tabela seja compatível com as colunas mencionadas no script. Certifique-se de que o formulário de exclusão esteja enviando o ID da reserva corretamente e que o nome do campo no formulário corresponda ao nome esperado no script.**

## Arquivo reservas.php

10. **O código reservas.php é um script PHP que lida com o processo de inserção de novas reservas no banco de dados. Ele inclui o arquivo config.php para estabelecer a conexão com o banco de dados e verifica se a conexão foi bem-sucedida. Aqui está uma análise detalhada do código:**

a) **Inclui o arquivo config.php para estabelecer a conexão com o banco de dados.**

b) **Verifica se há um erro de conexão. Se houver um erro, o script é encerrado e a mensagem de erro é exibida.**

c) **Verifica se o método de solicitação é POST, o que significa que o formulário foi enviado.**

d) **Prepara uma instrução SQL para inserir uma nova reserva na tabela de reservas. A instrução inclui espaços reservados (?) para os valores a serem inseridos.**

e) **Associa as variáveis (nome_completo, email, telefone, data_hora, num_pessoas, senha) aos espaços reservados na instrução SQL usando a**

função bind_param(). A string "ssssis" especifica os tipos de dados das variáveis (string, string, string, string, inteiro, string).

f) Atribui os valores dos campos do formulário às variáveis.

g) Executa a instrução SQL preparada usando a função execute(). Se a execução for bem-sucedida, exibe um alerta informando que a reserva foi feita com sucesso e redireciona o usuário para index.html. Caso contrário, exibe um alerta com a mensagem de erro e redireciona para index.html.

h) Fecha a instrução SQL preparada usando a função close().

i) Fecha a conexão com o banco de dados usando a função close().

O código está correto e funcionará conforme o esperado, desde que o arquivo config.php e a tabela de reservas estejam configurados corretamente e a estrutura da tabela seja compatível com as colunas mencionadas no script. Certifique-se de que o formulário de reserva esteja enviando os dados corretamente e que os nomes dos campos no formulário correspondam aos nomes esperados no script.

## Arquivo script.js

11. O código script.js é um arquivo JavaScript que contém três funções auto executáveis que lidam com a interação do usuário para a newsletter e duas janelas modais, uma para criar uma reserva e outra para visualizar uma reserva existente. Aqui está cada função:

a) Código da newsletter:

- Seleciona o elemento do formulário e cria um novo elemento <p> com uma mensagem de sucesso.
- Adiciona um ouvinte de evento 'submit' ao formulário.
- Quando o evento 'submit' é acionado, impede que o formulário seja enviado.

- **Seleciona os valores dos campos 'name' e 'email'.**

- **Aqui, você deve adicionar o código para enviar os dados do usuário ao seu banco de dados ou serviço de email marketing.**

- **Insere a mensagem de sucesso antes do próximo elemento do formulário e redefine o formulário.**

b) **Código da reserva:**

- **Seleciona o botão da reserva, a janela modal e o botão de fechar.**

- **Adiciona um ouvinte de evento 'click' ao botão da reserva que exibe a janela modal quando clicado.**

- **Adiciona um ouvinte de evento 'click' ao botão de fechar que oculta a janela modal quando clicado.**

- **Adiciona um ouvinte de evento 'click' à janela que fecha a janela modal se o usuário clicar fora do conteúdo da janela modal.**

c) **Código da minha reserva:**

- **Seleciona o botão da minha reserva, a janela modal da minha reserva e o botão de fechar.**

- **Adiciona um ouvinte de evento 'click' ao botão da minha reserva que exibe a janela modal da minha reserva quando clicado.**

- **Adiciona um ouvinte de evento 'click' ao botão de fechar da minha reserva que oculta a janela modal da minha reserva quando clicado.**

- **Adiciona um ouvinte de evento 'click' à janela que fecha a janela modal da minha reserva se o usuário clicar fora do conteúdo da janela modal.**

**O código está correto e funcionará conforme o esperado, desde que os elementos do DOM estejam configurados corretamente e correspondam aos seletores utilizados no script. No entanto, observe que o código para enviar os dados do**

usuário ao seu banco de dados ou serviço de email marketing não está presente no código da newsletter. Será preciso adicionar essa funcionalidade de acordo com as necessidades do projeto.

## Arquivo cardapio.html

12. O código cardapio.html está bem organizado e estruturado. No geral, o código parece bem construído e deve funcionar corretamente. No entanto, sempre é uma boa ideia testar o site em diferentes navegadores e dispositivos para garantir uma experiência consistente e agradável para os usuários.

