

Software para gestão da secretaria da Associação de Diabéticos de Santa Maria.

Vinícius Moura da Silva
Curso de Ciência da Computação
Universidade Franciscana
CEP 97010-032 – Santa Maria - RS - Brasil
v.moura@ufn.edu.br

Sylvio André Garcia Vieira
Curso de Ciência da Computação
Universidade Franciscana
CEP 97010-032 – Santa Maria - RS - Brasil
sylvio@ufn.edu.br

Resumo—A ADSM é uma Associação sem fins lucrativos que realiza o acompanhamento e auxílio aos diabéticos de Santa Maria, que necessita de um *software* para realizar a gestão de suas atividades diárias. O objetivo deste trabalho é apresentar o desenvolvimento de um sistema *Web* que permite a gestão automatizada de gerenciamento de profissionais, sócios, mensalidades e agendamento de consultas para a associação. As principais ferramentas usadas para o desenvolvimento do sistema foram HTML, CSS, Python, Django e Bootstrap.

Palavras-chave : Diabetes, Sistema, Python, Django.

I. INTRODUÇÃO

O Diabetes Mellitus (DM) é uma doença do tipo Doenças Crônicas Não-Transmissíveis (DCNT), uma doença caracterizada pelo comprometimento do metabolismo da glicose, resultando em hiperglicemia crônica [1].

Esse tipo de doença tende a transformar de forma intensa a vida de quem a possui, de acordo com Petermann et al. [1]:

“Tais doenças estão relacionadas às Atividades de Vida Diária (AVD) e, desde o estabelecimento do diagnóstico, podem produzir sentimentos de desespero e aflição, diminuindo a potência do indivíduo para agir e pensar. Por isso, essa situação impõe a necessidade de um cuidado integral à saúde, o que implica uma nova atitude, ou seja, as pessoas com doenças crônicas deverão aprender a gerenciar suas vidas em um processo longitudinal que objetive qualidade e autonomia”.

Segundo o IDF (*International Diabetes Federation*) em pesquisa realizada em 2021, existiam cerca de 537 milhões de adultos (entre 20-79 anos) vivendo com diabetes no mundo, no Brasil o número de adultos que convivem com o Diabetes era de aproximadamente 15,7 milhões, o que classificou o país em 6º lugar no ranking de maior número de casos no mundo [2].

O tratamento de doenças, crônicas ou não, pode ser auxiliado por sistemas computacionais, tanto na definição de diagnósticos quanto nas tarefas mais simples, como gerenciamento de consultas e prescrição de medicações de forma eletrônica.

Marin [3], define Sistemas de Informação em Saúde (SIS) como:

"[...] um conjunto de componentes inter-relacionados que coletam, processam, armazenam e distribuem a informação para

apoiar o processo de tomada de decisão e auxiliar no controle das organizações de saúde.”

Desta maneira, o SIS colabora para o planejamento, aperfeiçoamento e tomada de decisões dos profissionais na área da saúde, envolvendo pacientes e usuários do sistema [3].

Em Santa Maria, existe uma associação sem fins lucrativos que auxilia pessoas com diabetes, a ADSM (Associação dos Diabéticos de Santa Maria). Esta Associação realiza seus agendamentos de consultas e controle financeiro dos sócios sem o uso de um sistema de informação. Dessa forma, um sistema computacional poderá contribuir para um melhor atendimento aos pacientes portadores de diabetes que recebem a assistência da ADSM.

O desenvolvimento do sistema contou com o apoio do Laboratório de Práticas Computação da Universidade Franciscana (UFN).

A. Objetivos Gerais

Desenvolver uma aplicação *web* para gerenciar de profissionais, sócios, mensalidades, pagamentos e agendamentos de consultas para a ADSM (Associação dos Diabéticos de Santa Maria).

B. Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Utilizar a linguagem de programação Python, os *frameworks* Django e Bootstrap para o desenvolvimento da aplicação;
- Utilizar o *Scrum* para obter melhores resultados e efetividade durante o desenvolvimento;
- Analisar o funcionamento de Sistema para Clínicas;
- Desenvolver um módulo do sistema que permita a realização de cadastros dos sócios e profissionais, distintos por função;
- Desenvolver um módulo de gerenciamento financeiro;
- Construir um módulo de agendamento de consultas;

II. REFERENCIAL TEÓRICO

Nesta seção, serão apresentados revisões bibliográficas e conceitos que irão contribuir para o trabalho. Assim como as tecnologias que serão utilizadas para o desenvolvimento.

