- [ ] 1. Preparação do Ambiente de Desenvolvimento
  - [ ] 1.1. Instalar o .NET SDK (Software Development Kit)
  - [ ] 1.2. Instalar o MySQL Server e MySQL Workbench (ou outra ferramenta de gerenciamento)
  - [ ] 1.3. Instalar um editor de código (Visual Studio Code ou Visual Studio Community)
- [ ] 2. Criação do Projeto .NET MVC
  - [ ] 2.1. Criar um novo projeto ASP.NET Core MVC
  - [ ] 2.2. Entender a estrutura de pastas (Models, Views, Controllers, wwwroot)
- [ ] 3. Configuração Inicial do Projeto
  - [ ] 3.1. Configurar a string de conexão com o MySQL local
  - [ ] 3.2. Adicionar pacotes NuGet necessários (MySQL Connector, Entity Framework Core se for usar, Google Authentication)
- [ ] 4. Design e Layout Base (Material Design)
  - [ ] 4.1. Analisar os componentes do Figma e identificar como replicá-los com HTML/CSS/JS. Considerar bibliotecas como Materialize CSS ou Bootstrap com tema Material, ou criar estilos customizados.
  - [ ] 4.2. Criar o layout base (`_Layout.cshtml`) com a estrutura principal: header (com o nome do site/logo), menu lateral (acionado pelo botão de três listras) e footer.
  - [ ] 4.3. Estilizar o layout base com CSS e JavaScript para replicar o design do Figma. Implementar o menu lateral navegável com as opções: Início, Cardápio, Agendamento de Pedidos.
- [ ] 5. Banco de Dados e Modelos de Dados (Entidades)
  - [ ] 5.1. Definir os modelos (classes C# em `Models`): `Usuario` (para informações do login do Google), `Produto` (comida: nome, descrição, preço, imagem), `Agendamento` (usuário, data/hora, itens do pedido, status).
  - [ ] 5.2. Configurar o Entity Framework Core (ORM) para mapear esses modelos para tabelas no MySQL. Isso inclui criar um `DbContext` .
  - [ ] 5.3. Usar Migrations do EF Core para criar/atualizar o esquema do banco de dados MySQL local.
- [ ] 6. Funcionalidade de Autenticação com Google
  - [ ] 6.1. Registrar o aplicativo no Google Cloud Console para obter `ClientID` e `ClientSecret` .
  - [ ] 6.2. Instalar o pacote NuGet `Microsoft.AspNetCore.Authentication.Google` .
  - [ ] 6.3. Configurar o serviço de autenticação do Google no `Program.cs` (ou `Startup.cs`), adicionando o `ClientID` e `ClientSecret` .

    - [ ] 6.4. Criar um `AccountController` com Actions para `Login` (redireciona para o Google) e `GoogleCallback` (processa a resposta do Google e loga o usuário).
    - [ ] 6.5. Adicionar botões/links de "Login com Google" nas Views apropriadas e lógica para exibir informações do usuário logado ou o botão de login.
- [ ] 7. Desenvolvimento das Páginas e Funcionalidades Principais
    - [ ] 7.1. Página Inicial (`HomeController` , `Index` Action e View)
        - [ ] 7.1.1. Conteúdo: Descrição sobre a loja. Deve usar o `_Layout.cshtml` .
    - [ ] 7.2. Página de Cardápio (`CardapioController`)
        - [ ] 7.2.1. Action `Index` : Listar todos os produtos (comidas) do banco de dados. Exibir nome, descrição resumida, preço e imagem.
        - [ ] 7.2.2. View `Index` : Apresentar os produtos de forma organizada, seguindo o design. Cada produto pode ter um link/botão para "Ver detalhes" ou "Adicionar ao pedido".
        - [ ] 7.2.3. (Opcional) Action `Detalhes(int id)` : Mostrar detalhes de um produto específico.
    - [ ] 7.3. Página de Agendamento de Pedidos (`AgendamentoController`)
        - [ ] 7.3.1. Requer que o usuário esteja logado.
        - [ ] 7.3.2. Action `Criar` : Exibir um formulário para o usuário selecionar itens do cardápio (talvez um mini-carrinho), escolher data/hora para o agendamento.
        - [ ] 7.3.3. View `Criar` : Formulário com campos para os detalhes do agendamento. Validação de dados.
        - [ ] 7.3.4. Action `Criar` (POST): Salvar o agendamento no banco de dados, associado ao usuário logado.
        - [ ] 7.3.5. Action `MeusAgendamentos` : Listar os agendamentos feitos pelo usuário logado.
        - [ ] 7.3.6. View `MeusAgendamentos` : Exibir a lista de agendamentos.
- [ ] 8. Implementação do Menu Lateral Navegável
    - [ ] 8.1. Garantir que o botão de "três listras" no header abra/feche um menu lateral.
    - [ ] 8.2. O menu deve conter links para: Início, Cardápio, Agendamento de Pedidos (e talvez "Meus Agendamentos" e "Logout" se o usuário estiver logado).
- [ ] 9. Testes e Refinamentos
    - [ ] 9.1. Testar todas as funcionalidades: login, visualização do cardápio, criação de agendamento, visualização de agendamentos.
    - [ ] 9.2. Verificar a responsividade do layout em diferentes tamanhos de tela.
    - [ ] 9.3. Ajustar CSS e JavaScript conforme necessário para alinhar com o design do Figma e garantir boa usabilidade.

- [ ] 10. Preparação para Futuro Deploy no Azure e Migração do Banco
    - [ ] 10.1. Documentar os passos para criar um App Service no Azure.
    - [ ] 10.2. Documentar como configurar o Azure Database for MySQL.
    - [ ] 10.3. Explicar como alterar a string de conexão no projeto para apontar para o banco de dados na nuvem.
    - [ ] 10.4. Detalhar o processo de publicação do projeto .NET para o Azure App Service.
- [ ] 11. Criação do Cronograma e Documentação Final para o Usuário
    - [ ] 11.1. Organizar todos os passos de implementação em um guia detalhado, explicando o "como" e o "porquê" de cada etapa, com exemplos de código quando aplicável.
    - [ ] 11.2. Estimar um cronograma realista para um iniciante completar cada uma das fases principais.