**PROJETO INTEGRADO MULTIDISCIPLINAR IV**

**CURSOS SUPERIORES DE TECNOLOGIA**

**UNIPCARE:**

Cadastro de Pacientes Diagnosticados com COVID-19

**UNIP - PAULISTA**

**SÃO PAULO**

**2021**

**Projeto Integrado Multidisciplinar IV**

**Cursos Superiores de Tecnologia**

**UNIPCARE:**

Cadastro de Pacientes Diagnosticados com COVID-19

**José Nilo da Costa Silva**

**RA: 1750182**

Análise e Desenvolvimento de Sistemas.

4º Semestre

**UNIP - PAULISTA**

**SÃO PAULO**

**2021**

**SUMÁRIO**

[1. INTRODUÇÃO 5](#_Toc73642285)

[2. CONHECENDO O SISTEMA 6](#_Toc73642286)

[2.1. INSTALAR O SISTEMA 6](#_Toc73642287)

[2.2. ACESSANDO O SISTEMA 6](#_Toc73642288)

[2.3. CADASTRAR PACIENTE 7](#_Toc73642289)

[2.3.1. CAMPOS E TIPOS EXIGIDOS 7](#_Toc73642290)

[2.4. LISTAR PACIENTES 10](#_Toc73642291)

[2.5. ALTERAR SENHA 12](#_Toc73642292)

[2.6. ADICIONAR USUÁRIO 13](#_Toc73642293)

[3. PROCESSO DE DESENVOLVIMENTO 14](#_Toc73642294)

[4. CONCLUSÕES FINAIS 17](#_Toc73642295)

[REFERÊNCIAS BIBLIOGRÁFICAS 18](#_Toc73642296)

**RESUMO**

Devido uma pandemia global a necessidade de sistemas de informação para o auxílio na área da saúde aumentou imensamente. A partir disto, uma equipe de alunos da faculdade UNIP desenvolveu o sistema UNIPCARE para cadastrar pacientes diagnosticados com COVID-19.

Sistema desenvolvido em linguagem C, armazena os dados dos pacientes cadastrados em arquivo, e os que são considerados como pertencentes ao grupo de risco têm sua idade e cep escritos em um novo arquivo a parte, este para ser enviado a Secretaria Municipal de Saúde.

Projeto não adotou nenhum processo em específico por ter utilizado o modelo de Codifica-Remenda. Também foi realizado com grande dificuldade, principalmente pela gravação em arquivos e as validações dos dados informados pelos usuários. Houve inclusive a tentativa de adoção de Design Patterns (Padrão de Desenvolvimento e Codificação a fim de evitar ou facilitar a solução de problemas futuros).

O que se tornou notável neste projeto é a importância de prévio conhecimento na linguagem a ser desenvolvida e além disso, um bom planejamento que deve não só ser seguido, como também controlado.

**ABSTRACT**

Due to a global pandemic the need for information systems to aid in the health field has increased immensely. From this, a team of students from the UNIP faculty developed the UNIPCARE system to register patients diagnosed with COVID-19.

System developed in C language, stores the data of registered patients in a file, and those who are considered to belong to the risk group have their age and zip code written in a new separate file, this one to be sent to the Municipal Health Department.

Project did not adopt any specific process for having used the “patch code” model. It was also carried out with great difficulty, mainly by recording in files and validating the data provided by users. There was even an attempt to adopt Design Patterns (Development and Coding Patterns in order to avoid or facilitate the solution of future problems).

What became remarkable in this project is the importance of prior knowledge of the language to be developed and, in addition, good planning that must not only be followed, but also controlled.

# INTRODUÇÃO

Com a pandemia global, investimentos em tecnologias voltadas para a área da saúde se fez essencial, no Brasil em 2020 por exemplo, houve aceleração em vagas de tecnologia e saúde. Assim então surgiu o projeto UNIPCARE, um sistema simples para registrar pacientes diagnosticados com COVID-19, além de verificar se estes fazem parte do grupo de risco, em caso positivo um arquivo é gerado com os dados mais básicos, a fim de compartilhar com a Secretaria Municipal de Saúde.

Este sistema foi desenvolvido em linguagem C, para ser executado em Console do Windows. O modelo do ciclo de vida adotado para o desenvolvimento deste foi o “Codifica-Remenda” por não ser necessária a definição de nenhum processo, principalmente por este ter sido executado por uma única pessoa.

