**TGI-II 2022 em**

**VAL – ASSISTENTE VIRTUAL LINUX**

**Bruno M. S. Cerqueira, Jeanderson C. P. Sardinha, Matheus A. R. Cunha**

Universidade Cruzeiro do Sul

Av. Dr. Ussiel Cirílo, 225 - Vila Jacuí, São Paulo - SP, 08060-070 - Brasil

vhvic

**Resumo**

Atualmente no mundo a fora existem bilhões de pessoas que necessitam da tecnologia, seja para acomodar suas vidas trazer algo complicado de forma simples, pratica e rápida ou apenas se baseando em algo do dia a dia. Bem com isso decedimos trazer nossa assistente virtual a VAL. Onde ela ira executar tarefas/programas por controle de voz assim ajudando varias pessoas especiais, atualmente o numero de pessoas com problemas vem aumentando e sentimos que poderíamos mudar a vida dessas pessoas tornando a vida delas bem mais fácil e simples...

**Palavras-chaves:** VAL, Python, Yaml, PyAudio e Linux.

**Abstract**

Currently in the outside world there are billions of people who need technology, whether to accommodate their lives to bring something complicated in a simple, practical and fast way or just based on something day to day. Well with that we decided to bring our virtual assistant to VAL. Where she will perform tasks / programs by voice control thus helping several special people, currently the number of people with problems is increasing and we feel that we could change the lives of these people making their lives much easier and simple...

**Keyword:** VAL, Python, Yaml, PyAudio e Linux.

**1. Introdução**

**1.1 Apresentação do tema**

Conforme o numero maior de pessoas com problemas de comorbidades vem aumentando a necessidade da tecnologia vem como isto vem aumentando ano após anos varias pessoas hoje em dia não consegue navegar no Windows ou no Linux de maneira pratica, é por isso que estamos aqui desenvolvendo a nossa assistente virtual que poderá ajudar essas pessoas com praticidade e efetividade.  
Onde a pessoa não precisara de clicar ao iniciar um processo no sistema, a VAL fara todo o trabalho ao usuário assim tornando a vida mais pratica e fácil. A VAL é uma assistente virtual que fara com que o sistema inicie programas, faça algumas tarefas que estão listadas no manual dela que estará descrito nos próximos itens.

**1.2 Problema de pesquisa**

Devido a utilização da linguagem Python para criar o software ouve varios empecilhos um deles é a má configuração da biblioteca pyaudio que atualmente conta com vários problemas de reconhecimento de voz, não entendendo com clareza o que o usuário esta falando, tentamos de vários jeitos contornar o problema de reconhecimento do pyaudio mas isso é um problema dos próprios desenvolvedores da biblioteca, tivemos vários problemas também com a biblioteca Yaml que é quem faz a conversa entre software e sistema, isto pelo fato de haver varias versões da yaml diferentes espalhadas e a dificuldade de instalar a versão correta na versão do python, atualmente no python se a linguagem atualiza as bibliotecas tem que atualizar junto e a grande parte do progresso do software é perdido com essas atualizações.

**Por que Python?**

Vimos que as outras linguagens não oferecia melhor comorbidade que a linguagem python, pensamos em criar o projeto full Java ou shell script mas haveria muitos problemas em relação a funcionamento e servidores de áudio, por causa da vasta gama de distros linux as vezes essa gama tem N servidores de áudios onde a biblioteca do Java poderia quebra a qualquer momento

**1.3 Objetivo**

O objetivo deste artigo e software é realmente poder fazer com que as pessoas de morbidade ou dificuldades especiais venham a interagir melhor com o sistema operacional, assim eliminando boa parte da tarefa dos cliques com o mouse ou dificuldade de abrir determinado programa listados na VAL. Alem disso iremos demonstrar o programa funcionando neste artigo além de descreve-lo melhor abaixo

**1.4 Justificativa**

Os estudos sobre a utilização de um banco de dados não-relacional vem sendo cada vez mais atrativa no que se diz respeito a qualidade de acesso e manipulação de dados. O aumento na quantidade e complexidade dos dados também são geradores de desafios para as técnicas tradicionais de análise e tratamento de informações, tanto pelo armazenamento, quanto a velocidade de acesso a mesma.

