

# ArchMaster

## "Software Architecture Document Generator"

### Nome do Projeto:

Gerador de Recibos - Espaço Escolar Educ

### Descrição Geral:

O sistema desenvolvido permite que os administradores ou responsáveis do Espaço Escolar Educ gerem recibos personalizados para os alunos, contendo informações sobre o pagamento do curso, incluindo nome do aluno, nome do curso, valor pago e data de pagamento. O sistema oferece a opção de gerar um PDF com os dados inseridos e permite o download do recibo gerado.

### Stakeholders:

Proprietário da escola (Espaço Escolar Educ)  
Administradores do sistema  
Alunos (destinatários dos recibos gerados)  
Desenvolvedor Diego Nascimento

### Escopo:

Desenvolvimento de uma aplicação web com funcionalidades de formulário para inserção dos dados do aluno, curso, valor e data de pagamento.  
Geração de recibo em formato PDF com os dados inseridos.  
Interface de fácil usabilidade e responsive design para visualização em diversos dispositivos.

### Requisitos Funcionais:

O sistema deve permitir o preenchimento de informações sobre o aluno (nome), curso, valor pago e data de pagamento.  
Após o preenchimento do formulário, o sistema deve gerar um recibo na tela e possibilitar o download do recibo em formato PDF.  
O sistema deve exibir os dados da escola e do cliente no recibo gerado, além de permitir a assinatura digital ou manual.  
A aplicação deve ser responsiva, garantindo acesso e usabilidade em dispositivos móveis e desktop.

### Requisitos Não Funcionais:

A aplicação deve ser responsiva e garantir uma boa experiência em dispositivos móveis e desktop.  
O sistema deve ser rápido e capaz de gerar o recibo sem demora perceptível.  
A aplicação deve ser segura, evitando vazamento de informações sensíveis dos alunos.

## Diagrama de Componentes:

A aplicação será composta por três componentes principais:

Interface do Usuário (Frontend): Formulário para preenchimento dos dados e visualização do recibo gerado.

Lógica de Negócio (JavaScript): Responsável por manipular e processar os dados do formulário, gerar o recibo em PDF e exibir na interface.

Geração de PDF (jsPDF): Utilização da biblioteca jsPDF para converter as informações inseridas em um arquivo PDF.

## Visão de Camadas:

Camada de Apresentação: HTML, CSS, JavaScript (Frontend)

Camada de Lógica de Negócio: JavaScript para processamento de dados e criação do recibo PDF.

Camada de Persistência de Dados: Não aplicada diretamente, pois os dados são apenas manipulados e gerados dinamicamente, sem necessidade de banco de dados.

## Principais Tecnologias Utilizadas:

HTML, CSS, JavaScript (Frontend)

Biblioteca jsPDF para a geração de PDF

Design responsivo com CSS Flexbox e Media Queries.

## Front-end:

O front-end é construído com HTML, CSS e JavaScript. Utiliza formulários para capturar informações e exibir resultados em tempo real. O layout é responsivo, garantindo boa usabilidade em dispositivos móveis e desktop.

## Back-end:

Não há back-end, uma vez que o sistema não exige persistência de dados nem comunicação com servidores. Toda a lógica ocorre no cliente.

## Banco de Dados:

Não é necessário, pois as informações são processadas e geradas dinamicamente sem armazenamento persistente.

## Integração e APIs:

Integração com a biblioteca jsPDF para a geração de PDFs.

## Fluxo Principal:

O usuário acessa o sistema e preenche os campos do formulário com as informações necessárias.

Ao submeter o formulário, os dados são exibidos em uma área de pré-visualização do recibo.

O usuário clica no botão para gerar o PDF, e o sistema utiliza a biblioteca jsPDF para criar o arquivo PDF com os dados preenchidos.

O usuário pode então baixar o PDF gerado.

## Estrutura do Projeto:

```
/gerador-de-recibos
% % %   index.html           // Página inicial
% % %   css
%   % % % styles.css         // Estilo da página
% % %   js
%   % % % script.js          // Lógica JavaScript para o formulário e ge
```

## Padrões de Codificação:

- Código limpo e bem documentado
- Uso de indentação consistente (2 espaços)
- Nomes de variáveis e funções descritivos

## Controle de Versão:

- Uso do Git para controle de versão.
- Repositório GitHub para colaboração e versionamento.

## CI/CD:

Não aplicado, pois o projeto não requer integração contínua ou deployment automático, sendo uma aplicação simples.

## Autenticação e Autorização:

Não é necessária autenticação ou controle de acesso, pois a aplicação é pública e acessível a qualquer usuário.

## Proteção de Dados:

Nenhuma informação pessoal sensível é armazenada no sistema, pois os dados são manipulados localmente e não são enviados para servidores.

## Escalabilidade Horizontal/Vertical:

O projeto não requer escalabilidade, já que é uma aplicação simples e não exige processamento de grandes volumes de dados.

## Caching:

Não há necessidade de caching devido à natureza simples do sistema e ao uso de dados temporários.

## Monitoramento:

Não há monitoramento, pois o sistema não depende de back-end nem de servidores para seu funcionamento.v

### Recuperação de Desastres:

O sistema não exige recuperação de desastres devido à ausência de dependências externas ou persistência de dados.

### Backups:

Não se aplica, pois os dados não são armazenados de forma persistente.

### Próximos Passos:

Realizar testes com usuários reais para validar a experiência do usuário e a geração do PDF.  
Implementar melhorias no layout responsivo conforme feedback dos usuários.

### Conclusão:

Este projeto atende às necessidades do Espaço Escolar Educ, oferecendo uma solução simples e eficaz para a geração de recibos personalizados para alunos. O sistema é fácil de usar, rápido e funcional, e está pronto para ser utilizado em produção.