Para garantir o controle de versões das alterações realizadas durante a evolução, foi criado um repositório no GitHub que pode ser acessado através da url <[jnsilva1/Projetos-Unip: Projetos da Faculdade (github.com)](https://github.com/jnsilva1/Projetos-Unip)>.

# CONHECENDO O SISTEMA

## INSTALAR O SISTEMA

A Instalação do sistema é muito simples, basta descompactar o arquivo “UNIPCARE-1.0.zip” no diretório que quiser, e já está pronto para execução.

## ACESSANDO O SISTEMA

Ao executar o sistema, é solicitado que usuário informe os dados para acessar as opções disponíveis (menu). Há um login e senha pré-definidos para o primeiro acesso, login é ADMIN (maiúsculo) e a senha é #UNIP.123 (maiúsculo).

Caso alguma das informações inseridas para acesso esteja incorreta, uma mensagem é exibida alertando ao usuário conforme imagem de evidência abaixo.

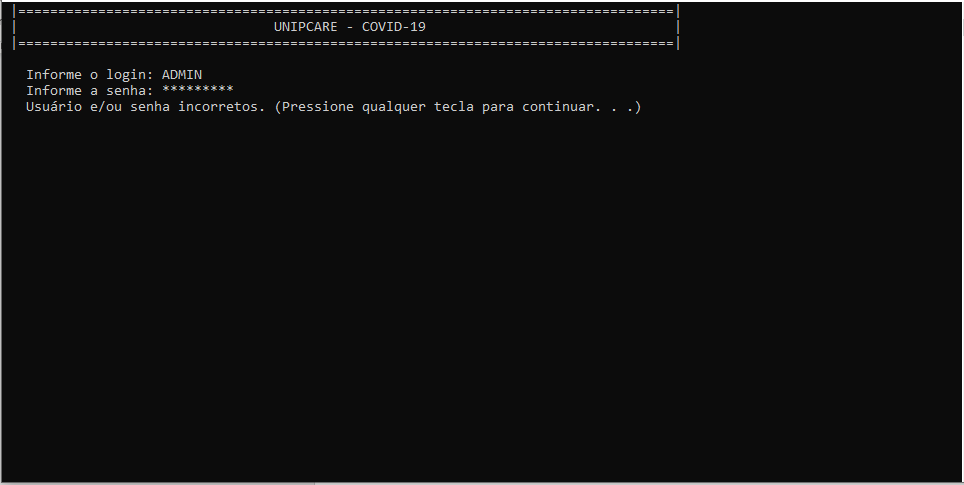


Figura 1 – Dados de Acesso Incorretos

Com acesso realizado com sucesso, é exibido o menu, a última opção do menu deve ser visível apenas para o usuário ADMIN.

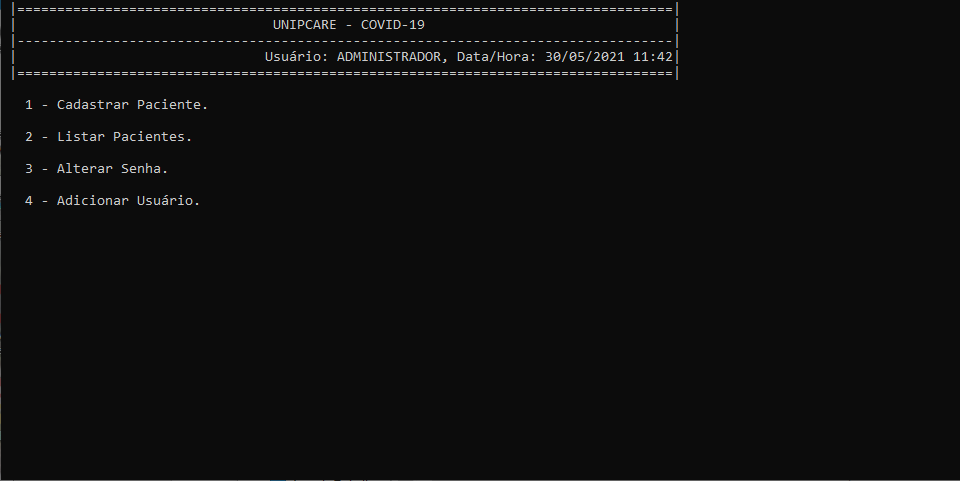


Figura 2 – Menu

## CADASTRAR PACIENTE

Apresentado como opção 1 no menu, o cadastro de pacientes exige determinados campos, para os textuais é indicado a não utilização de acentos ou caracteres especiais pois em alguns casos não serão permitidos.