Deste modo, grandes empresas já vêm implementando e atualizando suas próprias rotinas e meios para se adaptar ao crescente fluxo que passa por seus servidores. Visando nossa área de estudo, a saúde, esta implementação visará agilizar o atendimento ao paciente, fazendo com que esse atendimento possa ser prestado em qualquer lugar, uma vez que seus dados serão compartilhados com todas as unidades de saúde, fazendo com que os médicos possam ter acesso facilitado a estas informações, diminuindo os custos das análises de hipótese diagnostica e diminui a necessidade da solicitação de exames.

**2. Material e Métodos**

Inicialmente, construímos a concepção do projeto em base de artigos científicos, os quais tratam do mesmo assunto/objetivo que estamos propondo neste trabalho. Abordando a eficácia referente ao uso de Big Data em programas de gestão hospitalar, que engloba as perspectivas futuras da utilização dele no Brasil, o artigo de Alexandre D. P. C. Filho “Uso de big data em saúde no Brasil: perspectivas para um futuro próximo” presente em: <https://www.scielosp.org/article/ress/2015.v24n2/325-332/> faz uma elaboração das três principais áreas propensas ao uso desde tipo de estrutura de dados na área da saúde: medicina de precisão, prontuários eletrônicos do paciente e internet das coisas. Contudo, além da perspectiva futura de sua utilização, precisávamos buscar informações do seu uso de fato no mundo corporativo, com detalhes, para revisar as oportunidades e desafios que este tipo de implementação poderá trazer.

Visando tal discussão, encontramos um artigo que constrói a ideia da revisão de sua utilização no mundo corporativo. Tal artigo foi feito pela empresa Internacional Symposium on Project Management, Innovation and Sustainability, disponível em: <https://singep.org.br/4singep/resultado/245.pdf>.

Porém o que nos cativou a desenvolver um projeto baseado em um banco de dados não relacional foi a diferença de armazenamento e velocidade na recuperação de dados deste tipo de estrutura comparada com a estrutura relacional. Para o nosso protótipo, faremos testes visando os resultados apresentados pelo artigo de Marcos A. P. M. Filho, disponível em <https://www.cin.ufpe.br/~tg/2014-2/mapmf.pdf>, que realiza a análise de desempenho estre os dois tipos de banco de dados.

Tendo em mente os resultados obtidos nos artigos científicos citados anteriormente, nosso projeto visa por meio desta estrutura comprovar a eficácia de armazenamento e de velocidade do banco de dados não-relacional aplicado no sistema de saúde.

Para expressarmos um resultado adequado ao que estamos propondo, faz-se necessário o uso dos dois tipos de banco de dados (Relacional e não-relacional) para analisar e comparar a performance de um sistema usando cada um deles.

Para provar a efetividade do modelo não-relacional e para atingir o objetivo proposto, começaremos pela demonstração do modelo (relacional) que consideramos menos eficaz para esse tipo de sistema. O desenvolvimento do projeto que o utilizará, se baseará na linguagem de programação Java com a implementação de Servlets, classes que atuarão como gerenciadores de requisições e respostas entre o cliente e o servidor. Haverá uma Servlet gerencial (principal) que funcionará como filtro indicando para qual classe as requisições feitas deverão ser encaminhadas. Esta classe que recebe a requisição, implementará os métodos básicos para conseguir efetuar o tratamento da requisição e conseguir devolver uma resposta adequada para tal (Métodos doGet e doPost, que são nativos e obrigatórios para a implementação da Servlet).

A programação Java fará a “comunicação” com o usuário através de páginas JSP (JavaServer Pages) que implementarão a interface gráfica para o usuário poder interagir com o sistema, as quais serão construídas em HTML5, CSS3, Bootstrap4 e AJAX, e por fim farão contato com as classes gerenciais através da programação Javascript. O algoritmo das classes gerenciais, terão como objetivo “traduzir” a informação recebida através da requisição feita pelo Javascript e fazer contato com o banco de dados pelo modelo DAO, que utilizará JDBC para conseguir fazer essa ponte entre os diferentes tipos de ambiente.

O banco de dados será construído em uma linguagem de programação estruturada, conhecida formalmente como SQL e de forma geral este programa utilizará o modelo MVC em seu desenvolvimento, o qual funciona da seguinte forma: uma classe gerencial recebe a requisição do usuário, ela chama os métodos subsequentes podendo, ou não, realizar o contato com o banco de dados através de métodos DAO e retorna ao usuário uma resposta adequada para o que lhe foi pedido.

