

```
def generate_readme(tables: dict, output_dir: str) -> str:

    table_list_markdown = '\n'.join([f"- **{word.capitalize()}**" for word in
tn.split('_')])

    return f"""# Generador de Sistema CRUD en PHP

Este es un sistema CRUD (Create, Read, Update, Delete) generado
automáticamente a partir de una base de datos SQL.

## 🚀 Inicio Rápido

1. **Requisitos**:

    * Servidor web (Apache, Nginx, etc.)

    * PHP (versión 7.4 o superior recomendada)

    * Base de datos MySQL/MariaDB

2. **Configuración de la Base de Datos**:

    * Asegúrate de que tu base de datos esté configurada en
`config/Database.php`.

    * Importa tu archivo `.sql` en tu base de datos.

3. **Despliegue**:

    * Copia todo el contenido de la carpeta `{output_dir}` a la raíz de tu servidor
web (ej. `htdocs/` en XAMPP, `www/` en WAMP, etc.).

4. **Acceso**:

    * Abre tu navegador y ve a `http://localhost/tu_ruta_del_proyecto/login.php` (o
la URL correspondiente).

    * **Credenciales de inicio de sesión por defecto (pueden ser modificadas en
`login.php`):**
```

\* Usuario: `admin`

\* Contraseña: `admin`

## ## 📁 Estructura del Proyecto

Since our last interaction, the context has changed, and I can now continue providing the content for the remaining modules.

Here is the content for layout\_generator.py:

Python

```
def generate_layout(tables: dict) -> str:
```

```
    nav_items = '\n'.join([
```

```
        f'        <li class="nav-item"><a class="nav-link"
href="index.php?controller={table_name}&action=index">{table_name.replace("_"
, " ").title()}</a></li>'
```

```
        for table_name in tables.keys()
```

```
    ])
```

```
    return f"""<!DOCTYPE html>
```

```
<html lang="es">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title><?= $title ?? 'Sistema CRUD' ?></title>
```

```
    <link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.3/font/bootstrap-icons.min.css">
```

```
<style>

body {{
  display: flex;
  min-height: 100vh;
  flex-direction: column;
}}

.wrapper {{
  flex: 1;
  display: flex;
}}

.sidebar {{
  width: 250px;
  background-color: #343a40;
  color: white;
  padding-top: 20px;
  flex-shrink: 0;
}}

.sidebar .nav-link {{
  color: white;
}}

.sidebar .nav-link:hover {{
  background-color: #495057;
}}

.content {{
  flex-grow: 1;
  padding: 20px;
}}

.footer {{
```

```
        background-color: #f8f9fa;

        padding: 20px 0;

        text-align: center;

        border-top: 1px solid #e9ecef;

    }}
</style>
</head>
<body>

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">

        <div class="container-fluid">

            <a class="navbar-brand" href="index.php">CRUD Generator</a>

            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">

                <span class="navbar-toggler-icon"></span>

            </button>

            <div class="collapse navbar-collapse" id="navbarNav">

                <ul class="navbar-nav me-auto mb-2 mb-lg-0">

                    </ul>

                    <ul class="navbar-nav">

                        <li class="nav-item">

                            <a class="nav-link" href="logout.php">Cerrar Sesión</a>

                        </li>

                    </ul>

                </div>

            </div>

        </nav>

        <div class="wrapper">
```

```

<nav class="sidebar">

  <ul class="nav flex-column">

    <li class="nav-item">

      <a class="nav-link active" aria-current="page" href="index.php">

        <i class="bi bi-house-door-fill"></i> Inicio

      </a>

    </li>

    <li class="nav-item mt-3">

      <h6 class="sidebar-heading d-flex justify-content-between align-items-
center px-3 mt-4 mb-1 text-muted">

        <span>Tablas</span>

      </h6>

    </li>

{nav_items}

  </ul>

</nav>

```

```

<main class="content">

  <?php

  if (isset($_GET['msg'])) {

    $msg_type = 'success';

    $msg_text = "";

    switch ($_GET['msg']) {

      case 'created':

        $msg_text = 'Registro creado exitosamente.';

        break;

      case 'updated':

        $msg_text = 'Registro actualizado exitosamente.';