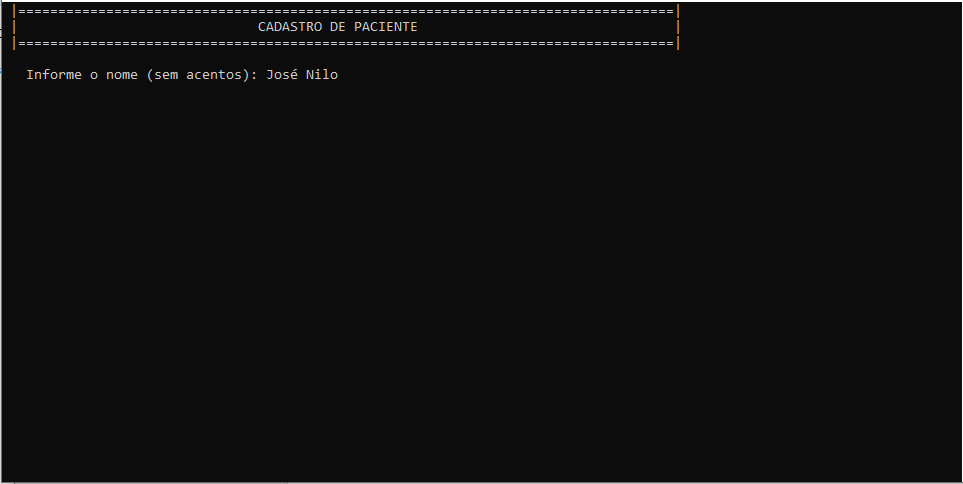


Figura 3 – Entrada de Cadeia de Caracteres com Acentuação

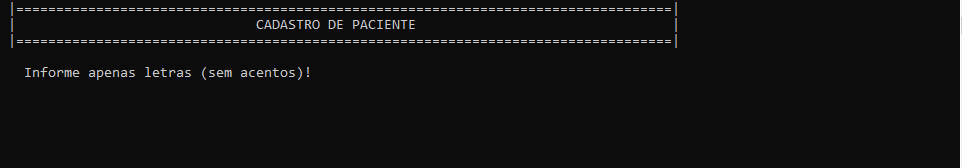


Figura 4 – Mensagem de Erro Referente Figura Anterior

### CAMPOS E TIPOS EXIGIDOS

|  |  |  |
| --- | --- | --- |
| **Descrição** | **Tipo** | **Obrigatório** |
| Nome | Cadeia de Caracteres Sem Acentos (50) | Sim |
| CPF | Numérico “Formatado” | Sim |
| Data Nascimento | Data “dia/mês/ano” | Sim |
| Sexo | Numérico | Sim |
| Altura | Decimal | Sim |
| Peso | Decimal | Sim |
| E-mail | Cadeia de Caracteres (60) | Sim |
| Telefone | Numérico “Formatado” | Sim |
| CEP | Numérico “Formatado” | Sim |
| Logradouro | Cadeia de Caracteres Sem Acentos (60) | Não |
| Número | Numérico | Não |
| Complemento | Cadeia de Caracteres (60) | Não |
| Bairro | Cadeia de Caracteres Sem Acentos (30) | Não |
| Cidade | Cadeia de Caracteres Sem Acentos (40) | Não |
| Estado (UF) | Cadeia de Caracteres Sem Acentos (2) | Não |
| **Comorbidades** | | |
| Diabete | Caractere (1) “S” ou ”N” | Não |
| Obesidade | Caractere (1) “S” ou ”N” | Não |
| Hipertensão | Caractere (1) “S” ou ”N” | Não |
| Tuberculose | Caractere (1) “S” ou ”N” | Não |
| HIV | Caractere (1) “S” ou ”N” | Não |
| Asma ou Doença Pulmonar | Caractere (1) “S” ou ”N” | Não |
| Doença Renal | Caractere (1) “S” ou ”N” | Não |
| Doença no Fígado | Caractere (1) “S” ou ”N” | Não |
| Gestação de Alto Risco | Caractere (1) “S” ou ”N” | Não |
| Transplante | Caractere (1) “S” ou ”N” | Não |
| Quimioterapia | Caractere (1) “S” ou ”N” | Não |
| Diálise | Caractere (1) “S” ou ”N” | Não |

Tabela 1 – Campos de Entrada, Tipo de Valores e Obrigatoriedade

Uma observação referente ao questionamento de comorbidades, o campo ‘Gestação de Alto Risco’ deve ser apresentado apenas para pacientes do sexo feminino.