Para a demonstração com uso de banco de dados não relacional analisamos dentre várias opções; MongoDB, Redis, Cassandra, entre outras. Escolhemos MongoDB por ser um banco extremamente popular e em estado maduro, bem como sua estrutura baseada em Documents que são muito semelhantes a arquivos JSON o que permite uma flexibilidade maior com a forma com que persistimos os dados, o que consideramos fundamental para o setor hospitalar e de prontuários digitais.

Optamos por criar um sistema de CRUD que simula as tarefas cotidianas do domínio hospitalar; cadastro de pacientes, registro de atendimentos, efetuar internações e obter listas de pacientes internados. Utilizamos a linguagem javascript com o uso da tecnologia Nodejs para criar o nosso backend, auxiliado pelo framework Express para criar a aplicação Web. A escolha foi motivada por conta da Stack de desenvolvimento Nodejs, Express e MongoDB ser bastante popular e difundida.

**3. Resultados**

A construção da interface gráfica de usuário final divergiu entre dois modelos diferentes, pois foram feitos em bases diferentes de programação. A aplicação que serviu de base para os testes do modelo relacional se inicia com uma tela de login de usuário (figura 01).

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Após realizar a requisição de POST, para a confirmação do login e senha, a aplicação irá conferir se este usuário é cadastrado no sistema, e se sim, passará para a próxima tela de menu. O menu principal (figura 02) atribui algumas funções básicas de gerenciamento, tanto do sistema quanto da parte de abertura das fichas de atendimento. Temos três opções que foram suficientes para a realização dos testes de tempo e resposta: Home, Administrativo, Internação.

A opção “Home” (figura 02), e a logo presente a sua esquerda, leva de volta para o menu principal.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

O menu “Administrativo” tem como propósito a gerência de usuários do sistema, apresentando duas opções de escolha: alterar a senha do usuário logado na sessão e gerenciar (Cadastrar e alterar) os usuários em sistema. O menu se auto adaptará para o tipo de usuário que iniciou a sessão. Caso seja um usuário comum, estará disponível apenas a tela de alterar sua senha. No entanto se for um administrador, ele terá o poder de criação e alteração conforme apresentado na figura 03.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

O menu de internação realiza uma das principais funções de um sistema hospitalar que é gerar a ficha de atendimento do paciente, conforme é apresentado na figura 06.

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

Inicialmente, é necessário informar o CPF do cidadão (figura 07) para realizar a pesquisa dele. Caso um paciente não possua cadastro no sistema, o usuário logado na sessão consegue realizar o registro na mesma tela conforme apresenta a figura 08.

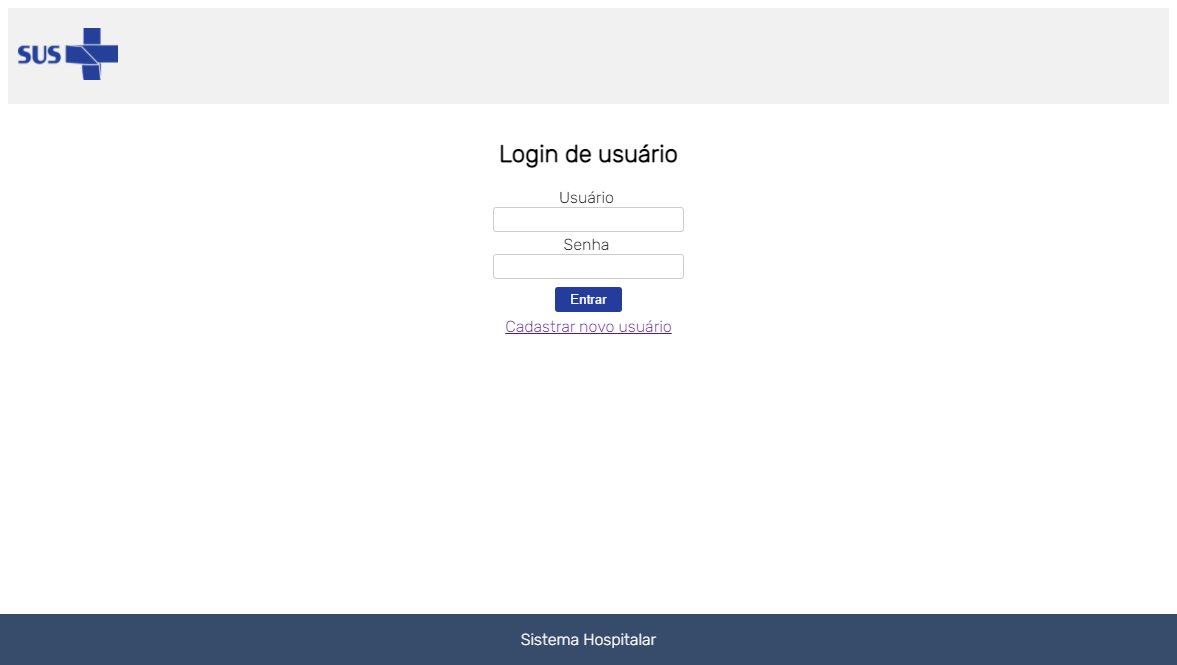
Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

