

**UTFPR - UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**Bacharelado em Engenharia de Software - 7º Período**

**DISCIPLINA:** *Oficina de Integração 2 - ES47C-IF66K*

**PROFESSOR:** *Antonio Carlos Fernandes da Silva*

---

# **Documento de**

# **Projeto de Software**

---

Raiane Stefane Campos Correia

Diomenes de Araujo Marangoni

Matheus Shiune Murasaki

Leonardo Conde Feitosa

Cornélio Procópio

2025

# **1. Introdução**

## **1.1 Contexto:**

Este projeto é desenvolvido no âmbito da disciplina extensionista **Oficina de Integração 2**, com o objetivo de promover a aplicação prática dos conhecimentos adquiridos em **Engenharia de Software**. A iniciativa incentiva os alunos a atenderem às demandas **reais do projeto de extensão ELLP - Ensino Lúdico de Lógica e Programação**.

## **1.2 Justificativa:**

O cadastro de participantes (professores, tutores e alunos), bem como o acompanhamento de escolas parceiras, o controle das oficinas e a emissão de documentos como convênios e certificados, quando realizados de forma manual, tornam-se suscetíveis a falhas, retrabalho e inconsistências nos dados.

A ausência de um sistema informatizado e integrado dificulta a centralização das informações, a geração de relatórios de desempenho e o monitoramento do impacto do projeto. Dessa forma, a implementação de uma solução tecnológica é fundamental para otimizar a gestão, garantir a integridade dos dados e ampliar a eficiência e a escalabilidade das ações do projeto de extensão.

## **1.3 Proposta:**

Propõe-se o desenvolvimento de um **Sistema de Gestão Integrado** para automatizar e centralizar a administração do projeto ELLP. A solução possuirá:

Gestão acadêmica e logística das atividades, abrangendo o cadastro de escolas, oficinas, turmas, professores, alunos, controle de inscrições, frequência, a emissão automatizada de certificados, geração de documentos formais (cartas convite e convênios) e o monitoramento do número de alunos atendidos por instituição.

## **1.4 Ferramentas:**

O desenvolvimento do projeto será atualizado no github:  
<https://github.com/RaianeCampos/OficinaIntegracao2>

Atribuição e acompanhamento de tarefas pelo quadro Kanban do Trello:  
<https://trello.com/invite/b/68c0be63bea36e3279739379/ATTI884446f96e983c9255ea8602bfa4a0d040AF9B36/kanban-projeto-oi2>

## 2. Descrição Geral do Sistema

### 2.1 Objetivos (Geral e Específicos)

- **Objetivo Geral:**

Desenvolver um sistema de software integrado e centralizado para automatizar e otimizar a gestão administrativa e pedagógica do projeto de extensão ELLP - Ensino Lúdico de Lógica e Programação, garantindo a eficiência no controle das oficinas, no relacionamento com as escolas parceiras e na geração de dados confiáveis para o monitoramento das atividades.

- **Objetivos Específicos:**

1. **Gerenciar pessoas:** Implementar o cadastro completo de professores e alunos.
2. **Cadastrar parceiros:** Centralizar o registro de escolas, atribuição de representantes.
3. **Organizar as oficinas:** Permitir a criação e o gerenciamento das escolas, oficinas e suas turmas, incluindo professor responsável e tutores.
4. **Controlar participação:** Automatizar as inscrições e o registro de frequência dos alunos.
5. **Emitir certificados:** Gerar certificados digitais, como código de validação para os concluintes de oficinas.
6. **Monitorar impacto:** Gerar relatórios para visualizar o número de alunos atendidos por escola.