## Arquivo contato.html

13. O arquivo contato.html parece estar bem estruturado e organizado. Aqui estão algumas observações e sugestões para melhorar ainda mais o código:

    a) É uma boa prática mover estilos inline para um arquivo CSS separado. Isso mantém a separação de preocupações e torna o código mais fácil de gerenciar. Por exemplo, os estilos inline na tag <h4> e nos elementos <label> podem ser movidos para o arquivo CSS.

    b) A tag  está obsoleta e não é recomendada. Em vez disso, use CSS para centralizar o conteúdo. Por exemplo, você pode adicionar text-align: center; na declaração de estilo para o elemento <h4>.

    c) No formulário de contato, o atributo action e o método (POST ou GET) estão faltando. Adicione o atributo action com o caminho para o arquivo de processamento do servidor e o método apropriado (geralmente POST para envio de formulários).

    d) Na tag <input> com o tipo tel para o campo de email, o tipo deve ser alterado para email.

e) **Na modal de reserva, o tipo de <input> para os campos "Nome completo" e "Endereço de e-mail" está incorreto como tel. Altere-os para text e email, respectivamente.**

f) **Ao usar um script externo, como jQuery, certifique-se de que a versão seja a mais recente e segura disponível. Além disso, verifique se os arquivos JavaScript referenciados estão disponíveis e funcionando corretamente.**

g) **Pode ser útil adicionar comentários ao código para explicar as diferentes seções e funcionalidades do código, facilitando a manutenção e o entendimento por outras pessoas.**

h) **Certifique-se de que todas as imagens referenciadas estão disponíveis e funcionando corretamente.**

**No geral, o código parece bem construído e deve funcionar corretamente. No entanto, sempre é uma boa ideia testar o site em diferentes navegadores e dispositivos para garantir uma experiência consistente e agradável para os usuários.**

## Arquivo contato.html

14. **O arquivo sobre.html é uma página da web que apresenta informações sobre o restaurante fictício "Cantinho da Gula". Segue a análise do código:**

a) **A estrutura básica do documento HTML, com a declaração <!DOCTYPE html> e as tags <html>, <head> e <body>.**

b) **Dentro da tag <head>:**

- **A tag <meta charset="UTF-8"> define a codificação de caracteres do documento.**
- **A tag <title> define o título da página como "Cantinho da Gula".**
- **A tag <link> importa o arquivo de estilos CSS chamado "style.css".**

c) **Dentro da tag <body>:**

- **O cabeçalho da página é definido com a tag <header> e inclui o logotipo do restaurante, bem como a navegação com links para as páginas "Início", "Cardápio", "Contato" e "Sobre".**

- **A tag <main> contém o conteúdo principal da página, que consiste em duas seções: "SOBRE NÓS" e "Nossa História". Ambas as seções incluem títulos e parágrafos estilizados com atributos style.**

- **A tag <footer> define o rodapé da página, que inclui o logotipo do restaurante e o texto de direitos autorais.**

- **Um botão "RESERVAS" é incluído, que abre um modal para realizar reservas. O modal contém um formulário com campos para nome completo, e-mail, telefone, data e hora da reserva, número de pessoas e senha.**

- **Um botão "Minhas Reservas" também é incluído no modal, que abre outro modal para consultar as reservas existentes. Esse modal contém um formulário com campos para e-mail e senha.**

d) **Os scripts JavaScript e bibliotecas incluídas no final da tag <body>:**

- **jQuery: Uma biblioteca JavaScript popular para simplificar a manipulação do DOM, eventos e animações.**

- **Popper.js: Uma biblioteca para gerenciar pop-ups e tooltips.**

- **Bootstrap: Um framework de componentes de design responsivo.**

- **script.js: Um arquivo JavaScript personalizado para adicionar funcionalidades específicas do site.**

**Em resumo, o arquivo sobre.html apresenta informações sobre o restaurante "Cantinho da Gula" e permite aos usuários fazer e consultar reservas através de modais. Além disso, a página possui navegação para outras páginas do site e utiliza bibliotecas populares para melhorar a experiência do usuário.**

# Análise Geral

Após uma análise detalhada do código, identifiquei algumas inconsistências e possíveis melhorias que poderiam ser aplicadas ao código:

1) No cabeçalho (header) da página, há um trecho de código comentado. Embora isso não seja um erro, é uma boa prática remover códigos comentados não utilizados antes de colocar a página em produção.

2) Na tag <h1> da seção "Nossa História", a tag de fechamento está incorreta. Está como </h2>, mas deveria ser </h1>. O código correto seria:

```
<h1 style="font-weight: bold; font-size: 350%; font-family: 'Courier New', Courier, monospace;">Nossa História</h1>
```

3) Nas tags de input do formulário de reservas e do formulário de consulta de reservas, o atributo type está incorreto para os campos "Nome completo" e "Endereço de e-mail". O atributo type deve ser text e email, respectivamente. O código correto seria:

```
<!-- Formulário de Reservas -->
<label for="nome-completo">Nome completo:</label>
<input type="text" id="nome-completo" name="nome-completo" required>

<label for="email">Endereço de e-mail:</label>
<input type="email" id="email" name="email" required>


<!-- Formulário de Consulta de Reservas -->
<label for="email-reserva">Endereço de e-mail:</label>
<input type="email" id="email-reserva" name="email-reserva" required>
```

4) No código, o atributo style é usado para aplicar estilos CSS diretamente nos elementos HTML. Embora isso não seja um erro, seria mais apropriado mover esses estilos para o arquivo CSS (style.css) e aplicar classes aos elementos HTML. Isso tornará o código mais fácil de manter e seguirá a prática recomendada de separação de preocupações.

**Após corrigir essas inconsistências e melhorias sugeridas, o código ficará mais limpo, fácil de manter e seguirá as práticas recomendadas de desenvolvimento web.**

**A conclusão final é de que o código funciona corretamente tendo todas suas “partes” verificadas as alterações seriam como uma sugestão para a melhoria da funcionalidade do mesmo.**