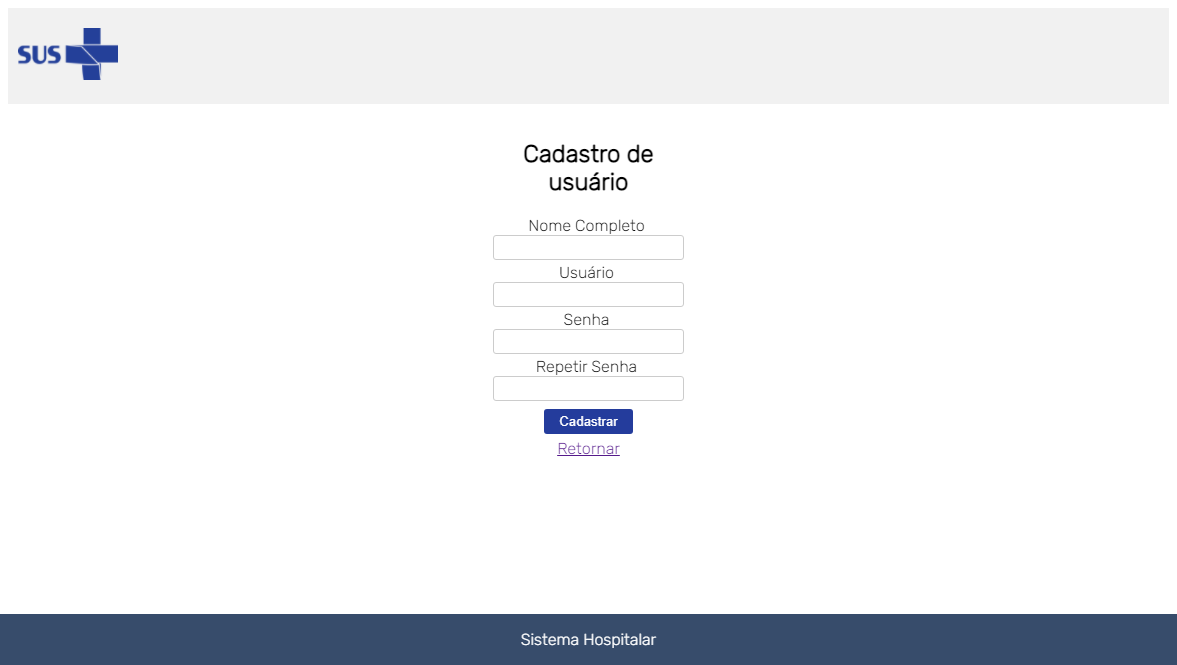
Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Para o sistema de banco de dados não relacional (NoSQL), a interface gráfica foi desenvolvida através do NodeJS no back-end em conjunto com Express para facilitação das requisições em semântica. O front-end foi desenvolvido através do EJS, CSS3 e HTML5, sendo toda base do banco dados feita em mongoDB por conta da sua estrutura de dados gerar objetos JSON, do tipo Documento, tendo como interface inicial o login para os funcionários.



Através do método POST enviando a requisição para o banco de dados para buscar a existência do usuário e fazendo comparativo da senha, enviando alerta caso o usuário não tenha cadastro no sistema. O usuário tem acesso a criação de um novo cadastro para login ainda na mesma tela.



Optamos por um cadastro simples mediante o fato de demonstração, porém num sistema hospitalar são necessários cadastros completo dos dados, já que para uma integração unificada e de funcionamento amplo outros módulos são necessários como “Administrativo”, com registro dos funcionários para os Recursos Humanos, e complementos importantes para utilização de um prontuário 100% eletrônico como dados de formação dos especialistas, números de “Conselhos multiprofissionais” e etc. O que a princípio não é importante para a demonstração.