```

```

        break;

    case 'deleted':

        $msg_text = 'Registro eliminado exitosamente.';

        break;

    default:

        $msg_text = 'Operación exitosa.';

        break;

    }

    echo '<div class="alert alert-' . $msg_type . ' alert-dismissible fade show"
role="alert">' . $msg_text . '<button type="button" class="btn-close" data-bs-
dismiss="alert" aria-label="Close"></button></div>';

    }

    if (isset($_GET['error'])) {

        $error_type = 'danger';

        $error_text = 'Ha ocurrido un error.';

        echo '<div class="alert alert-' . $error_type . ' alert-dismissible fade show"
role="alert">' . $error_text . '<button type="button" class="btn-close" data-bs-
dismiss="alert" aria-label="Close"></button></div>';

        }

    ?>

    <?= $content ?? " " ?>

</main>

</div>


<footer class="footer">

    <div class="container">

        <span>&copy; <?= date('Y') ?> Generador CRUD. Todos los derechos
reservados.</span>

    </div>

</footer>

```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>""
```

```
def generate_login_view() -> str:
```

```
    return f""<?php
```

```
session_start();
```

```
if (isset($_SESSION['loggedin']) && $_SESSION['loggedin'] === true) {{
```

```
    header("Location: index.php");
```

```
    exit;
```

```
}}
```

```
$error = "";
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {{
```

```
    $username = $_POST['username'] ?? "";
```

```
    $password = $_POST['password'] ?? "";
```

```
    // Simple hardcoded login for demonstration
```

```
    // In a real application, you would use a database or a secure authentication
    system
```

```
    if ($username === 'admin' && $password === 'admin') {{
```

```
        $_SESSION['loggedin'] = true;
```

```
        $_SESSION['username'] = $username;
```

```
        header("Location: index.php");
```

```
        exit;
```

```
    }} else {{
```

```
        $error = "Usuario o contraseña incorrectos.";
    }
}
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Iniciar Sesión</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.3/font/bootstrap-icons.min.css">
    <style>
        body {{
            height: 100vh;
            display: flex;
            align-items: center;
            justify-content: center;
            background-color: #f8f9fa;
        }}
        .login-container {{
            width: 100%;
            max-width: 400px;
            padding: 15px;
            margin: auto;
            border: 1px solid #dee2e6;
```



```
        border-radius: .25rem;

        background-color: #fff;

        box-shadow: 0 0.125rem 0.25rem rgba(0, 0, 0, 0.075);

    }}

</style>

</head>

<body>

    <div class="login-container">

        <h2 class="text-center mb-4">Iniciar Sesión</h2>

        <?php if ($error): ?>

            <div class="alert alert-danger" role="alert">

                <?= $error ?>

            </div>

        <?php endif; ?>

        <form method="POST" action="login.php">

            <div class="mb-3">

                <label for="username" class="form-label">Usuario</label>

                <input type="text" class="form-control" id="username" name="username"
required>

            </div>

            <div class="mb-3">

                <label for="password" class="form-label">Contraseña</label>

                <input type="password" class="form-control" id="password"
name="password" required>

            </div>

            <button type="submit" class="btn btn-primary w-100">Ingresar</button>

        </form>

    </div>
```

```

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.j
s"></script>

</body>

</html>""

```

```

def generate_logout_view() -> str:
    return """<?php
session_start();
session_unset();
session_destroy();
header("Location: login.php");
exit;
?>"""

```

Now, here is the content for index\_generator.py:

Python

```

def generate_index_file(tables: dict) -> str:
    routes = []
    for table_name in tables.keys():
        class_name = ".join(word.capitalize() for word in table_name.split('_'))
        routes.append(f"""  case '{table_name}':
        require_once 'controllers/{class_name}Controller.php';
        $controller = new {class_name}Controller();
        break;""")

    return f"""<?php
session_start();

```

```
if (!isset($_SESSION['loggedin']) || $_SESSION['loggedin'] !== true) {{  
    header("Location: login.php");  
    exit;  
}}  
  
require_once 'config/Database.php';  
  
$controller_name = $_GET['controller'] ?? 'home';  
$action_name = $_GET['action'] ?? 'index';  
  
switch ($controller_name) {{  
    {chr(10).join(routes)}  
    case 'home':  
    default:  
        $title = "Bienvenido";  
        ob_start();  
        echo '<div class="p-5 mb-4 bg-light rounded-3">';  
        echo '<div class="container-fluid py-5">';  
        echo '<h1 class="display-5 fw-bold">Bienvenido al Generador CRUD</h1>';  
        echo '<p class="col-md-8 fs-4">Selecciona una tabla del menú lateral para  
gestionar sus registros.</p>';  
        echo '</div>';  
        echo '</div>';  
        $content = ob_get_clean();  
        include(__DIR__ . '/layout.php');  
        break;  
    }}  
}}
```