Para os campos obrigatórios, quando não preenchidos corretamente será apresentada mensagem “Campo obrigatório” ou alguma outra mensagem específica do campo referente a sua validação.

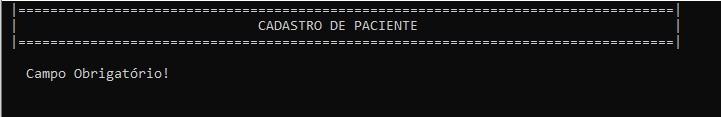


Figura 5 – Mensagem de Erro para Entrada Vazia para o Campo Nome

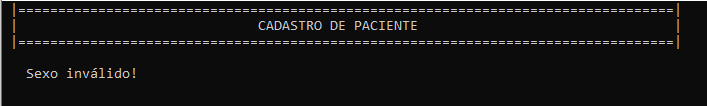


Figura 6 – Mensagem de Erro Personalizada para Entrada Vazia no Campo Sexo

Para os campos do tipo numéricos que possuem a informação “Formatado”, significa que aceita caracteres especiais referentes a formatação do dado informado, e serão removidos automaticamente. Tomemos como exemplo o CPF em que podemos informar das seguintes maneiras: 123.456.789-01 ou 123.456.789.01 ou 12345678901 e mesmo assim será considerado como válido.

Ao final de todo o preenchimento, será apresentada mensagem informando que o paciente foi cadastrado com sucesso!

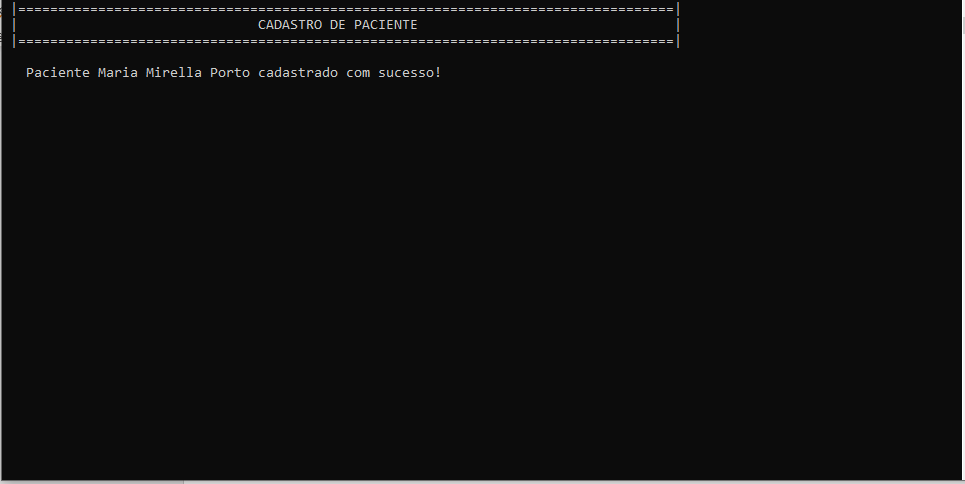


Figura 7 – Mensagem de Gravação do(a) Paciente

Caso o paciente possua alguma comorbidade ou tenha IMC superior a 40 ou idade superior ou igual a 65 anos, este será considerado como parte do grupo de risco, então seu CEP e Idade serão armazenados em um arquivo específico de extensão “.txt” para que possa ser compartilhado com a Secretaria Municipal de Saúde. O Sistema irá informar ao usuário esta condição e o local onde o arquivo foi armazenado.

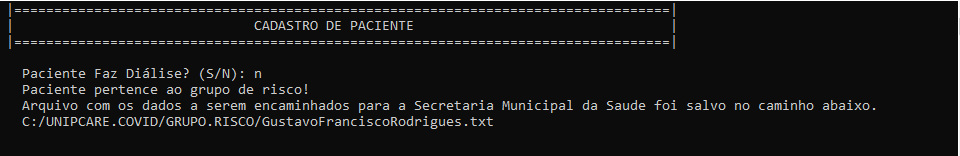


Figura 8 – Mensagem de Gravação do(a) Paciente Grupo de Risco

Após este processo, será questionado ao usuário se deseja cadastrar outro paciente. Caso responda que sim “S”, então será retomado o cadastro, caso contrário serão apresentadas outras opções de ação.