Sequentemente, após cadastro e login do usuário, partimos para o “HOME” Que apresenta inicialmente um menu de acesso no cabeçalho com os principais links da demonstração, e no corpo da página uma listagem dos pacientes internados.



A lista utiliza de método GET, e através de renderização no back-end constrói a listagem em tempo real, com suas principais informações do paciente. Inicialmente o funcionário deverá pesquisar a existência do paciente através do seu CPF.



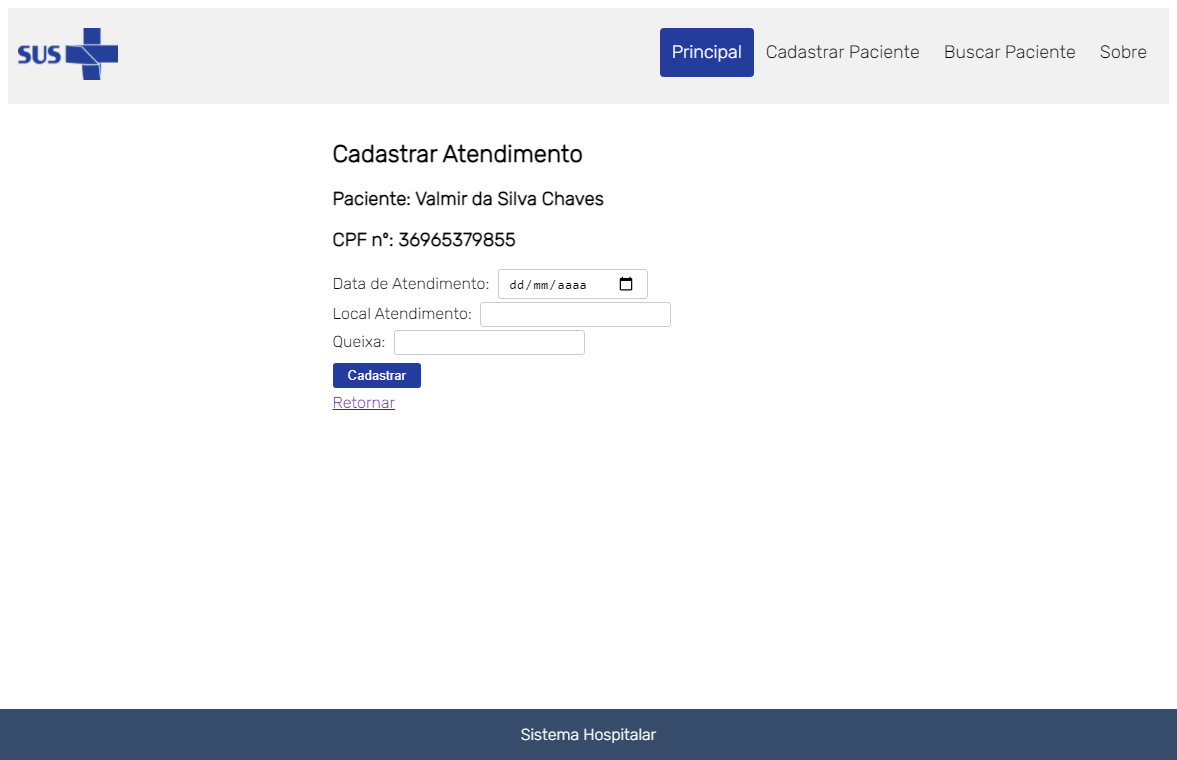
Caso o paciente exista, será renderizada uma tabela com os dados básicos dele para comparação e 4 botões para ações referentes as necessidades do mesmo, como Atendimento, Internação, Histórico etc. Se ele não estiver cadastrado, a tabela aparecerá em branco.