### 2.2 Descrição das entidades do sistema

- **Escola**

- o Chave primária: IdEscola

Atributos:

- o IdEscola (integer)
- o NomeEscola (varchar)
- o CnpjEscola (varchar)
- o CepEscola (numeric(8))
- o RuaEscola (varchar)
- o ComplementoEscola (varchar)
- o TelefoneEscola (varchar)
- o EmailEscola (varchar)

Relacionamentos:

- o 1:N com Professor
- o 1:N com Documento

- **Professor**

- o Chave primária: IdProfessor
- o Chave estrangeira: IdEscola → Escola(IdEscola)

Atributos:

- o IdProfessor (integer)
- o NomeProfessor (varchar)
- o EmailProfessor (varchar)
- o TelefoneProfessor (numeric)
- o ConhecimentoProfessor (varchar)
- o CargoProfessor (varchar)
- o IdEscola (integer)
- o Representante (boolean)
- o Administrador (boolean)

Relacionamentos:

- o N:1 com Escola
- o 1:N com Oficina

- **Oficina**

- o Chave primária: IdOficina
- o Chave estrangeira: IdProfessor → Professor(IdProfessor)

Atributos:

- o IdOficina (integer)

- NomeOficina (varchar)
- TemaOficina (varchar)
- DescricaoOficina (varchar)
- CargaHorariaOficina (numeric)
- DataOficina (varchar)
- StatusOficina (varchar)
- IdProfessor (integer)

Relacionamentos:

- N:1 com Professor
- 1:N com Turma
- 1:N com Documento

### - **Turma**

- Chave primária: IdTurma
- Chave estrangeira: IdOficina → Oficina(IdOficina)

Atributos:

- IdTurma (integer)
- IdOficina (integer)
- NomeTurma (varchar)
- PeriodoTurma (varchar)
- SemestreTurma (varchar)

Relacionamentos:

- N:1 com Oficina
- 1:N com Inscricao
- 1:N com Chamada

### - **Aluno**

- Chave primária: IdAluno
- Chave estrangeira: IdTurma → Turma(IdTurma)

Atributos:

- IdAluno (integer)
- NomeAluno (varchar)
- TelefoneAluno (varchar)
- EmailAluno (varchar)
- RaAluno (varchar)
- NascimentoAluno (date)
- IdTurma (integer)

Relacionamentos:

- 1:N com Documento
- 1:N com Inscricao
- 1:N com Presenca

### - Inscricao

- Chave primária: IdInscricao
- Chaves estrangeiras:
- IdTurma → Turma(IdTurma)
- IdAluno → Aluno(IdAluno)

Atributos:

- IdInscricao (integer)
- IdTurma (integer)
- IdAluno (integer)
- StatusInscricao (boolean)

Relacionamentos:

- N:1 com Turma
- N:1 com Aluno

### - Chamada

- Chave primária: IdChamada
- Chave estrangeira: IdTurma → Turma(IdTurma)

Atributos:

- IdChamada (integer)
- IdTurma (integer)
- DataChamada (date)
- Relacionamentos:
- N:1 com Turma
- 1:N com Presenca

### - Presenca

- Chave primária composta: (IdAluno, IdChamada)
- Chaves estrangeiras:
- IdAluno → Aluno(IdAluno)
- NomeAluno → Aluno(IdAluno)
- IdChamada → Chamada(IdChamada)

Atributos:

- IdAluno (integer)

- NomeAluno (varchar)
- IdChamada (integer)
- Presente (boolean)

Relacionamentos:

- N:1 com Aluno
- N:1 com Chamada

### - Documento

- Chave primária: IdDocumento
- Chave estrangeiras:
  - IdOficina → Oficina(IdOficina)
  - NomeOficina → Oficina(IdOficina)
  - TemaOficina → Oficina(IdOficina)
  - IdEscola → Escola(IdEscola)
  - IdAluno → Aluno(IdAluno)
  - IdProfessor → Professor(IdProfessor)

Atributos:

- IdDocumento (integer)
- IdOficina (integer)
- IdEscola (integer)
- idProfessor (integer)
- IdAluno (integer)
- TipoDocumento (varchar)
- StatusDocumento (varchar)
- Emissao (date)
- NomeOficina (varchar)
- TemaOficina (varchar)

Relacionamentos:

