Gabriel Ferreira Dantas
Igor da Silva Sant'anna
Ronivaldo Domingues de Andrade

Relatório Final do Projeto – MedSuam

Rio de Janeiro – RJ
2025

Gabriel Ferreira Dantas
Igor da Silva Sant'anna
Ronivaldo Domingues de Andrade

Relatório Final do Projeto – MedSuam

Centro Universitário Augusto Motta – UNISUAM
Curso Superior de Análise e Desenvolvimento de Sistemas
Projeto de Desenvolvimento em Back-End

Rio de Janeiro – RJ
2025

SUMÁRIO

1 INTRODUÇÃO

O presente relatório tem como finalidade documentar, de forma clara e estruturada, o desenvolvimento das quatro sprints que compõem o projeto realizado na disciplina de Projeto de Back-end II. Cada sprint teve como objetivo a construção progressiva de funcionalidades essenciais para um sistema seguro, escalável e alinhado às boas práticas de desenvolvimento.

Na **Sprint 1**, foi estabelecida a base do sistema, incluindo a modelagem do banco de dados, o cadastro de usuários, o login e as primeiras validações de segurança. Essa etapa preparou a infraestrutura fundamental sobre a qual as demais funcionalidades foram construídas.

A **Sprint 2** concentrou-se no desenvolvimento de um módulo administrativo completo. Nessa fase, foram implementados níveis hierárquicos de permissão (Admin e Super Admin), além de um CRUD robusto para gerenciamento de usuários, garantindo controle e segurança sobre operações administrativas.

A **Sprint 3** ampliou significativamente a segurança do sistema, introduzindo a autenticação em dois fatores (2FA), desenvolvida por meio da integração com a API SendGrid. Essa funcionalidade adicionou uma camada verificadora adicional ao processo de login, reforçando a confiabilidade e proteção dos acessos.

Por fim, a **Sprint 4** abordou o fluxo completo de redefinição de senha, envolvendo validações front-end, envio de códigos de verificação, tratamento seguro de sessões e atualização criptografada de credenciais.

Assim, este relatório apresenta detalhadamente os objetivos, as atividades realizadas, as funcionalidades implementadas e os resultados alcançados em cada sprint, seguido de uma conclusão que sintetiza os aprendizados e a evolução técnica adquirida durante o desenvolvimento do projeto.

2 SPRINT 1 ESTRUTURA INICIAL DO SISTEMA DE AUTENTICAÇÃO

2.1 Objetivos

A Sprint 1 teve como objetivo construir a base do sistema de autenticação, incluindo o cadastro de usuários (pacientes e médicos), login funcional e preparação da estrutura necessária para recuperação de senha.

2.2 Atividades Executadas

Durante esta etapa foram utilizados PHP, MySQL, HTML, CSS, JavaScript e o banco `bd_medsuam.sql`. Seguindo orientações da disciplina, foram aplicadas boas práticas como sanitização de dados, verificação de e-mails duplicados e o uso da função `password_hash()` para proteção das senhas.

A primeira tarefa consistiu na criação e organização do banco de dados. O script inicial criou tabelas como: `paciente`, `medico`, `telefone`, `endereco`, `rg`, `especialidade`, entre outras que se fizeram necessárias. A conexão com o banco foi centralizada no arquivo `dbMedsuam.php`.

2.2.1 Cadastro de Pacientes

O cadastro de pacientes sanitiza os dados com `mysqli_real_escape_string()`, valida campos obrigatórios e bloqueia o registro caso o e-mail já exista nas tabelas de pacientes ou médicos. Após validação, o sistema insere os dados do paciente e registra endereço, telefone e RG vinculados ao seu ID.

2.2.2 Cadastro de Médicos

O cadastro de médicos segue lógica semelhante, com criptografia de senha e registro de telefone e especialidade após validações.

2.2.3 Login

O login foi implementado no arquivo `userpage.php`. As credenciais são verificadas no banco, e uma sessão segura é iniciada em caso de sucesso, permitindo acesso a áreas internas restritas.