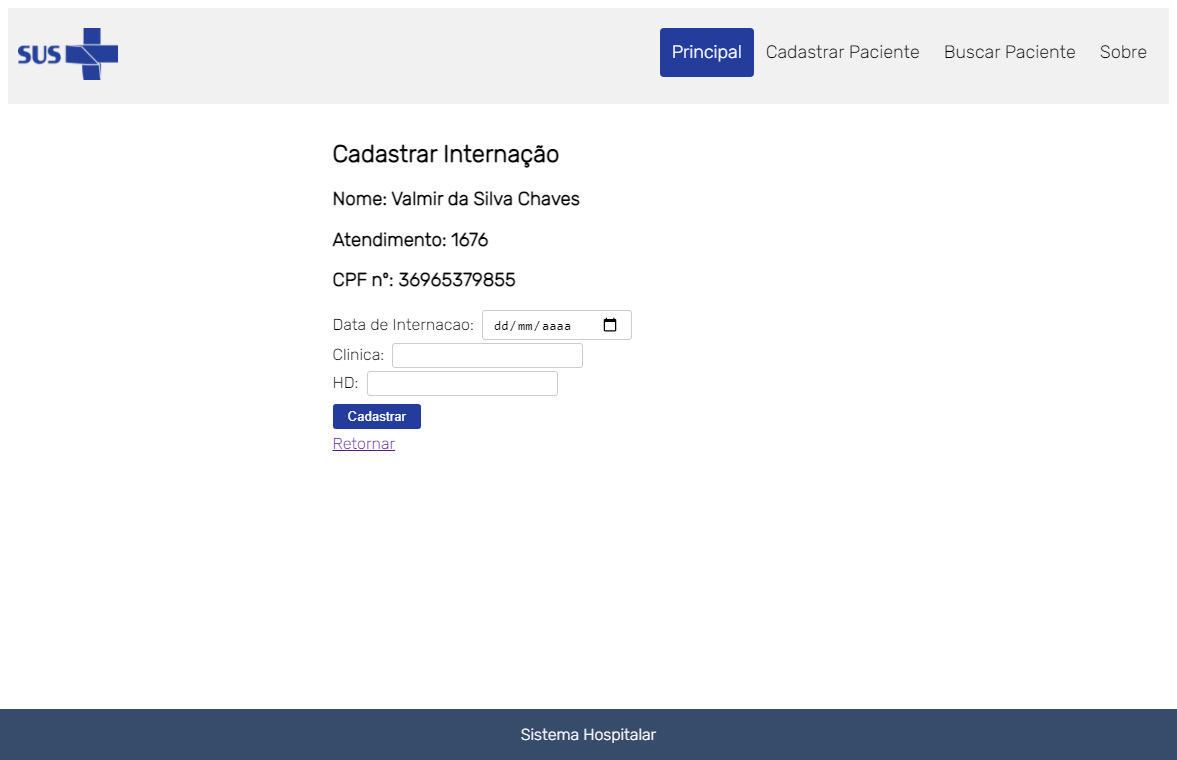
A partir disso o funcionário acessará a página de cadastro de paciente, que através de formulário simples preenche todos os dados necessários para a finalização do processo.



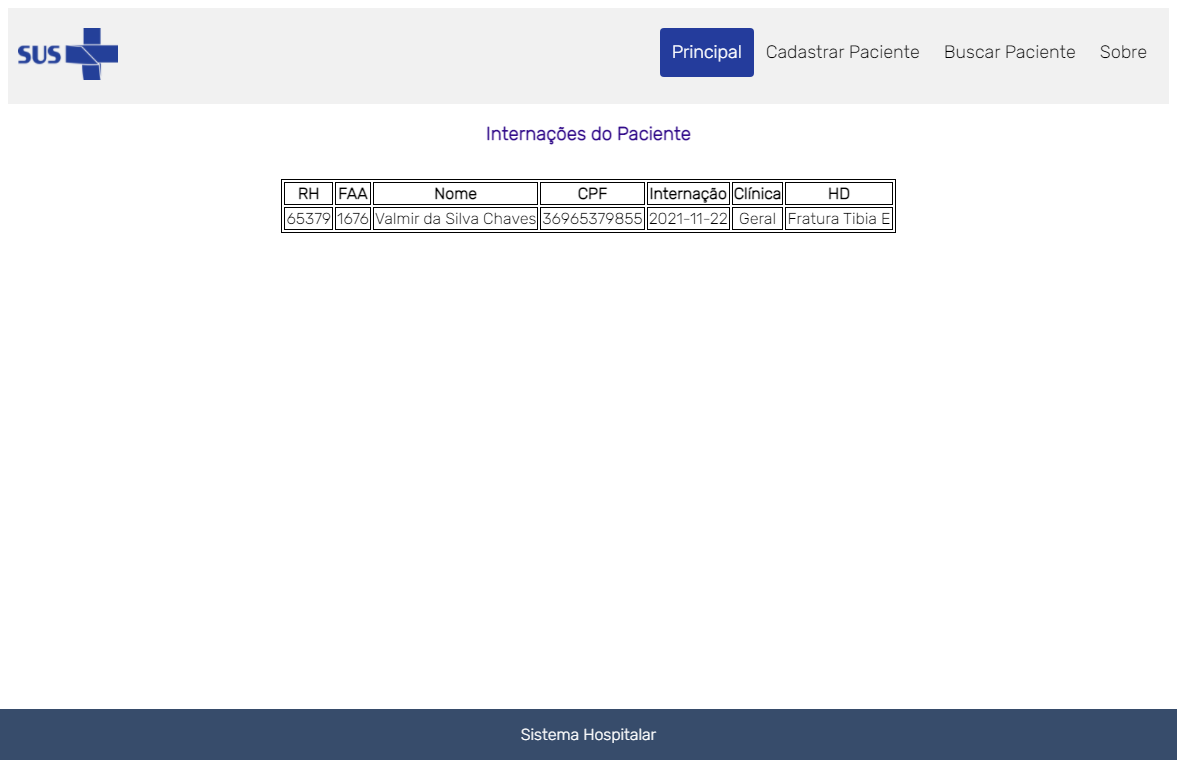
Após o cadastro, o paciente deverá ser buscado novamente para inclusão do atendimento que abrirá uma nova janela para complementação de dados.