- N:1 com Oficina
- N:1 com Escola
- N:1 com Aluno
- N:1 com Professor

### - OficinaTutor

- Chave primária composta: (IdOficina, IdProfessor)
- Chaves estrangeiras:
  - IdOficina → Oficina(IdOficina)
  - IdProfessor → Professor(IdProfessor)

Atributos:

- IdOficina (integer)
- IdProfessor(integer)

Relacionamentos:

- N:1 com Oficina
- N:1 com Professor

#### - Dashboard

- TotalEscolas (integer)
- TotalProfesssores (integer)
- TotalAlunos (integer)
- TotalOficinas (integer)
- TotalOficinasEmAndamento (integer)

### 2.3 Funcionalidades do sistema

- **Gestão de Pessoas:** Cadastrar, editar, consultar e excluir professores (podendo também serem tutores e representantes), e alunos.
- **Gestão de Oficinas:** Criar novas oficinas, definir seus temas, professores responsáveis, tutores, carga horária e datas.
- **Gerenciamento de Inscrições:** Inscrever alunos em oficinas, seja individualmente ou em lote (por turma).
- **Controle de Frequência:** Registrar a presença dos alunos em cada encontro da oficina para calcular a frequência final.
- **Emissão de Certificados:** Gerar automaticamente certificados digitais para os alunos que atingirem os critérios de conclusão e cartas convite e termos de convênio.
- **Gestão de Escolas:** Cadastrar, editar e consultar informações das escolas parceiras.
- **Gestão de Turmas:** Cadastrar as turmas de cada oficina que estão envolvidas no projeto.
- **Controle de Atendimento:** Gerar relatórios para visualizar o número de alunos atendidos por escola, turma ou período.

## 3. Requisitos funcionais

|             |  |           |
|-------------|--|-----------|
| <b>RF01</b> | O sistema deve permitir o cadastro de Escolas.                       | Essencial |
| <b>RF02</b> | O sistema deve permitir a listagem de Escolas cadastradas.           | Essencial |
| <b>RF03</b> | O sistema deve permitir a edição de Escolas cadastradas.             | Essencial |
| <b>RF04</b> | O sistema deve permitir a exclusão de Escolas cadastradas.           | Essencial |
| <b>RF05</b> | O sistema deve permitir o cadastro de Alunos.                        | Essencial |
| <b>RF06</b> | O sistema deve permitir a listagem de todos os Alunos cadastrados.   | Essencial |
| <b>RF07</b> | O sistema deve permitir a edição de dados de Alunos.                 | Essencial |
| <b>RF08</b> | O sistema deve permitir a exclusão de Alunos cadastrados.            | Essencial |
| <b>RF09</b> | O sistema deve permitir o cadastro de Professores.                   | Essencial |
| <b>RF10</b> | O sistema deve permitir a edição de dados de Professores.            | Essencial |
| <b>RF11</b> | O sistema deve permitir a listagem de Professores cadastrados.       | Essencial |
| <b>RF12</b> | O sistema deve permitir a exclusão de Professores cadastrados.       | Essencial |
| <b>RF13</b> | O sistema deve permitir o cadastro de Oficinas.                      | Essencial |
| <b>RF14</b> | O sistema deve permitir a listagem de todas as Oficinas cadastradas. | Essencial |
| <b>RF15</b> | O sistema deve permitir a edição de dados de Oficinas.               | Essencial |
| <b>RF16</b> | O sistema deve permitir a exclusão de Oficinas.                      | Essencial |
| <b>RF17</b> | O sistema deve permitir a criação de Documentos.                     | Essencial |
| <b>RF18</b> | O sistema deve permitir a listagem de Documentos emitidos.           | Essencial |
| <b>RF19</b> | O sistema deve permitir o cadastro de Turmas.                        | Essencial |

