📝 Product Backlog - Sistema de Visualização de Horários

**Épico:** Sistema Web para Gerenciamento e Consulta de Horários e Visualização de Ambientes Acadêmicos

**História 1 - Prototipação (Figma)**

Como time de desenvolvimento,

Quero ter uma prototipagem validada no Figma,

Para garantir uma interface intuitiva antes da codificação.

**Tarefas:**

* Criar Wire frames detalhados para as principais páginas do sistema no Figma.
* Definição de Cor primária e secundária do projeto.
* Validar com o cliente/usuário.
* Refinar até aprovação final.

**Prioridade: Alta**

**Critérios de Aceite:**

* Wire frames contemplam todas as funcionalidades solicitadas.
* Todos os fluxos principais do sistema são representados no protótipo.
* Aprovar o protótipo antes do desenvolvimento do front-end.

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

Como usuário,

Quero que o sistema funcione bem em dispositivos móveis e desktop,

Para ter uma boa experiência em qualquer contexto.

**Tarefas:**

* 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.

**Prioridade: Alta**

**Critérios de Aceite:**

* Sistema é utilizável em smartphones e tablets.
* Layout não quebra em diferentes tamanhos de tela.
* Sistema navega entre as telas corretamente.

**História 3 – Desenvolvimento do Front-End - Usabilidade**

Como desenvolvedor Front-End,

Quero que o sistema se comunique com o Back-End e esteja bem estruturado,

Para consumir os dados do banco e tratá-los para exibir ao usuário.

**Tarefas:**

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

**Prioridade: Alta**

**Critérios de Aceite:**

* Sistema com lógicas para consumo dos end-points funcionando.
* Sistema com validações de segurança mínimas.

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

RNF01

Como secretaria acadêmica,

Quero que o sistema possua um mapa interativo no qual exibe,

Mapa de salas com a opção interativa de clicar e visualizar a situação

Do ambiente (ex. aulas alocadas ao longo dos turnos do dia).

**Tarefas:**

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

**Prioridade: Alta**

**Critérios de Aceite:**

* Filtros por turma, professor, turno e ambiente funcionam corretamente
* Resultados atualizados dinamicamente
* Mapa é utilizável em smartphones e tablets.
* Layout não quebra em diferentes tamanhos de tela.
* Sistema navega entre os diferentes mapas corretamente.

**História 5 - Interatividade de Ambientes**

RNF01

Como usuário,

Quero ver um mapa das salas e clicar em cada sala para ver as alocações do dia,

Para ter uma visão clara e visual da ocupação da faculdade.

**Tarefas:**

* Implementar interação via click (mostrar horários).
* Exibir status (livre/ocupada) no mapa.
* Configurar o mapa para receber informações do back-end para a interação.

**Prioridade: Alta**

**Critérios de Aceite:**

* Mapa exibe status visual de ocupação.
* Clique em cada ambiente abre painel com horários detalhados.
* Mapa está consumindo as informações do back-end corretamente.

**História 6 - Definição de estrutura BackEnd**

Como time de desenvolvimento,

Quero ter uma estrutura de como vai ser o projeto do servidor,

Para garantir um código organizado e robusto para fácil entendimento e manutenção.

**Tarefas:**

* Definir o design do projeto (Estrutura das Pastas).
* Escolher a Stack do back-end (ex.: Node.js com Express).
* Configurar o ambiente inicial de desenvolvimento.
* Configurar variáveis de ambiente (ex.: dotenv).
* Configurar um servidor básico que responde "OK" na rota raiz.

**Prioridade: Alta**

**Critérios de Aceite:**

* Tecnologias bem definidas.
* Design do projeto organizado e estrutura coesa.
* O projeto está configurado com as ferramentas escolhidas e o ambiente é executável.
* A rota raiz (GET/) responde com status 200 e uma mensagem de confirmação.
* Estrutura de pastas está organizada para suportar escalabilidade.