A. *Diabetes Mellitus*

De acordo com Ministério da Saúde [4] o conceito do Diabetes Mellitus

"É um grupo de doenças metabólicas caracterizadas por hiperglicemia e associadas a complicações, disfunções e insuficiência de vários órgãos, especialmente olhos, rins, nervos, cérebro, coração e vasos sanguíneos."

Ainda pode resultar de defeitos de secreção ou da insulina que envolvem processos patogênicos específicos, como: destruição das células beta do pâncreas, resistência à insulina, entre outros [4].

Os tipos mais frequentes de diabetes conforme o Ministério da Saúde [4] são:

"Os tipos de diabetes mais frequentes são o diabetes tipo 1, anteriormente conhecido como diabetes juvenil, que compreende cerca de 10% do total de casos, e o diabetes tipo 2, anteriormente conhecido como diabetes do adulto, que compreende cerca de 90% do total de casos. Outro tipo de diabetes encontrado com maior frequência e cuja etiologia ainda não está esclarecida é o diabetes gestacional, que, em geral, é um estágio pré-clínico de diabetes, detectado no rastreamento pré-natal."

O diabetes tipo 1, normalmente surge na infância ou adolescência, podendo ser diagnosticado quando adulto em alguns casos [5]. Ocorre quando o sistema imunológico ataca equivocadamente as células beta, dessa forma é liberada pouca insulina para o corpo, como consequência a glicose fica no sangue ao invés de ser utilizada como energia no corpo [5].

Alguns dos sintomas do diabetes tipo 1 são: vontade de urinar diversas vezes, fome frequente, sede constante, perda de peso, fadiga, náusea e vômito [6].

O tipo 2 se manifesta na maioria dos casos nos adultos, porém crianças também podem apresentar [5]. Ocorre quando o organismo não é capaz de usar adequadamente a insulina que produz, ou não produz insulina suficiente por controlar a taxa de glicemia [5]. Pode ser controlada por meio de atividade física e planejamento alimentar, mas alguns casos necessita o uso de insulina ou medicamentos para controle da glicose [5].

Os sintomas do diabetes tipo 2 são: infecções frequentes, alteração visual, dificuldade na cicatrização de feridas, formigamento no pés e furúnculos [6].

O diabetes é uma das principais causas de mortalidade, insuficiência, cegueira e doença cardiovascular, além de acarretar custos financeiros e custos associados à dor, ansiedade e menor qualidade de vida, afetando tanto os doentes quanto suas famílias [4].

B. *Sistemas para clínicas*

De acordo com Câneo e Rondina [7], o surgimento de novas gerações de sistemas de informação hospitalar, tem como objetivo otimizar e qualificar o atendimento, reduzir os custos, além de reunir informações para compor um perfil de saúde em determinadas regiões.

Desse modo o Prontuário Eletrônico do Paciente (PEP), é uma das principais ferramentas usadas pelos médicos e suas equipes no trabalho diário [7] [8].

O PEP é um Sistema de Registro Eletrônico de Saúde (SRES) para registro de informações referentes à assistência à saúde de indivíduos, tendo como finalidade construir um prontuário do paciente no formato eletrônico [8]. Sendo assim a PEP é constituída por 4 modalidades: Consultório Individual, Clínicas/Ambulatório, Internação e Pronto Atendimento [8].

Para Massad et al. (2003 apud [9])

"O prontuário eletrônico é um meio físico, um repositório onde todas as informações de saúde, clínicas e administrativas, ao longo da vida de um indivíduo estão armazenadas".

Segundo Abrahão (2003 apud [9]), além das facilidades de armazenamento, essas novas tecnologias da informação possibilitam que os dados possam ser transmitidos, publicados, facilitando as trocas eletrônicas de informações entre médico e paciente, entretanto podem afetar a segurança e privacidade das informações.

C. *Astah UML*

O Astah UML é uma ferramenta projetada para o desenvolvimento especificamente para UML (Unified Modeling Language), simples de aprender, simples de utilizar, permitindo o usuário criar diagramas UML [10].

Possui a opção de gerar código diretamente do diagrama utilizando as ferramentas do software, assim como a possibilidade de usar engenharia reversa em Java, C# e C++ [10].