|             |  |           |
|-------------|--|-----------|
| <b>RF20</b> | O sistema deve permitir a edição de Turmas.  | Essencial |
| <b>RF21</b> | O sistema deve permitir a listagem de Turmas cadastradas.  | Essencial |
| <b>RF22</b> | O sistema deve permitir a exclusão de Turmas cadastradas.  | Essencial |
| <b>RF23</b> | O sistema deve permitir o registro de Chamadas, vinculando Alunos a Presença em Oficinas.  | Essencial |
| <b>RF24</b> | O sistema deve permitir a listagem de Chamadas por Oficina.  | Essencial |
| <b>RF25</b> | O sistema deve permitir associar Professor a uma Escola.   | Essencial |
| <b>RF26</b> | O sistema deve permitir atribuir cargo ao Professor (Representante).   | Essencial |
| <b>RF27</b> | O sistema deve permitir associar um Professor como responsável da Oficina.   | Essencial |
| <b>RF28</b> | O sistema deve permitir incluir Professores como tutores de Oficina.   | Essencial |
| <b>RF29</b> | O sistema deve permitir excluir Professores como tutores de Oficina.   | Essencial |
| <b>RF30</b> | O sistema deve permitir que Oficinas tenham Turmas.  | Essencial |
| <b>RF31</b> | O sistema deve permitir que Turmas tenham inscrições de Alunos.  | Essencial |
| <b>RF32</b> | O sistema deve permitir que os Professores e Tutores da Oficina alterem o status de cada Aluno Inscrito nas Turmas de suas Oficinas. | Desejável |
| <b>RF33</b> | O sistema deve permitir que as Turmas criem Chamadas com data.   | Desejável |

|             |  |           |
|-------------|--|-----------|
| <b>RF34</b> | O sistema deve permitir que Chamadas tenham as presenças de todos Alunos na Turma.                         | Desejável |
| <b>RF35</b> | O sistema deve permitir a criação de Documentos do tipo Convite, Termos, Convênio, Presença e Certificado. | Desejável |
| <b>RF36</b> | O sistema deve permitir a associação de uma Escola ao Documento  | Desejável |
| <b>RF37</b> | O sistema deve permitir a associação de um Aluno ao Documento  | Essencial |