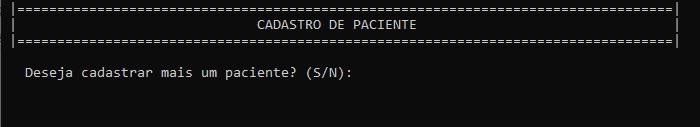


Figura 9 – Questionamento ao Usuário se Deseja Cadastrar outro Paciente

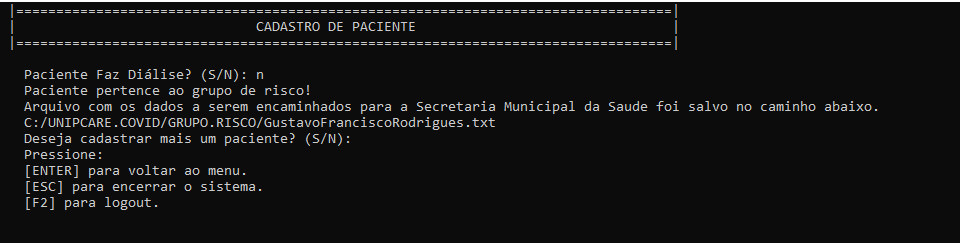


Figura 10 – Opções adicionais de ação

## LISTAR PACIENTES

Apresentado como segunda opção no menu, a listagem de pacientes apresenta de forma ordenada por nome em ascendência. Nesta tela é apresentado todos os dados obtidos durante o cadastro, na seção de comorbidades, serão apresentados apenas aquelas quais foram ditas como sim para o paciente, e caso este(a) não possua, a mensagem “Não possui comorbidades” será apresentada na seção.

A imagem abaixo apresenta uma evidência de listagem de dois pacientes, um do sexo masculino e outro feminino, um deles possui comorbidades e sendo assim faz parte do grupo de risco.

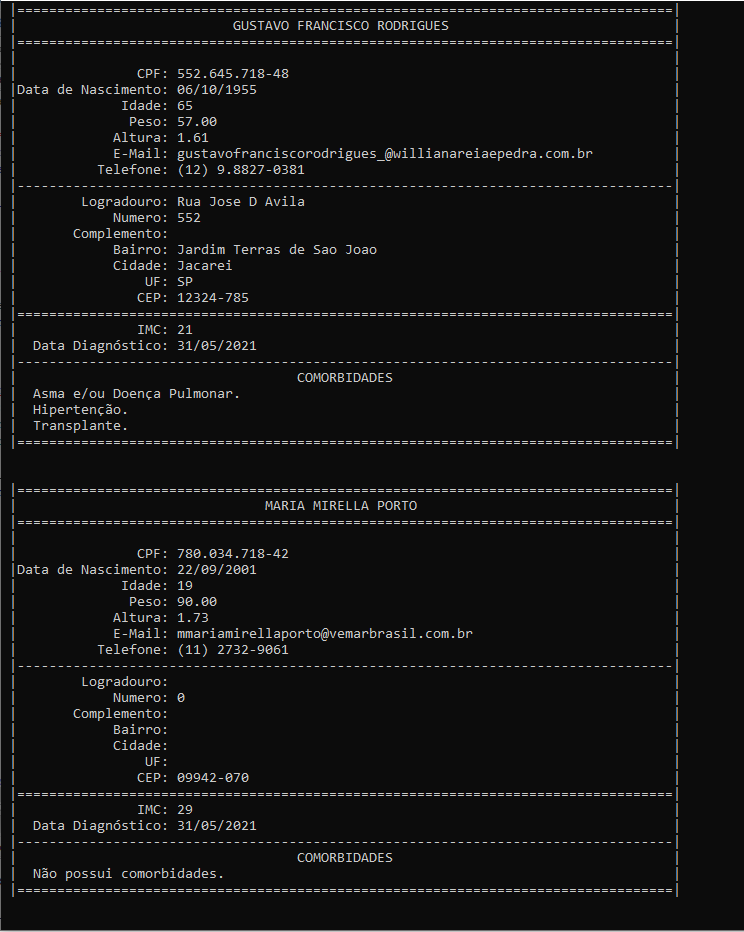


Figura 11 – Listagem de Pacientes Cadastrados (Paciente 1)