D. *Bootstrap*

Lançado originalmente em 19 de Agosto de 2011, o Bootstrap é um dos *frameworks* de *front-end* e projeto de código aberto mais populares do mundo [11].

É um *open source*, que possui bibliotecas HTML, CSS, JS que facilitam os desenvolvedores a criar sites e aplicativos [12].

Além de ser responsivo e móvel, algo essencial por torná-lo tão popular entre os desenvolvedores [12].

E. *Python*

Linguagem criada em 1990 por Guido van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda (CWI), era focada para físicos e engenheiros [13].

De acordo com Borges [13], o Python é uma linguagem de alto nível orientada a objeto, de tipagem dinâmica e forte, interpretada e interativa. Além de ser uma linguagem que possui estruturas de alto nível, vasta coleção de módulos para uso e *frameworks* que podem ser adicionados por terceiros [13].

Python possui código aberto, o que permite que cada pessoa possa alterar sua codificação e criar suas próprias

bibliotecas. Também permite a utilização de outras ferramentas e funcionalidades, por exemplo a interação com outras linguagens como C/C++, Java, C#, NET, PHP, entre outras [14].

F. Django

O Django é um *framework* desenvolvido para cumprir prazos rápidos e ajudar os desenvolvedores experientes, cuidando da parte *web* para que o programador possa se concentrar apenas no seu aplicativo [15].

Django possui suporte pronto para uso de tarefas comuns no desenvolvimento *web*, como: autenticação de usuários, testes, modelos de banco de dados, rotas de URL, interface de administração e segurança [16].

G. Banco de Dados MySQL

MySQL é o banco de dados de código aberto mais popular do mundo. É um banco de dados relacional, dessa forma armazena os dados em tabelas separadas ao invés de guardar todos os dados em apenas um grande repositório [17].

É um banco de dados rápido, confiável, escalável e fácil de usar, além de suportar muitas linguagens e drivers de desenvolvimento, por exemplo: PHP, Python, JAVA/JDBC, C, C++, etc. Está presente em muitos aplicativos de constante acesso como Facebook, Twitter, Netflix e Uber [17].

H. Scrum

O *Scrum* foi criado por Jeff Sutherland juntamente com Ken Schwaber em 1993, com o propósito de desenvolver softwares de uma maneira mais rápida, confiável e eficiente [18].

A maior parte dos softwares nesta época eram desenvolvidos no método cascata, onde o projeto é concluído em etapas diferentes e levado passo a passo até que fosse entregue para os usuários [18].

De acordo com Sutherland [18] :

"O processo era lento, imprevisível, e muitas vezes não resultava em um produto que as pessoas quisessem ou pelo qual se dispusessem a pagar".

Além disso, Sutherland [18] afirma:

"O Scrum incorpora os conceitos de aprimoramento contínuo e produto viável mínimo (MVP, na sigla em inglês) para receber retorno imediato dos consumidores, em vez de esperar até que o projeto esteja completo."

Conforme Cruz [19], no *Scrum* os projetos são divididos em ciclos, também chamados de *Sprints*, que podem durar de 2 a 4 semanas com metas e objetivos estabelecidos.

O *Scrum* consiste em um conjunto formado por times pequenos, onde cada Time *Scrum* é composto por: *Scrum-master*, *Product Owner* e o time [19].

O *Scrummaster* é o responsável pelo treinamento e desenvolvimento do Time Scrum, assim como garantir que o Time Scrum siga o Fluxo Scrum [19]. O *Product Owner* fica responsável pelo gerenciamento do *Backlog* do Produto, além de agregar valor ao produto e cliente [19].

Por fim, o time é composto por desenvolvedores que devem transformar o *Backlog* do Produto em funcionalidades que possam ser apresentadas para o cliente[19].

I. Requisitos do Sistema

Os requisitos do sistema podem ser classificados como Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) [20]. Os Requisitos Funcionais são aqueles que não dependem da tecnologia ou ferramenta utilizada e descrevem o que o sistema tem a oferecer [20].

Os Requisitos Não Funcionais são aqueles que indicam à confiabilidade do sistema, tempo de respostas e tolerância à falhas [20].

J. Diagrama de Caso de Uso

O Diagrama de Caso de Uso, segundo Rangel e Carvalho Jr. [20]:

"são explicitações de utilização de uma funcionalidade do sistema, sendo uma ótima maneira de representar os requisitos funcionais encontrados."