**História 7 - Banco de Dados**

Como desenvolvedor,

Quero modelar e configurar o banco de dados,

Para armazenar os dados necessários ao sistema de forma eficiente e organizada.

**Tarefas:**

* Escolher o banco de dados relacional (PostgreSQL).
* Modelar e criar as tabelas iniciais:
  1. Semestre
  2. Cursos
  3. Período (Horários e dias)
  4. Matérias
  5. Docentes
  6. Ambientes
* Adicionar migrações para facilitar mudanças futuras no banco.
* Testar a conexão com o banco e garantir que as tabelas foram criadas corretamente.

**Prioridade: Alta**

**Critérios de Aceite:**

* Banco de dados configurado e acessível pelo back-end.
* Tabelas necessárias criadas conforme especificado na modelagem.
* Conexão testada e funcional.

**História 8 - Endpoints CRUD**

RF02

Como desenvolvedor Back-End,

Quero criar endpoints para operações CRUD de semestres, cursos, períodos (horários e dias), matérias, docentes e ambientes,

Para permitir a manipulação desses dados pelo sistema.

**Tarefas:**

* Criar rotas para cada entidade:
  1. GET /entidade (Listar)
  2. POST /entidade (Criar)
  3. PUT /entidade/:id (Atualizar)
  4. DELETE /entidade/:id (Excluir)
* Implementar controladores para cada operação.
* Validar campos obrigatórios antes de salvar ou atualizar registros.
* Retornar mensagens de erro e códigos de status apropriados.

**Prioridade: Alta**

**Critérios de Aceite:**

* Endpoints CRUD estão implementados e integrados ao banco de dados.
* Operações retornam respostas adequadas (ex.: 201 para criação, 400 para erros de validação).
* Validações de dados garantem integridade mínima das informações.

**História 9 - Exportação em PDF**

RF04

Como secretaria acadêmica,

Quero gerar um PDF com os horários e alocações,

Para facilitar a impressão ou o envio digital do cronograma.

**Tarefas:**

* Criar endpoint de geração de PDF no backend
* Definir layout do PDF (mapa e tabelas)
* Botão de exportar na interface de consulta

**Prioridade: Média**

**Critérios de Aceite:**

* PDF gerado contém todas as informações necessárias
* Layout legível e pronto para impressão

**História 10 - Upload de CSV e Processamento**

RF01

Como secretaria acadêmica,

Quero fazer upload de arquivos CSV com dados do sistema,

Para alimentar rapidamente o banco de dados com informações em massa

**Tarefas:**

* 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.

**Prioridade: Média**

**Critérios de Aceite:**

* O sistema aceita arquivos CSV com estrutura válida.
* Dados válidos são salvos no banco; inválidos são listados como erros.
* Feedback detalhado sobre o processamento é exibido.

**História 11 - Validação de Regras de Negócio**

RF03

Como secretaria acadêmica,

Quero validar as regras de alocação de horários,

Para evitar conflitos como professores em dois lugares ao mesmo tempo.

**Tarefas:**

* 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.

**Prioridade: Alta**

**Critérios de Aceite:**

* Sistema bloqueia alocações inválidas e lista erros para revisão.
* Pendências são salvas no sistema para correção futura.
* Dados válidos são armazenados sem interrupções.

**História 12 - Segurança**

Como desenvolvedor,

Quero implementar autenticação e autorização no sistema,

Para proteger endpoints e garantir que apenas usuários autorizados tenham acesso.

**Tarefas:**

* 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.

**Prioridade: Alta**

**Critérios de Aceite:**

* Endpoints protegidos por autenticação e autorização.
* Tokens JWT são gerados e validados corretamente.
* Ações são restritas de acordo com o nível de permissão do usuário.

**História 13 - Documentação**

Como desenvolvedor,

Quero documentar a API,

Para garantir que o sistema seja confiável e fácil de usar por outros times.

**Tarefas:**

* Criar documentação para os endpoints utilizando Insominia ou Postman.

**Prioridade: Média**

**Critérios de Aceite:**

* Documentação dos endpoints está acessível e detalhada.