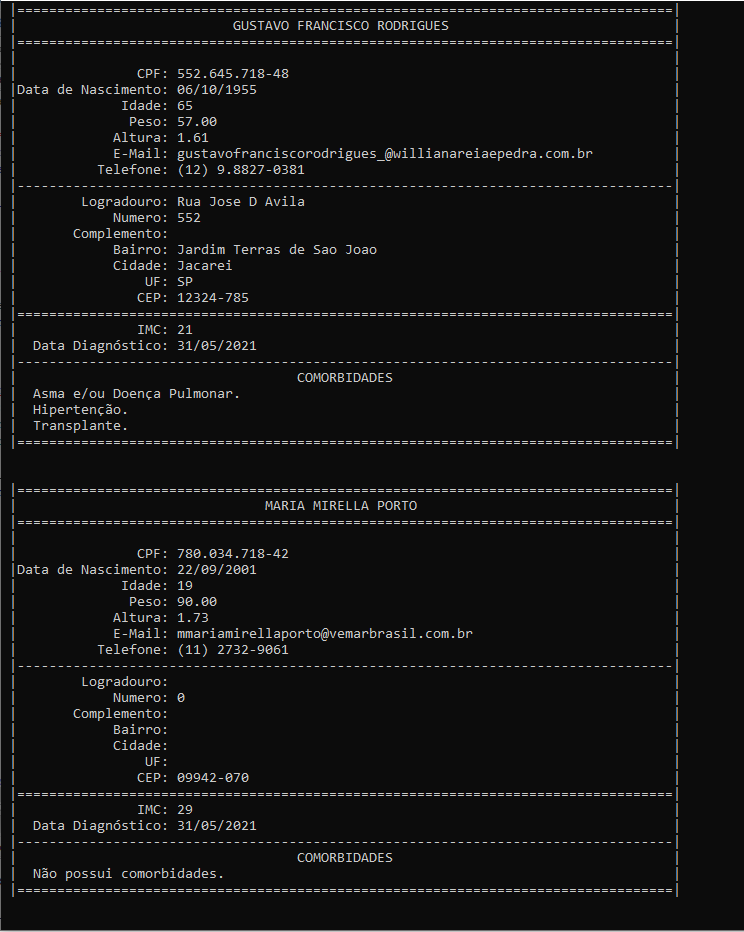


Figura 12 – Listagem de Pacientes Cadastrados (Paciente 2)

Assim como no final do processo de cadastro, no término da listagem é apresentada as opções de ação ao usuário.

## ALTERAR SENHA

Para que o usuário tenha sua liberdade de escolha da senha, foi criado a tela para realizar a alteração. Os campos quais preenchimentos serão exigidos são: senha atual, nova senha e confirmação de senha.

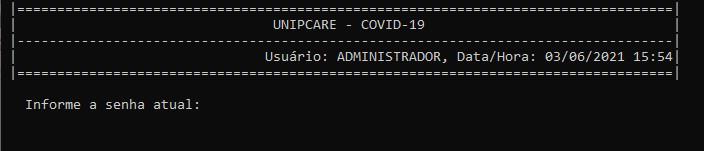


Figura 13 – Alteração de Senha, Campo: Senha Atual

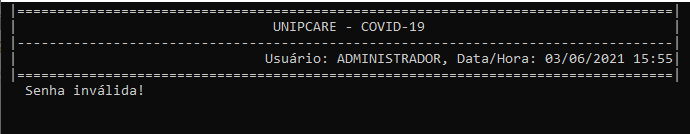


Figura 14 - Alteração de Senha, Mensagem Erro Referente Senha Atual Informada

Ao Informar a nova senha, será solicitado que confirme a inserindo-o novamente. Caso a nova senha não coincida com a de confirmação, será apresentado mensagem de erro informando o corrido, e será exigido que repita o preenchimento destes dois últimos campos.