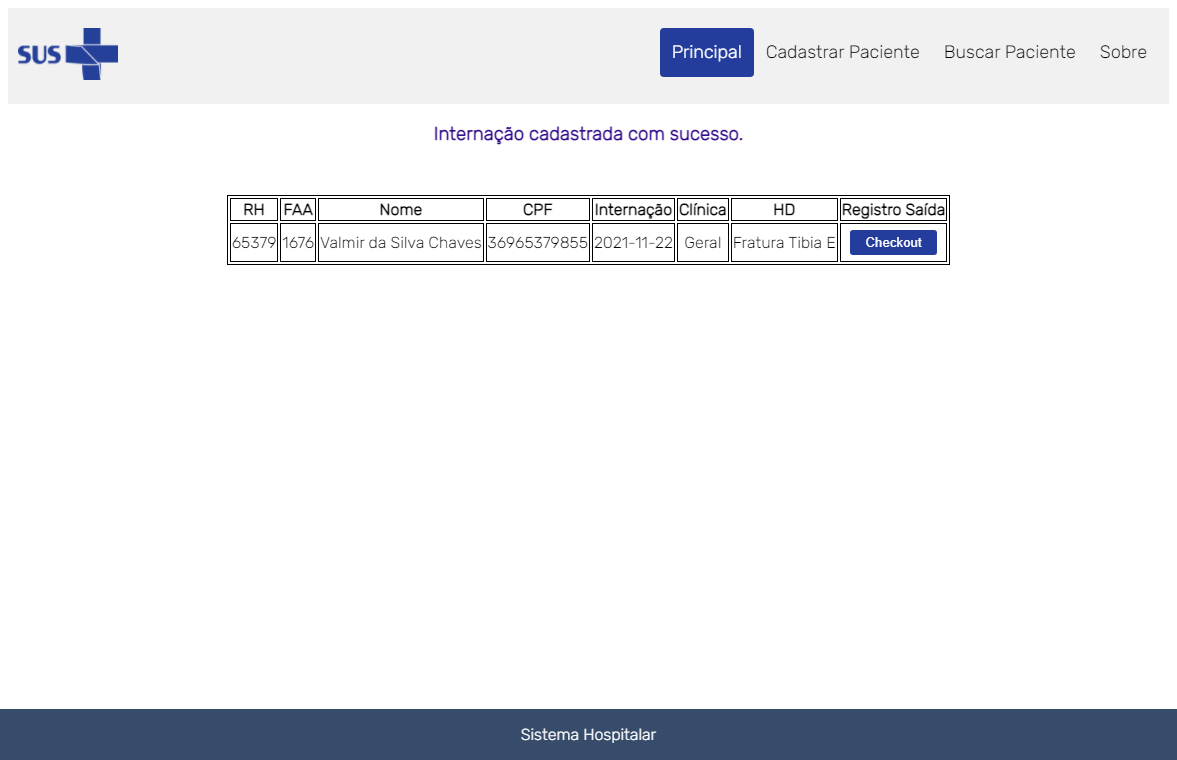
O sistema solicitará os seguintes dados onde serão adicionados a data, setor onde o paciente será atendido e a queixa dele. O mesmo processo deve ser feito para internação com diferencial no campo de queixa que se torna “Hipótese Diagnóstica”.



