**Sprint 1 (Planejamento inicial, Infraestrutura e Preparação para Backend e Frontend)**

**Objetivos principais:**

* Iniciar a infraestrutura básica do sistema (estrutura do backend e banco de dados).
* Iniciar o desenvolvimento do frontend com prototipação e responsividade inicial.
* Garantir que as bases de dados e a estrutura do backend estejam configuradas.

**Histórias a serem realizadas:**

1. **História 2 – Desenvolvimento do Front-End- Design e Responsividade**

* Criar as telas transformando layout do figma em código.
* Garantir layout adaptável (CSS puro, sem frameworks).
* Ajustar elementos para telas menores.
* Testes em dispositivos diferentes.

1. **História 3– Desenvolvimento do Front-End - Grade de Horários**

* Garantir layout adaptável (CSS puro, sem frameworks).
* Endpoint de consulta no backend
* Tela de manutenção da grade.
* Criar interface de filtro e busca.
* Tela de somente exibição da grade.
* Exibir resultados em formato de tabela responsiva.
* Ajustar elementos para telas menores.
* Testes em dispositivos diferentes.

1. **História 4– Desenvolvimento do Front-End - Mapa Interativo**

* Garantir layout adaptável (CSS puro, sem frameworks).
* Criar interface de filtro e busca
* Endpoint de consulta no backend
* Tela de manutenção do mapa.
* Exibir resultados em formato de mapa responsivo
* Tela de somente exibição do mapa.
* Ajustar elementos para telas menores.
* Testes em dispositivos diferentes.

1. **História 5 – Desenvolvimento do Front-End - Usabilidade**

* Desenvolver lógicas para consumir os dados coletados nos end-points da API.
* Aplicar métodos de segurança no front-end

1. **História 12- Upload de CSV e Processamento**

* Criar endpoint para upload de arquivos CSV.
* Implementar lógica de leitura e validação de arquivos CSV:
  1. Verificar formato do arquivo.
  2. Validar campos obrigatórios (ex.: semestre, periodo).
* Salvar dados válidos no banco de dados.
* Retornar feedback ao usuário:
  1. Quantidade de registros processados com sucesso.
  2. Lista de erros encontrados.

1. **História 13 - Validação de Regras de Negócio**

* Desenvolver lógica de validação para:
  1. Conflito de horários de professores.
  2. Duplicidade de ambiente.
  3. Horários sobrepostos para a mesma turma.
* Implementar sistema para listar pendências identificadas.
* Salvar parcialmente dados válidos e marcar pendências para correção posterior.

1. **História 14 - Segurança**

* Configurar autenticação utilizando JWT.
* Implementar middleware de autorização para proteger rotas sensíveis.
* Criar sistema de roles (ex.: administrador, usuário padrão) para restringir ações específicas.
* Adicionar controle de sessões para expiração de tokens.

1. **História 15 – API Restful**

* Definir os endpoints para operações CRUD de usuários
* Implementar o padrão de requisições HTTP (GET, POST, PUT, DELETE)
* Desenvolver a estrutura de resposta em formato JSON
* Configurar autenticação via token JWT
* Implementar tratamento de erros e códigos de status HTTP adequados
* Realizar testes de integração dos endpoints

1. **História 16 – Deploy Front-End**

* Preparar o projeto para build de produção
* Configurar variáveis de ambiente para o ambiente de produção
* Otimizar assets (imagens, CSS, JavaScript) para melhor performance
* Realizar configuração de HTTPS para comunicação segura
* Implementar monitoramento e alertas de disponibilidade
* Configurar pipeline de CI/CD para automatizar o processo de deploy

1. **História 17- Deploy Back-End**

* Configurar ambiente de produção no servidor/cloud
* Preparar o projeto para build de produção
* Configurar variáveis de ambiente para produção
* Implementar conexão com o banco de dados de produção
* Implementar monitoramento de recursos e performance
* Configurar backup automático do banco de dados
* Implementar pipeline de CI/CD para automatizar o deploy
* Configurar HTTPS e certificados SSL

1. **História 18- Deploy Banco de Dados**

* Configurar servidor/instância de banco de dados no ambiente de produção
* Criar scripts de migração para estrutura do banco de dados
* Implementar estratégia de versionamento do esquema do banco
* Configurar usuários e permissões com privilégios mínimos necessários
* Implementar mecanismos de backup e recuperação automatizados
* Configurar replicação para alta disponibilidade
* Otimizar índices e consultas para melhor performance

**Critérios de Aceite da Sprint 2:**

* Front-end responsivo e utilizável
* Grade de horários funcional e estilizada
* Mapa interativo estilizado
* Front-end consumindo API (Back-End)
* Upload do CSV em funcionamento, inserindo no banco
* Regras de negócio de acordo com as entidades
* Segurança através de JWT em funcionamento
* API Restful (Back-End) em funcionamento
* Front-end rodando na nuvem e sendo consumido
* Back-end rodando na nuvem e sendo consumido
* Banco de dados rodando na nuvem e sendo consumido