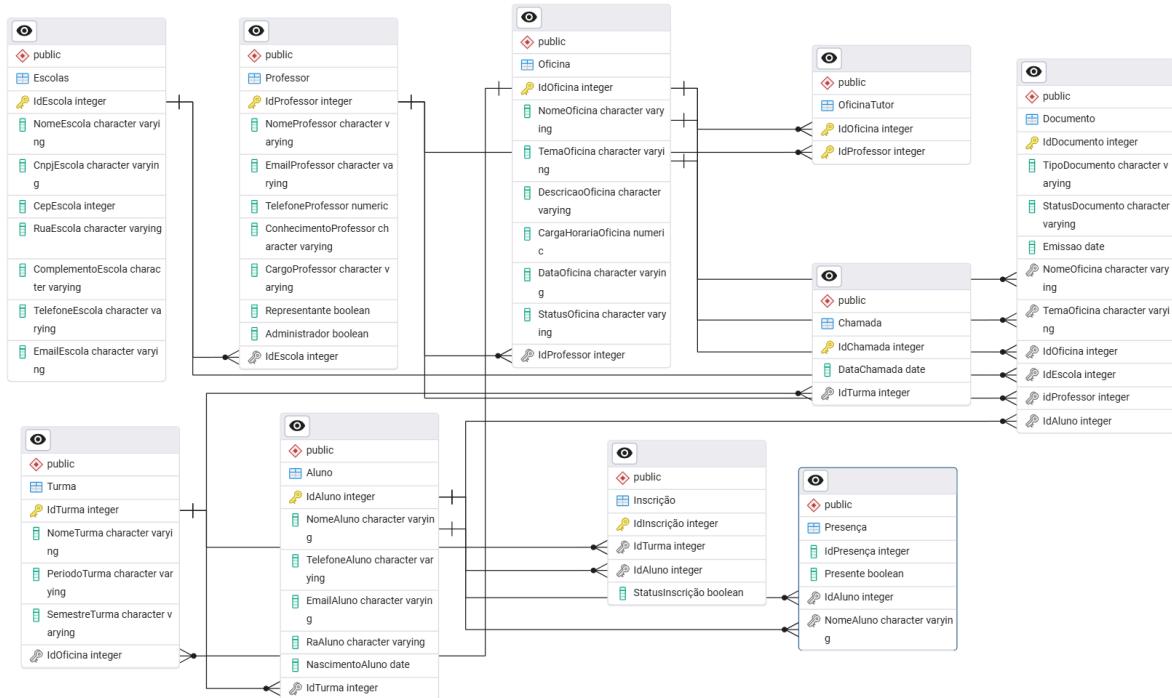
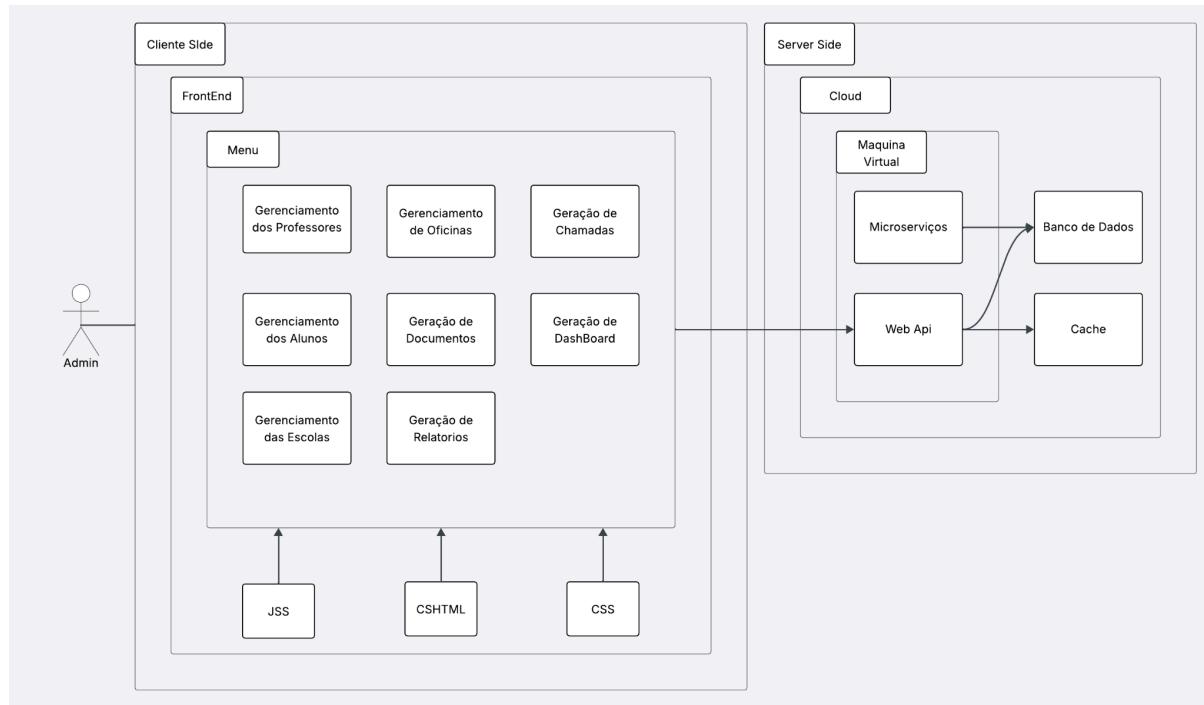
#### Detalhes dos requisitos funcionais:

- **RF01-Cadastro de Escolas:** O sistema deve permitir que os dados de Escolas possam ser registrados.
- **RF02-Listagem de Escolas:** O sistema deve permitir que os dados de Escolas possam ser listados.
- **RF03-Edição de Escolas:** O sistema deve permitir que os dados de Escolas possam ser editados.
- **RF04-Exclusão de Escolas:** O sistema deve permitir que os dados de Escolas possam ser excluídos.
- **RF05-Cadastro de Alunos:** O sistema deve permitir que os Alunos possam registrar seus dados
- **RF06-Listagem de Alunos:** O sistema deve permitir que os dados de Alunos possam ser listados.
- **RF07-Edição de Alunos:** O sistema deve permitir que os dados de Alunos possam ser editados.
- **RF08-Exclusão de Alunos:** O sistema deve permitir que os dados de Alunos possam ser excluídos.
- **RF09-Cadastro de Professores:** O sistema deve permitir que os Professores possam registrar seus dados.
- **RF10-Edição de Professores:** O sistema deve permitir a edição dos dados dos Professores cadastrados.

