

A SEGUIR SÃO APRESENTADAS TODAS AS ETAPAS DO DESENVOLVIMENTO DO PROJETO PARA O 4º BIMESTRE DA DISCIPLINA DE SW-I

Ao final também é apresentado um **checklist do que será avaliado** nos trabalhos entregues ao final do 4º bimestre.

Etapa 1 – Escolha do Tema e Levantamento de Requisitos

Etapa 1 – Escolha do Tema e Levantamento de Requisitos

Objetivo: Escolher um dos temas e listar o que o sistema irá fazer (funcionalidades básicas).

Entregáveis:

- Tema escolhido
- Descrição do sistema (1 parágrafo)
- Lista de funcionalidades (mínimo 3)

Etapa 2 – Modelagem do Banco de Dados (DER)

Objetivo: Criar o diagrama com as tabelas e os relacionamentos.

Ferramentas sugeridas:

- dbdiagram.io
- draw.io
- papel mesmo, se necessário

Dica:

Explique o básico de **entidade, atributo, relacionamento, chave primária e chave estrangeira**.

Etapa 3 – Criação do Banco no MySQL

Objetivo: Criar o banco de dados e as tabelas com base no DER.

Entregáveis:

- Script SQL com:
 - Criação do banco
 - Criação das tabelas
 - Inserção de dados fictícios

Etapa 4 – Estruturação do Projeto em PHP

Objetivo: Criar os arquivos básicos do projeto.

Sugestão de estrutura:

lua

/projeto

```
|-- index.php           // Página inicial
|-- sobre.php           // Página "Sobre"
|-- login.php           // Tela de login
|-- dashboard.php       // Tela pós-login
|-- /includes
    |-- conexao.php      // Conexão com o banco
    |-- header.php       // Cabeçalho
    |-- footer.php       // Rodapé
|-- /crud               // Telas de controle (CRUDs)
```

Etapa 5 – Tela de Login com Sessão

Objetivo: Criar sistema básico de login com sessão.

Requisitos:

- Tabela `usuarios` (com `email`, `senha` criptografada com `password_hash`)
- Validação de login
- Redirecionamento para `dashboard.php` se logado
- `logout.php` para sair

Etapa 6 – Telas de Controle (CRUDs)

Objetivo: Criar as telas para cadastrar, listar, editar e excluir registros.

Exemplo (tema: Veterinária):

- Cadastrar animal (nome, espécie, dono)
- Editar dados do animal
- Excluir animal
- Listar todos os animais

Dica:

Utilizar formulários com `POST`, exibir dados com `SELECT`, e `UPDATE/DELETE` com base no ID.

Etapa 7 – Melhorias Visuais e Navegação

Objetivo: Melhorar a aparência e a usabilidade com HTML/CSS.

Sugestões:

- Usar um CSS simples ou framework como Bootstrap
- Criar um menu fixo no `header.php`
- Exibir mensagens de sucesso/erro com cores

Etapa 8 – Testes e Apresentação

Objetivo: Validar o funcionamento e apresentar o projeto.

Checklist para entrega final:

- Login funcionando?
- CRUDs completos?
- Conexão com o banco estável?
- Navegação entre páginas?
- Interface clara?
- Comentários no código?

Resumo de Funcionalidades por Tema

Tema	Entidades principais	Ações (CRUDs) principais
Veterinária	Animais, Donos, Consultas	Cadastrar animais, agendar consultas
Concessionária	Carros, Clientes, Vendas	Cadastrar veículos, registrar venda
Biblioteca	Livros, Usuários, Empréstimos	Cadastrar livros, registrar empréstimos
Review de Filmes	Filmes, Usuários, Avaliações	Cadastrar filmes, postar avaliações

Checklist Geral para Avaliação dos Projetos

1. Planejamento e Documentação

- ☐ Tema do projeto claramente definido e apropriado
 - ☐ Descrição do sistema e funcionalidades bem detalhadas
 - ☐ Modelo do banco de dados (DER) entregue e consistente com o sistema
 - ☐ Script SQL para criação do banco e tabelas entregue e funcional
-

2. Banco de Dados

- ☐ Tabelas criadas corretamente conforme o DER
 - ☐ Relacionamentos (chaves estrangeiras) definidos e aplicados
 - ☐ Dados de teste inseridos para demonstração
 - ☐ Uso correto de tipos de dados e índices para performance
-

3. Estrutura do Projeto em PHP

- ☐ Arquivos organizados em pastas (ex: includes, crud, etc.)
 - ☐ Conexão com o banco de dados isolada em arquivo próprio
 - ☐ Código PHP limpo e comentado
 - ☐ Uso adequado de includes para evitar repetição
-

4. Funcionalidades do Sistema

- ☐ Página inicial criada e funcional
 - ☐ Página "Sobre" criada e informativa
 - ☐ Tela de login funcionando corretamente (com validação)
 - ☐ Controle de sessão implementado (login e logout)
 - ☐ Área restrita acessível somente após login
 - ☐ CRUDs implementados para entidades principais (Create, Read, Update, Delete)
 - ☐ Mensagens de feedback (sucesso/erro) nas ações do usuário
-

5. Segurança

- ☐ Senhas armazenadas usando hash seguro (`password_hash`)
 - ☐ Validação dos dados recebidos via formulário (ex: sanitização)
 - ☐ Prevenção contra SQL Injection (uso de prepared statements ou escape adequado)
 - ☐ Proteção das páginas restritas contra acesso sem login
-

6. Interface e Usabilidade

- ☐ Layout organizado e navegação intuitiva
 - ☐ Uso consistente de cabeçalho e rodapé em todas as páginas
 - ☐ Formulários claros e fáceis de usar
 - ☐ Feedback visual para erros e confirmações
 - ☐ Uso de CSS básico ou framework para melhorar aparência
-

7. Testes e Funcionamento

- ☐ Todas as funcionalidades principais testadas e funcionando
 - ☐ Login e logout operam corretamente
 - ☐ CRUDs realizam operações no banco sem erros
 - ☐ Sistema lida com dados inválidos de forma adequada
 - ☐ Não há erros ou warnings exibidos no navegador
-

8. Apresentação e Organização

- ☐ Código organizado, legível e comentado
 - ☐ Projeto entregue dentro do prazo
 - ☐ Demonstração clara das funcionalidades no momento da apresentação
 - ☐ Respostas adequadas às perguntas sobre o sistema e código
-