2.3 Resultados Alcançados

A Sprint 1 permitiu criar uma base sólida para o restante do sistema. O grupo adquiriu experiência prática com criação de CRUD básico, organização de dados no banco, integração entre front-end e back-end, sanitização de dados, proteção de senhas e controle de sessões.

3 SPRINT 2 DESENVOLVIMENTO DO CRUD ADMINISTRATIVO

3.1 Objetivos

A Sprint 2 teve como objetivo desenvolver um módulo administrativo completo, permitindo que usuários com privilégios especiais gerenciassem os demais usuários do sistema.

3.2 Atividades Executadas

- Criação do diretório `admin/` na raiz do projeto para conter os códigos do Admin, fazendo com que o acesso ao painel admin seja pela rota *localhost/medsuam/admin* em caso de uso local.
- Implementação do CRUD completo (Criar, Ler, Atualizar e Deletar).
- Criação dos níveis de permissão: *Admin* e *Super Admin*.
- Desenvolvimento dos controles de acesso para cada tipo de administrador.
- Implementação da lógica de bloqueio para impedir que administradores comuns gerenciem outros admins.

3.3 Funcionalidades Implementadas

- Cadastro, listagem, edição e exclusão de usuários.
- Sistema de hierarquia:
  - **Super Admin**: acesso total.
  - **Admin**: acesso total, exceto sobre perfis administrativos.

OBS.: Não foi implementado um script específico para inserção do primeiro usuário Super Admin, nesse caso foi criado o banco de dados já com ele inserido ou inserido manualmente no PHPmyAdmin quando se necessitava de testes, a criação desse script foi deixada para uma futura implementação real do sistema.

3.4 Resultados Alcançados

O módulo administrativo foi concluído com sucesso, oferecendo controle interno robusto, seguro e totalmente funcional.

4 SPRINT 3 AUTENTICAÇÃO DE DOIS FATORES (2FA)

4.1 Objetivos

Implementar um sistema de autenticação em dois fatores para aumentar a segurança no processo de login.

4.2 Atividades Executadas

- Integração da API SendGrid para envio de códigos de verificação. A escolha do SendGrid se dá pela facilidade de implementação.
- Criação da página de verificação pós-login.
- Implementação do fluxo completo de geração, envio e validação do código.

4.3 Funcionalidades Implementadas

OBS.: Nessa parte do projeto, foi usada duas bibliotecas prontas, que foram instaladas via `composer`, são elas: a `Dotenv\Dotenv` e `SendGrid\Mail\Mail`, normalmente em casos assim o arquivo `.gitignore` deve excluir do versionamento o diretório `vendor/` por questões de tamanho do projeto, porém nesse caso não o fizemos por conta de falta de familiaridade e compatibilidade do uso da ferramenta `composer`.

- Envio automático de um código de 6 dígitos para o email informado pelo usuário.

  Em `utilsMail.php` A função `generate2FACode()` gera um código de 6 dígitos e o salva em uma variável da sessão, ou seja, o usuário já deve ter sido aprovado no processo de login ou cadastro e uma sessão para ele já deve ter sido iniciada.

  A função `send2FACode()`, recebe o parâmetro obrigatório *$email* e é responsável por enviar o e-mail com o código de 6 dígitos gerado pela função anterior, ela recupera o código atravéz da sessão aberta para o usuário, recupera a chave da API SendGrid do arquivo *.env* usando o *Dotenv* e cria um corpo básico de e-mail com assunto e corpo e o envia ao e-mail passado no parâmetro.

  A função `verify2FACode()` recebe o parâmentro obrigatório *$inputCode* que é o código digitado pelo usuário, em seu fluxo, ela recupera da variável da sessão o código original e os compara retornando *true* ou *false* ela faz outras coisas como

controlar o número máximo de tentativas permitidas e envia junto mensagens caso esse tempo seja ultrapassado. A fim de facilitar nosso trabalho e testes foi adicionado a essa função um código Bypass — 000000 para evitar que sempre necessite aguardar o e-mail. Importante destacar que tal recurso é utilizado exclusivamente para fins de teste, não sendo aplicado em ambiente de produção.