- **RF11-Listagem de Professores:** O sistema deve permitir que os Professores cadastrados possam ser listados.
- **RF12-Exclusão de Professores:** O sistema deve permitir a exclusão de Professores cadastrados.
- **RF13-Cadastro de Oficinas:** O sistema deve permitir o cadastro de Oficinas.
- **RF14-Listagem de Oficinas:** O sistema deve permitir que as Oficinas cadastradas possam ser listadas.
- **RF15-Edição de Oficinas:** O sistema deve permitir a edição dos dados das Oficinas cadastradas.
- **RF16-Exclusão de Oficinas:** O sistema deve permitir a exclusão de Oficinas cadastradas.
- **RF17-Geração de Documentos:** O sistema deve permitir a criação de Documentos.
- **RF18-Listagem de Documentos:** O sistema deve permitir a listagem de Documentos gerados.
- **RF19-Cadastro de Turmas:** O sistema deve permitir o Cadastro de Turmas.
- **RF20-Edição de Turmas:** O sistema deve permitir a edição dos dados das Turmas cadastradas.
- **RF21-Listagem de Turmas:** O sistema deve permitir a listagem de Turmas cadastradas.
- **RF22-Listagem de Turmas:** O sistema deve permitir a exclusão de Turmas cadastradas.
- **RF23-Geração de Chamadas:** O sistema deve permitir o registro de Chamadas, vinculando Alunos a Presença em Oficinas.
- **RF24-Listagem de Chamadas:** O sistema deve permitir a listagem de Chamadas.
- **RF25-Associação de Professores com uma Escola:** O sistema deve permitir a associação de Professores a uma Escola numa relação N:1.
- **RF26-Atribuição de cargo:** O sistema deve permitir a atribuição de cargos aos Professores cadastrados.
- **RF27-Associação de Professor com uma Oficina:** O sistema deve permitir que um Professor seja associado como responsável por uma Oficina numa relação 1:N.

- **RF28-Inclusão de Tutores:** O sistema deve permitir que Professores possam ser incluídos em Oficinas como Tutores.
- **RF29-Exclusão de Tutores:** O sistema deve permitir que Professores possam ser excluídos de Oficinas.
- **RF30-Associação de Oficinas com Turmas:** O sistema deve permitir a associação de Turmas com Oficinas em uma relação N:1.
- **RF31-Associação de Turmas com Alunos:** O sistema deve permitir a associação de Turmas com Alunos através da tabela de Inscrição.
- **RF32-Alteração de Status de Aluno pelos Professore:** O sistema deve permitir que Professores possam alterar o status de Alunos cadastrados em suas Oficinas
- **RF33-Criação de Chamadas:** O sistema deve permitir que Turmas possam criar Chamadas com data.
- **RF34-Chamadas com Presença:** O sistema deve permitir que as Chamadas registrem todas as presenças dos Alunos cadastrados em suas Turmas.
- **RF35-Criação de Documentos Prefeitos:** O sistema deve permitir a criação de Documentos do tipo Convite, Termos, Convênio, Presença e Certificado.
- **RF36-Associação de Escola e Documento:** O sistema deve permitir a associação de Escolas com Documentos em uma relação 1:N.
- **RF36-Associação de Aluno e Documento:** O sistema deve permitir a associação de Aluno com Documentos em uma relação 1:N.