Utiliza ferramentas como gráficos, textos, eventos, que auxiliam tanto o desenvolvedor quanto o cliente em relação as suas expectativas para o produto [20].

K. Diagrama de Classes

De acordo com Rangel e Carvalho Jr. [20]:

"O Diagrama de Classes permite expressar o domínio da aplicação e definir a estrutura da implantação do projeto de software, mapeando o modelo de negócios no código necessário à sua implantação."

Ainda permite que o usuário possa especificar os detalhes da implementação como os valores que transitam entre os objetos, tipos de dados, regras e acessos [20].

III. TRABALHOS CORRELATOS

Nesta seção serão apresentados trabalhos relacionados ao tema ou que contenham ferramentas e tecnologias semelhantes para o desenvolvimento deste projeto.

A. Aplicação web para gerenciamento de pacientes diabéticos do município de Guarapuava

O trabalho de Silva e Ribeiro [21] trata-se de um sistema *web* para realizar o gerenciamento de pacientes diabéticos do município de Guarapuava, Paraná. Com o objetivo de elaborar um sistema para traçar o perfil de pacientes que fazem o uso de insulina, utilizando informações já existentes em seus prontuários, facilitando um possível acompanhamento desses pacientes.

Outros objetivos de seu trabalho são criar um banco de dados contendo todas as informações obrigatórias dos pacientes, gerenciar a quantidade de entrada e saída de insulina pelo sistema, além de gerar relatórios de acordo com o consumo de insulina realizado pelos pacientes.

Para a análise de requisitos do sistema, foi realizada uma entrevista do tipo Abordagem Dedutiva, com objetivo de recriar os processos que os funcionários realizavam. Por

meio dessas informações criou-se uma prévia da interface do sistema.

O sistema foi desenvolvido na linguagem PHP, para o *frontend* foi utilizado o *framework* de desenvolvimento *web* Bootstrap e para armazenamento de dados o SGBD MySQL.

Após o sistema ser entregue, os usuários tiveram um tempo de adaptação contando com a ajuda dos desenvolvedores para tirar dúvidas que surgissem.

O sistema apresentou como resultado, maior facilidade nas atividades dos profissionais de saúde, inclusão de 150 pacientes no sistema, além de relatórios e informações mais detalhadas dos pacientes, como por exemplo faixa etária, gênero, insulina mais consumida, entre outros.

B. Uso da arquitetura orientada a serviço (soa) aplicada na área de saúde

O trabalho de Oliveira et al. [22] tem como objetivo desenvolver um sistema em uma arquitetura orientada a serviços, visando facilitar os agendamentos médicos por meio de um sistema *web* na rede de saúde privada.

Alguns dos objetivos específicos do sistema são agilizar os processos de encaminhamento e pedidos de exames, ser acessível por meio da *web* e facilitar a comunicação entre clínicas de saúde.

A metodologia utilizada neste trabalho, é a Arquitetura Orientada a Serviços (SOA), que realiza a integração e a reutilização de serviços. Para conter as informações dos pacientes, foi utilizado o Prontuário Eletrônico do Paciente (PEP), tecnologia que pode facilitar a integração de sistemas de saúde.

Foi utilizado para a modelagem do sistema a ferramenta Astah Community, linguagem de programação Python e para o desenvolvimento *web* o *framework* Python Django.

Como resultado, o projeto conseguiu desenvolver um sistema para prontuários eletrônicos, facilitando e agilizando o encaminhamento de exames a serem feitos. As partes do sistema implementadas durante o projeto foram: página inicial, tela de pacientes e médicos cadastrados, página com lista de prontuários e atestados, página de exames e agendamentos.

C. Sistema de informação para controle de veículos e viagens de uma transportadora

O trabalho de Silva e Neto [23] trata de um software *web* para gerenciar os veículos e viagens de uma transportadora de pequeno porte.

A justificativa do trabalho foi a necessidade de melhoria de gerenciamento da frota e dos fretes da Mota Transportes, empresa especializada no transporte de cargas do agronegócio.

Os objetivos do sistema são facilitar o dia a dia, analisar o desempenho dos veículos em relação ao custo de manutenção, combustível e quantidade de fretes realizados além de auxiliar nas tomadas de decisões da empresa.

Por meio de uma entrevista com o *stakeholder* da empresa, os autores obtiveram as informações sobre os requisitos do sistema e a partir disto criaram o modelo de dados conceitual e lógico do projeto. O sistema foi desenvolvido na linguagem de programação Python, utilizando o *framework* Django. Para o *front-end* a ferramenta Bootstrap e para o armazenamento de informações, o Banco de Dados MySQL.