- Verificação obrigatória antes de acessar o perfil.
- Bloqueio de tentativas inválidas por tentativas e por tempo de expiração do código enviado.

4.4 Resultados Alcançados

O 2FA foi integrado com sucesso ao sistema, fortalecendo a segurança do processo de autenticação e garantindo maior confiabilidade para os usuários e muito aprendizado no desenvolvimento.

5 SPRINT 4 RESET DE SENHA

5.1 Objetivos

A Sprint 4 teve como objetivo permitir que os usuários redefinissem sua senha de forma segura, utilizando validações JavaScript e envio de códigos pelo SendGrid.

5.2 Atividades Executadas

A funcionalidade foi desenvolvida com PHP, HTML, CSS, JavaScript, MySQL e SendGrid. Foram criadas quatro páginas PHP:

5.2.1 `login.php`

Inclui o link Esqueceu sua senha?, que redireciona o usuário para `emailCollect.php`.

5.2.2 `emailCollect.php`

Nesta página o usuário informa seu e-mail. Há validação em JavaScript e, no back-end, sanitização e verificação no banco de dados. Caso seja encontrado, sessões são criadas e o usuário é redirecionado.

5.2.3 `autenticacao.php`

A API SendGrid envia um código de seis dígitos. O código inserido pelo usuário é validado. Se correto, segue para `reset.php`.

5.2.4 `reset.php`

O usuário insere sua nova senha duas vezes. Validações em JavaScript garantem que as senhas coincidam. No back-end, o sistema localiza o usuário e atualiza a nova senha já criptografada com `password_hash()`.

5.3 Resultados Alcançados

A Sprint 4 permitiu ao grupo aprender validações em JavaScript, sanitização segura, uso de APIs externas, manipulação de sessões e atualização de senhas com boas práticas de segurança.

6 CONCLUSÃO

O desenvolvimento das quatro sprints resultou em um sistema funcional, seguro e estruturado, que evoluiu progressivamente em complexidade e maturidade. Na Sprint 1, foi construída a base do sistema de autenticação, incluindo cadastro, login e organização do banco de dados. Na Sprint 2, implementou-se um CRUD administrativo robusto, com hierarquia de permissões. A Sprint 3 ampliou a segurança com autenticação de dois fatores, integrando envio de códigos por e-mail. Por fim, a Sprint 4 consolidou um processo completo e seguro de redefinição de senha.

O grupo adquiriu experiência prática em desenvolvimento back-end, integração com banco de dados, segurança da informação, uso de APIs externas, boas práticas de organização e programação, validações, controle de sessões e arquitetura do sistema. O conjunto das sprints resultou em um sistema coerente, estável e alinhado com os objetivos da disciplina.

A LINKS DOS REPOSITÓRIOS E MATERIAIS DO PROJETO

Este apêndice reúne os principais links utilizados no desenvolvimento do sistema, incluindo o repositório responsável pelo código-fonte das quatro sprints e o quadro do Trello, onde foram organizadas as atividades, etapas e responsabilidades do grupo ao longo do projeto.

A.1 Repositório MedSuam no GitHub

A seguir, encontra-se o repositório que armazena o código produzido no decorrer das fases do desenvolvimento:

- **Repositório Geral do Projeto:**
  `https://github.com/UnisuamWorkSpace/medSuam.git`

A.2 Quadro do Trello

O quadro do Trello foi utilizado para organização interna, registro das atividades, acompanhamento das tarefas por sprint e controle do fluxo de desenvolvimento. Seu acesso está disponível abaixo:

- **Trello Organização das Sprints:**
  `https://trello.com/b/ZZraZxb7/medsuam`