## 4. Arquitetura em alto nível do sistema



## 5. Estratégia de automação de testes do sistema

A estratégia ideal segue a pirâmide de testes, que organiza os tipos de testes por profundidade e custo de execução:

| Tipo de Teste    | Objetivo   | Ferramentas                                  |
|------------------|--|--|
| Unitário         | Validar métodos e classes isoladamente           | xUnit, NUnit                                 |
| Integração       | Testar comunicação entre componentes             | Microsoft.AspNetCore.Mvc.Testing, TestServer |
| End-to-End (E2E) | Simular o fluxo completo do usuário na aplicação | Selenium                                     |

### 1. Testes Unitários

- Foco na camada de Model e Controller
- Cobertura de regras de negócio, validações e lógica de fluxo
- Mocking de dependências com bibliotecas como Moq

### 2. Testes de Integração

- Validação da interação entre Controllers e Services
- Testes com banco de dados em memória (ex: SQLite in-memory)
- Verificação de rotas, respostas HTTP e status codes

### 3. Testes E2E

- Automação da interface HTML/CSS com simulação de usuário
- Verificação de elementos visuais, formulários e navegação
- Execução em múltiplos navegadores com Selenium WebDriver

### Integração com CI/CD

- Configurar pipelines no GitHub Actions, Azure DevOps ou GitLab CI
- Executar testes automaticamente a cada commit ou pull request
- Gerar relatórios de cobertura e falhas para análise contínua

## 6. Tecnologias utilizadas no projeto

### • Gerenciamento e Colaboração

**Trello:** Plataforma de gerenciamento de projetos e tarefas. Permite organizar fluxos de trabalho, atribuir responsabilidades e acompanhar o progresso em tempo real.

**Meet (Google Meet):** Ferramenta de videoconferência online da Google, usada para reuniões e chamadas em grupo.

**Google Drive:** Serviço de armazenamento em nuvem que permite guardar, compartilhar e colaborar em arquivos.

### • Desenvolvimento e Programação

**Visual Studio:** Ambiente de desenvolvimento integrado (IDE) completo, especialmente usado para desenvolvimento em C#.

**Visual Studio Code (VSCode):** Editor de código leve e altamente customizável, para outras linguagens e desenvolvimento visual.

**C#:** Linguagem de programação moderna e orientada a objetos.

**JavaScript:** Linguagem de programação voltada para desenvolvimento web.

**HTML:** Linguagem de marcação usada para estruturar páginas da web.

**CSS:** Linguagem usada para estilizar elementos HTML.

**SQL Server:** Sistema de gerenciamento de banco de dados.

### • Controle de Versão

**GitHub:** Plataforma de hospedagem de código que usa Git para controle de versão. Permite colaboração em projetos, controle de alterações e integração contínua.

## 7. Cronograma

| Data          | Objetivos  |
|---------------|--|
| 15/09 à 20/09 | Início da configuração<br>Implementação da arquitetura e execução do projeto na IDE (Visual Studio).<br>Download dos pacotes a serem implementados / Testes. |
| 21/09 à 27/09 | Criação e relacionamento de banco de dados / Testes.   |
| 28/09 à 04/10 | Criação das classes e domínios na IDE.   |
| 05/10 à 11/10 | Criação da API a ser utilizada, com nomeação das chamadas públicas e autorização dos usuários / Testes.  |
| 12/10 à 18/10 | Criação das controllers e funções a serem utilizadas no projeto / Testes.  |
| 19/10 à 25/10 | Review de backlog e correção de problemas.   |
| 26/10 à 01/11 | Definição da view de Dashboard.<br>Entrega da primeira sprint.   |
| 02/11 à 08/11 | Review da sprint.<br>Atualização de cronograma.<br>Definição das views de cadastros.   |
| 09/11 à 15/11 | Implementação das views para gerar o certificado/ Testes.  |
| 16/11 à 22/11 | Criação dos modelos de relatórios a serem implementados.   |
| 23/11 à 29/11 | Implementação das views e do modelo de negócio dos relatórios / Testes.  |
| 30/11 à 06/12 | Elaboração de testes em escala e níveis mais elaborados.   |
| 07/12 à 12/12 | Review geral do backlog do projeto.<br>Entrega da segunda sprint.  |

