**Instruções para o teste do código:**

Primeiramente, criar um banco de dados MySQL com o nome “clientes” e duas tabelas; A primeira tabela terá o nome “tasks” e possuirá 6 colunas ( id, nome, descrição, anexos, prioridade, usuário); A segunda tabela receberá o nome de “usuarios” e terá 5 colunas (id, nome, email, senha, usuario). O site irá rodar localmente (<http://localhost>).

**Desenvolvimento:**

* **Index:** É a pagina inicial do projeto, consiste em uma mensagem de bem vindo e um botão para autenticação Google. Para essa parte do código, foi necessário algumas vídeo aulas, para o desenvolvimento correto do sistema de validação da conta Google inserida, uma vez que não possuia muito domínio sobre esse assunto. Além de vídeo aulas, procurei compreender o tutorial do Google API para a utilização dos scripts.   
   O e-mail inserido na validação será comparado com o email que está registrado na tabela “usuários”, coluna “email”, do banco de dados “clientes”.  
  Foram utilizados os scripts disponíveis na pagina da API Google.
* **Validacao:** De suma importância para a página Web, pois é aqui onde será feita a verificação de fato, porque o laço “if” processará a informação recebida pelo input Google, uma vez que a partir da condição estabelecida foi possível determinar se haveria o redirecionamento para a pagina “main.php” ou se será apresentado uma mensagem de Erro. Os parâmetros usados dentro do laço, é a variável “resultado” ser TRUEe se o número de linhas é diferente de zero, isto é encontrar um resultado compatível com o banco de dados.   
  Foram necessárias vídeo aulas, pois não havia tido muito contato, até então, com a linguagem php. Dessa forma, foi preciso entender melhor sua sintaxe e formas para criar a condição de validação.  
  .
* **Main:** Pagina onde o usuário é redirecionado após a autenticação. Consiste em um link que leva ao “inserir.php”, onde poderão ser criadas novas tasks. Além disso, apresenta uma tabela com as tasks que existem no banco de dados e existem botões de “editar”, que redirecionam a “editar.php”, e de “excluir” capaz de remover a task do banco de dados. Por fim, um link para sair da conta Google logada, utilizando scripts disponibilizados pelo Google.

Foi assistido vídeos aulas para entender melhor sobre a sintaxe php e como usar a função “foreach”.

* **Inserir/Editar:** Ambos os códigos possuem estruturas parecidas, uma vez que um é para a inserção de novas tasks e outro para alteração. Consiste em uma tabela para adicionar uma task ou alterar informações de uma task já existente.  
  Dentro do projeto, o “inserir.php” e o “editar.php” foram as partes com menor nível de dificuldade, pois consistiu em inputs do tipo texto, file, submit e hidden.
* **Conexao:** Essa parte do código, está responsável em armazenar as funções utilizadas durante todo a pagina Web, desde abrir o banco de dados, até alterações e exclusões de tasks. Além disso, a função para redirecionar a pagina pro “main.php”, a capaz de montar a array que contem todas as task, que compõe a tabela da pagina “main.php” e a que consegue identificar o id da task selecionada para edição.

**Dificuldades e Desafios:**

Como foi apresentado no primeiro contato que tive com a Voxus, eu possuo conhecimento na linguagem JAVA e sobre a lógica de programação, sendo assim, esse desafio proposto foi um grande motivador para o entendimento da sintaxe da linguagem php, html e javascript, conhecimentos usados para o desenvolvimento do mesmo. Foram necessárias vídeo aulas e pesquisas para compreender corretamente o uso das funções em php e como usar o Banco de Dados. Acredito que foi possível aprender bastante com esse Desafio Voxus.