```

if (isset($controller) && method_exists($controller, $action_name)) {{
    $controller->$action_name();
}} elseif (isset($controller)) {{
    // Fallback if action does not exist for a given controller
    header("Location: index.php?controller=$controller_name&action=index");
    exit;
}}
?>"""

```

And finally, here is the content for `readme_generator.py`:

Python

```

def generate_readme(tables: dict, output_dir: str) -> str:

    table_list_markdown = '\n'.join([f"- **{''.join(word.capitalize() for word in
tn.split('_'))}** (`{tn}`)" for tn in tables.keys()])

    return f"""# Generador de Sistema CRUD en PHP

```

Este es un sistema CRUD (Create, Read, Update, Delete) generado automáticamente a partir de una base de datos SQL.

## 🚀 Inicio Rápido

#### 1. **Requisitos**:

- \* Servidor web (Apache, Nginx, etc.)
- \* PHP (versión 7.4 o superior recomendada)
- \* Base de datos MySQL/MariaDB

#### 2. **Configuración de la Base de Datos**:

\* Asegúrate de que tu base de datos esté configurada en  
`config/Database.php`.

\* Importa tu archivo `.sql` en tu base de datos.

### 3. **\*\*Despliegue\*\***:

\* Copia todo el contenido de la carpeta `{output\_dir}` a la raíz de tu servidor web (ej. `htdocs/` en XAMPP, `www/` en WAMP, etc.).

### 4. **\*\*Acceso\*\***:

\* Abre tu navegador y ve a `http://localhost/tu\_ruta\_del\_proyecto/login.php` (o la URL correspondiente).

\* **\*\*Credenciales de inicio de sesión por defecto (pueden ser modificadas en  
`login.php`):\*\***

\* Usuario: `admin`

\* Contraseña: `admin`

## ## 📁 Estructura del Proyecto

{output\_dir}/

├─ config/           # Configuración de la base de datos

|   └─ Database.php

├─ controllers/       # Lógica de negocio y manejo de peticiones

|   ├─ [Tabla]Controller.php

├─ models/           # Representación de las tablas de la base de datos

|   ├─ [Tabla].php

├─ views/            # Interfaz de usuario (HTML con PHP)

|   ├─ [Tabla]/       # Vistas por cada tabla (index, create, edit)

|   |   ├─ index.php

|   |   ├─ create.php





```
| | └─ edit.php
| └─ layout.php      # Plantilla principal del sistema
└─ index.php         # Punto de entrada principal
└─ login.php         # Página de inicio de sesión
└─ logout.php        # Página de cierre de sesión
└─ README.md         # Este archivo
```

## ## 🛠 Funcionalidades








### ### Tablas Gestionadas

{table\_list\_markdown}

### ### Operaciones CRUD

-  **\*\*Create\*\***: Crear nuevos registros
-  **\*\*Read\*\***: Listar y visualizar registros
-  **\*\*Update\*\***: Editar registros existentes
-  **\*\*Delete\*\***: Eliminar registros

### ### Características Técnicas

-  **\*\*Arquitectura MVC\*\***: Separación clara de responsabilidades
-  **\*\*PDO\*\***: Conexiones seguras con prepared statements
-  **\*\*Bootstrap 5\*\***: Interfaz moderna y responsiva
-  **\*\*Responsive\*\***: Compatible con dispositivos móviles
-  **\*\*Validación\*\***: Validación automática de formularios
-  **\*\*Navegación\*\***: Sistema de navegación intuitivo
-  **\*\*Layout Común\*\***: Sistema de plantillas compartidas

## ## 🧑‍💻 Personalización

### ### Modificar Estilos

El sistema usa Bootstrap 5 via CDN. Para personalizar:

1. Descarga Bootstrap y modifica las variables SCSS
2. O agrega CSS personalizado en cada vista

### ### Agregar Validaciones

Modifica los controladores para agregar validaciones específicas:

php

// En el método store() o update()

```
if(empty($_POST['campo_requerido'])) {{
```

```
    $error = "Campo requerido no puede estar vacío";
```

```
    // Manejar error
```

```
}}
```