O resultado final apresentou um sistema que vai ajudar na tomada de decisão da empresa, pois estará disponível vinte e quatro horas por dia via internet e auxiliará na fiscalização e acompanhamento das viagens e motoristas, por parte do gerente e dono da empresa.

D. Considerações sobre os trabalhos correlatos

O trabalho de Silva e Ribeiro [21] vem a colaborar com este projeto em função de possuir o mesmo tema, visto que trata-se de um sistema para pacientes que possuem diabetes e necessitam de um acompanhamento, contendo também a inserção de um cadastro do perfil dos pacientes com suas informações.

No trabalho de Oliveira et al. [22] um de seus objetivos é facilitar a relação entre médico e paciente, dispensando a papelada e utilizando um software para ter acesso aos registros, algo que se deseja implementar neste projeto.

Além de possuir algumas funções semelhantes como a realização de agendamentos e local, além de identificar os médicos de acordo com sua especialidade.

O trabalho de Silva e Neto [23] possui diferentes níveis hierárquicos dentro de seu sistema onde cada usuário contém suas respectivas permissões, assim como possui um CRUD (*Create, Read, Update and Delete*) para os usuários. O trabalho de Silva e Neto [23] dispõe basicamente todas as ferramentas e tecnologias que serão abordadas neste projeto.

IV. METODOLOGIA

Para o desenvolvimento deste trabalho, foi inicialmente realizada uma revisão bibliográfica abordando temas como ferramentas e trabalhos correlatos, visando adquirir conhecimento acerca do tema escolhido para o projeto.

Para isso também foi estudado o *Scrum* que permitiu um melhor planejamento e execução deste trabalho.

De acordo com a metodologia *Scrum*, realizaram-se entregas mensais ao cliente, onde cada módulo possui certas funcionalidades do sistema.

A. Módulos do sistema

Na Figura 1 são apresentados os módulos do sistema que foram implementados em cada etapa.

O módulo I do sistema contém as aplicações Usuário, Profissional e Sócio. O aplicativo Usuário, necessita de informações como: nome, cpf, matrícula, email, tipo. Neste módulo também foram desenvolvidos o aplicativo Profissionais e Sócios, detalhados na Figura 7 e Figura 8.

Para o módulo II do desenvolvimento, foram desenvolvidos os aplicativos Consultório e Agendamentos.

O aplicativo Consultório controla os consultórios da associação, gerenciando-os caso estejam ativos ou inativos para realizar agendamentos.

Para o aplicativo Agendamento, é necessário selecionar o profissional, o sócio (ativo no sistema) e consultório, posteriormente escolher data e hora do agendamento.

Por fim, no módulo III foram desenvolvidos os aplicativos Mensalidade, Pagamento e Consulta.

O aplicativo Mensalidade contém o valor da mensalidade da associação, a data de alteração e qual mensalidade esta ativa no momento.

No aplicativo Pagamento, é realizado o cadastro de pagamentos dos sócios, incluindo data do pagamento e mês referente ao pagamento. Nele é possível realizar consultas e buscas, filtrando o mês dos pagamentos ou o nome dos sócios.

Para o aplicativo Consulta, é necessário que o profissional selecione o agendamento e posteriormente complete o relatório da Consulta.

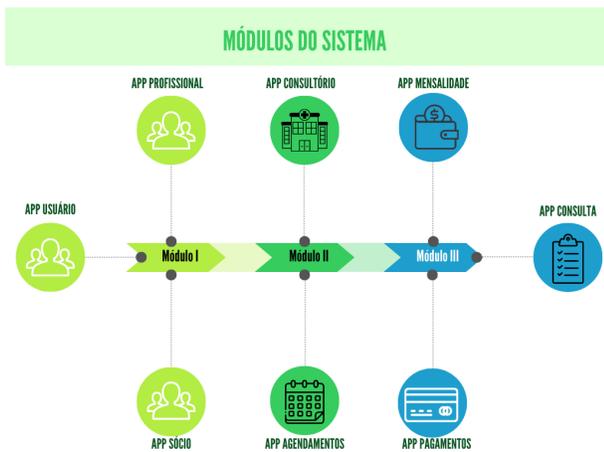


Figura 1. Módulos do Sistema [dos autores]

B. Requisitos Funcionais

Na Tabela 1 são listados os Requisitos Funcionais do sistema.