O banco de dados também armazena o histórico de atendimentos e internações do paciente.



O registro de Alta do paciente pode ser feito através da tela principal onde ficará listado no botão “CHECKOUT”, o que ocorrerá também para os pacientes em atendimento.



O Banco de dados foi fragmentado em uma pasta principal e cinco subpastas, onde são separados os dados, já que o JSON permite que um banco de dados seja incluído dentro de outro como um objeto interno formando um objeto ainda maior mantendo a independência de ambos os bancos.

Tabela

Descrição gerada automaticamente

Com os dois modelos prontos, realizamos o teste de performance para provar o nosso principal tópico deste trabalho. Pudemos analisar que os resultados obtidos pelo modelo relacional foram menos eficazes do que o não-relacional.

Tanto o tempo de mandar a requisição/resposta para o servidor quanto o tempo do retorno acabaram sendo maiores comparado ao não-relacional, o que impacta na ideia que queremos transmitir com este teste: o modelo não-relacional retorna os dados mais rapidamente pois trabalha apenas com objetos JSON, que é um texto legível aos humanos no formato atributo-valor.

O resultado do modelo relacional foi de 0.12ms para mandar a requisição e 1.20s de espera para retornar a resposta.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

O resultado do modelo não-relacional foi de 0.10ms para mandar a requisição e 34.19ms de espera para retornar a resposta.

Texto

Descrição gerada automaticamente

**4. Discussão**

Com base nos resultados apresentados anteriormente, obtivemos uma pequena demonstração da diferença entre os bancos de dados citada no artigo “SQL X NOSQL: ANÁLISE DE DESEMPENHO DO USO DO MONGODB EM RELAÇÃO AO USO DO POSTGRESQL”, onde Marcos André Pereira comenta sobre a incapacidade dos bancos de dados tradicionais de sustentar uma estrutura com abundância de dados ou Big Data. Apesar dos resultados obtidos por nosso protótipo não serem expressivos, não demonstrarem diferença de tempo significativa, em uma estrutura Big Data esta mesma diferença de tempo custará performance do sistema de maneira geral e tempo de sincronização de dados nas requisições trocadas.

//TERMINAR DISCUSSAO

//FAZER CONCLUSAO

**6. Referências bibliográficas**

André Pereira Martins Filho, Marcos. SQL X NOSQL: ANÁLISE DE DESEMPENHO DO USO DO

MONGODB EM RELAÇÃO AO USO DO POSTGRESQL. Universidade Federal de Pernambuco, fev. de 2015. Disponível em: <https://www.cin.ufpe.br/~tg/2014-2/mapmf.pdf>. Acesso em: 15 de jun. de 2021.

Barbara de Oliveira, Saulo. O PLANEJAMENTO ESTRATÉGICO E A VISÃO BASEADA EM RECURSOS (RBV). RECADM, São Paulo SP - Brasil, 2013. Disponível em: <https://doaj.org/article/f956be0b766e463b9cd1470602409223>. Acesso em: 15 de jun. de 2021.

Diego Carlos Farias e Fernando Oliveira de Araújo. Gestão hospitalar no Brasil: revisão da literatura visando ao aprimoramento das práticas administrativas em hospitais, jun. de 2017. Disponível em: [https://www.scielosp.org/article/csc/2017.v22n6/1895-1904/pt/#](https://www.scielosp.org/article/csc/2017.v22n6/1895-1904/pt/). Acesso em: 16 de out. de 2021.

Silveira, Márcio. O BIG DATA E SEU USO CORPORATIVO: UMA REVISÃO DE LITERATURA. SINGEP, out. de 2015. Disponível em: <https://singep.org.br/4singep/resultado/245.pdf>. Acesso em: 15 de jun. de 2021.

Machline, Claude. Rede hospitalar nacional usa indicadores gerenciais na

administração de suas unidades. O Mundo da Saúde, jan. de 2011. Disponível em: <https://bvsms.saude.gov.br/bvs/artigos/rede_hospitalar_nacional_usa_indicadores_gerenciais.pdf>. Acesso em: 15 de jun. de 2021.

Artur de Souza, Antônio. CONTROLE DE GESTÃO EM ORGANIZAÇÕES HOSPITALARES. Revista de Gestão USP, set. de 2009. Disponível em: <https://www.revistas.usp.br/rege/article/view/36675/39396>. Acesso em: 21 de jun. de 2021.