Requisitos Funcionais	
ID	Descrição
RF1	O usuário deverá efetuar um <i>login</i>
RF2	O administrador deverá realizar: cadastro de profissionais: RF2.1 Cadastrar profissional RF2.2 Consultar profissional RF2.3 Editar profissional RF2.4 Excluir Profissional
RF3	O administrador deverá realizar o cadastro de sócios: RF3.1 Cadastrar sócio RF3.2 Consultar sócio RF3.3 Editar sócio RF3.4 Excluir sócio
RF4	O administrador deverá gerenciar os consultórios.
RF5	O administrador deverá gerenciar a mensalidade.
RF6	O administrador deverá realizar o agendamento de horários de consultas.
RF7	O administrador deverá ter acesso aos pagamentos.
RF8	A aplicação deverá permitir o profissional gerenciar agendamentos e consultas.
RF9	A aplicação deverá permitir o sócio visualizar agendamentos e consultas.
RF10	Apenas sócios ativos no sistema podem ser selecionados para o agendamento.

C. Requisitos Não Funcionais

Na Tabela 2 são listados os Requisitos Não Funcionais do sistema.

Na Figura 6 é possível visualizar a página inicial do sistema pela visão do Administrador.

Incluindo a parte lateral com o menu de acesso aos aplicativos criados.

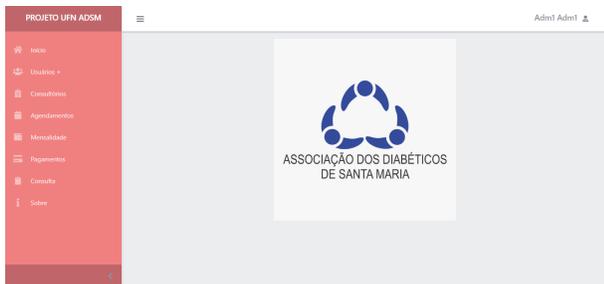


Figura 6. Home

A Figura 7 exibe a lista de profissionais registrados no sistema, incluindo informações como nome, e-mail, profissão, último acesso, situação e opções para editar e excluir.

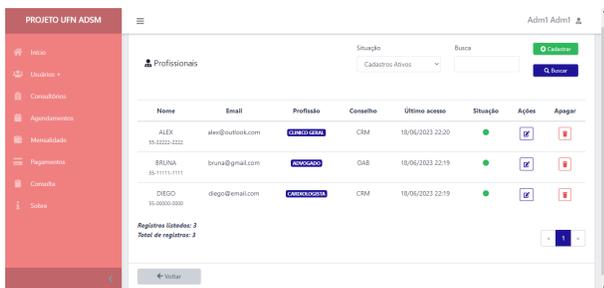


Figura 7. Listagem de profissionais do sistema

Na Figura 8 é possível visualizar o formulário de cadastro de sócio ao sistema, onde o administrador deve inserir o tipo sanguíneo, alergias, doenças, telefone de emergência, entre outras informações.

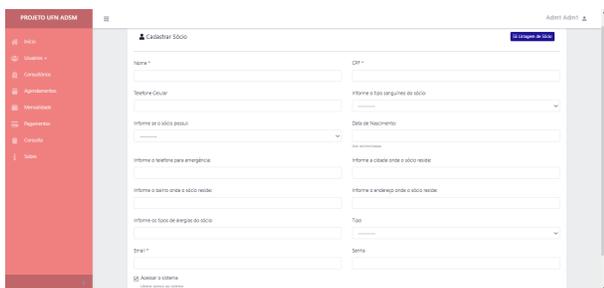


Figura 8. Formulário para cadastro de sócio

A Figura 9 apresenta a lista de agendamentos, onde é possível visualizar data, hora, profissional, sócio e consultório. Administradores e profissionais do sistema podem gerenciar esse aplicativo.

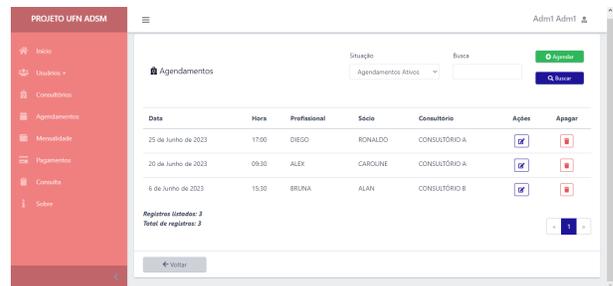


Figura 9. Agendamentos

A Figura 10 mostra a mensalidade, contendo o valor, data de alteração e se está ativa no momento.

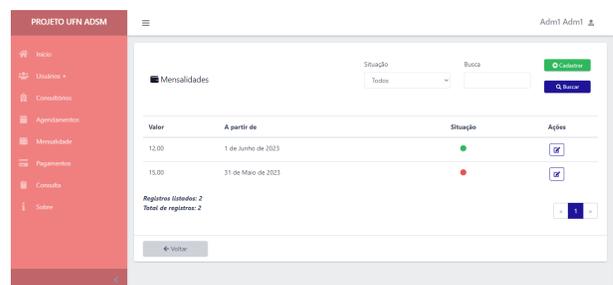


Figura 10. Mensalidades

Na Figura 11 é possível visualizar a tela de pagamentos, onde administrador poderá realizar o registro de um novo pagamento ou buscas, filtrando tanto o mês do pagamento ou nome do sócio.

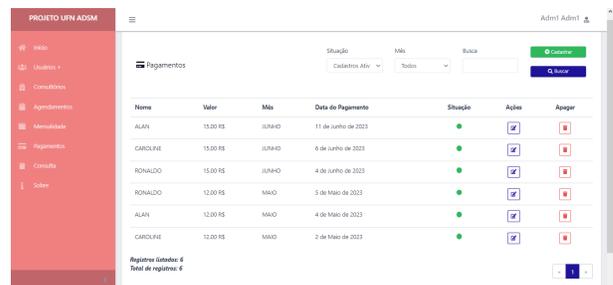


Figura 11. Pagamentos

VI. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho é apresentado o desenvolvimento de um sistema Web que auxilie no gerenciamento da secretaria da Associação de Diabéticos de Santa Maria, permitindo a gestão automatizada de: profissionais, sócios, consultórios, agendamentos, consultas, mensalidades e pagamentos da ADSM.

Os trabalhos correlatos auxiliaram no desenvolvimento do projeto, visto que foi possível compreender melhor o

funcionamento das ferramentas utilizadas no sistema, principalmente a arquitetura do modelo MVT (*Models, Views, Templates*) do Django.

Adotando a metodologia *Scrum*, foram feitas entregas mensais ao cliente, inicialmente pela *landpage* (site inicial do sistema) para divulgação da associação.

Em seguida foi realizada a entrega do Módulo I, permitindo o gerenciamento de cadastro dos administradores, sócios e profissionais no sistema.

As demais entregas foram ocorrendo em sequência, onde o segundo módulo possibilitou efetuar o gerenciamento dos consultórios e agendamentos de consultas.

O módulo final foi entregue, contendo as aplicações de gerenciamento de mensalidade, registro e busca de pagamentos e registro de consulta.

O sistema está hospedado no domínio <https://adsm.lapinf.ufn.edu.br/> sob o suporte do Laboratório de Práticas dos cursos de Computação da Universidade Franciscana. Dessa forma, permitiu uma automação das atividades diárias da associação, agilizando e otimizando o gerenciamento dos serviços prestados.

Como proposta para trabalhos futuros, poderia ser desenvolvido uma integração do Google Calendar API com o sistema visando um melhor controle dos agendamentos.

Outras propostas são o desenvolvimento de um prontuário eletrônico de pacientes (PEP), contendo os dados e anotações de todas consultas realizadas no sistema, além de um aplicativo Medicamentos, onde o profissional possa inserir medicamentos em um receituário com as prescrições e orientações ao sócio, no momento da consulta.

Também poderão ser desenvolvidas as funcionalidades não previstas para a primeira versão, que chegarão como demanda do Laboratório de Práticas para serem implementadas por estudantes que venham a desenvolver seus trabalhos de conclusão de curso em continuidade a este.

REFERÊNCIAS

- [1] X. B. Petermann et al. “Epidemiologia e cuidado à diabetes mellitus praticado na atenção primária à saúde: uma revisão narrativa”. Em: *Saúde (Santa Maria)* 41.1 (2015). Disponível em: <https://periodicos.ufsm.br/index.php/revistasaude/article/view/14905>, pp. 49–56.
- [2] International Diabetes Federation. *IDF Diabetes Atlas 10th edition*. IDF, 2021.
- [3] H. de F. Marin. “Sistemas de informação em saúde: considerações gerais”. Em: *Journal of Health Informatics* 2.1 (2010). Disponível em: <https://jhi.sbis.org.br/index.php/jhi-sbis/article/view/4>.
- [4] MINISTÉRIO DA SAÚDE Secretaria de Atenção à Saúde Departamento de Atenção Básica. *Diabetes Mellitus Cadernos de Atenção Básica - n.º 16*. Editora MS, 2006.
- [5] Sociedade Brasileira de Diabetes. *Diabetes*. URL: <https://diabetes.org.br>. (accessed: 05.10.2022).
- [6] Ministério da Saúde. *Diabetes*. URL: <https://bvsms.saude.gov.br/diabetes/>. (accessed: 05.10.2022).
- [7] Paula Krater Canêo e João Marcelo Rondina. “Prontuário Eletrônico do Paciente: conhecendo as experiências de sua implantação”. Em: *Journal of Health Informatics* 6.2 (2014). Disponível em: <https://jhi.sbis.org.br/index.php/jhi-sbis/article/view/289/197>, pp. 67–71.
- [8] Sociedade Brasileira de Informática em Saúde. *Categorias e Modalidades Certificáveis*. URL: <http://sbis.org.br/categorias-modalidades/>. (accessed: 21.10.2022).
- [9] Odirlei Antonio Magnagnago, Edimara Mezzomo Luciano e Rafael Mendes Lübeck. “Como proteger informações do prontuário eletrônico do paciente: proposta de mecanismos”. Em: *Ciência da Informação* 49.2 (2020). Disponível em: <https://revista.ibict.br/ciinf/issue/view/290/69>, pp. 23–39.
- [10] Astah. *Astah UML*. URL: <https://astah.net/products/astah-uml/>. (accessed: 05.10.2022).
- [11] Bootstrap. *About*. URL: <https://getbootstrap.com/docs/5.2/about/overview/>. (accessed: 29.09.2022).
- [12] Philippe Hong. *Practical Web Design: Learn the fundamentals of web design with HTML5, CSS3, Bootstrap, jQuery, and Vue.js*. Packt Publishing, 2018.
- [13] Luiz Eduardo Borges. *Python para desenvolvedores: Aborda Python 3.3*. Novatec Editora, 2014.
- [14] C. P. Santiago et al. “Desenvolvimento de sistemas Web orientado a reuso com Python, Django e Bootstrap”. Em: (2020).
- [15] Django. *Overview*. URL: <https://www.djangoproject.com/start/overview/>. (accessed: 29.09.2022).
- [16] William S. Vincent. *Django for Beginners: Build websites with Python and Django*. WelcomeToCode, 2018.
- [17] Oracle. *What is MySQL?* URL: <https://www.oracle.com/mysql/what-is-mysql/>. (accessed: 05.10.2022).
- [18] Jeff Sutherland. *Scrum: a arte de fazer o dobro do trabalho na metade do tempo*. LeYa Editora Ltda, 2019.
- [19] Fábio Cruz. *Scrum e PMBOK unidos no gerenciamento de projetos*. Brasport, 2013.
- [20] Pablo Rangel e J.G. de Carvalho Jr. *Sistemas Orientados a Objetos teoria e prática com UML e Java*. Brasport Livros e Multimídia Ltda, 2021.
- [21] Rafael Souza da Silva e Richard Duarte Ribeiro. “Aplicação web para gerenciamento de pacientes diabéticos do município de Guarapuava”. Em: (2015). Trabalho de Conclusão de Curso. Disponível em <<http://repositorio.utfpr.edu.br/jspui/handle/1/16883>>.

- [22] Eduardo Augusto Rodrigues de Oliveira, Renan dos Santos e Márcio Augusto Ernesto de Moraes. “Uso da arquitetura orientada a serviço (soa) aplicada na área de saúde.” Em: (2019). Trabalho de Conclusão de Curso. Disponível em <<http://repositorio.unitau.br/jspui/handle/20.500.11874/4587>>.
- [23] Renato Queiroz Silva e Olegário Correa da Silva Neto. “Sistema de informação para controle de veículos e viagens de uma transportadora”. Em: (2022). Trabalho de Conclusão de Curso. Disponível em <<https://repositorio.pucgoias.edu.br/jspui/handle/123456789/3989>>.