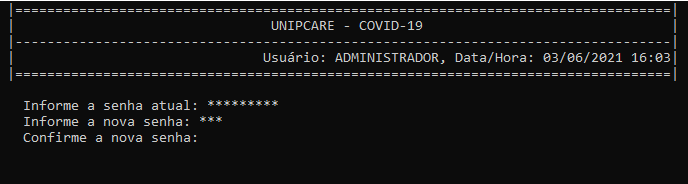


Figura 15 - Solicitado Nova Senha e a Confirmação

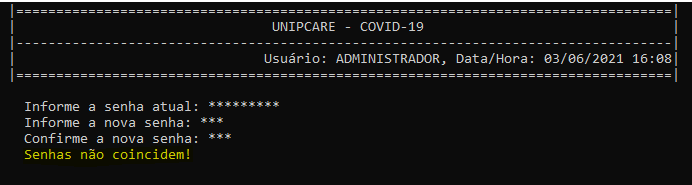


Figura 16 – Mensagem Erro Referente Nova Senha e Confirmação não Coincidirem

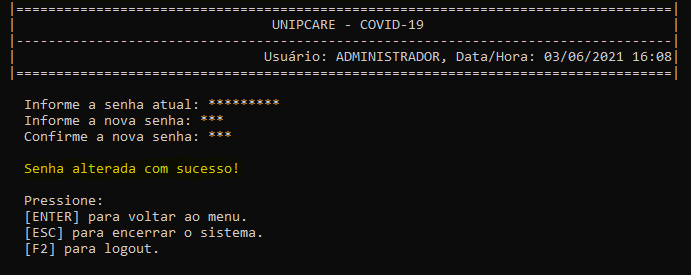


Figura 17 – Mensagem da Confirmação de Senha Alterada

Uma ressalva referente as figuras 16 e 17, para as mensagens destacadas na cor amarela, essa cor foi alterado apenas na imagem, no sistema a cor permanece a mesma que as demais.

## ADICIONAR USUÁRIO

A fim de permitir que mais usuários tenham acesso ao sistema, foi criado a tela para realizar este cadastro. Apenas o usuário ADMIN deve ter permissão para realizar esta operação.

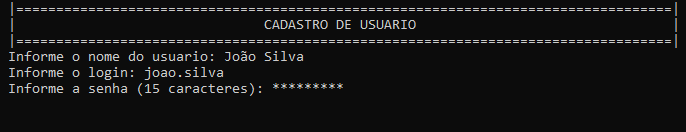


Figura 18 – Cadastro de Usuário

Após inserir os dados no novo usuário, será questionado se deseja adicionar mais.

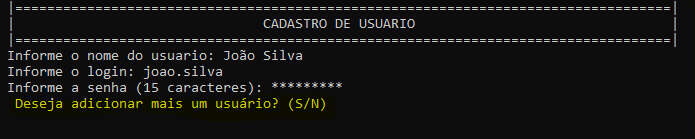


Figura 19 – Questionamento Referente a Intenção de Adicionar Outro Usuário

Caso informe a entrada seja diferente de (S) ou (s), então será apresentada as demais opções de ação para o usuário bem como ocorre na figura 10.

# PROCESSO DE DESENVOLVIMENTO

Conforme dito anteriormente, este sistema não passou por planejamento prévio e nenhum processo em específico foi adotado, o modelo do ciclo de vida foi o mesmo comumente utilizado por vários outros projetos, no entanto muito embaraçoso e mais ariscado, o codifica-remenda.

Este modelo foi adotado principalmente pelo fato de a equipe de desenvolvimento ser composta por uma única pessoa, além disto, esta não tem conhecimento profundo na linguem C, sendo assim, conforme a necessidade foi estudado componentes específicos para a solução de um problema por vez.

Durante todo o desenvolvimento, surgiram problemas que precisaram de muito tempo para ser estudado e então desenvolvido uma solução, a principal delas foi gravar os dados do paciente em arquivo e recuperá-las posteriormente.

O desenvolvedor por estar muito habituado a utilizar a linguagem derivada C#, e a orientação a objetos, utilizou, portanto, estruturas de dados (struct) para se sentir mais confortável na codificação. Houve a tentativa de aplicação do Design Patterns S.O.L.I.D., no entanto não foi possível por não se tratar da possibilidade do uso de classes nesta linguagem, ainda sim foi implementado a responsabilidade única, em que cada estrutura tem sua responsabilidade única e cada função responsável pela execução de um processo específico.

Na especificação de requisitos, há a exigência para a validação da entra de valores dos usuários, e assim como para realizar a gravação dos dados em arquivo, esta validação foi muito complicada para ser implantada. Seria mais fácil apenas receber os dados da entrada e gravar, mas validar se o usuário realmente digitou apenas números em campos que exigia apenas esse tipo de dado foi mais complexo do que apenas utilizar a função “scanf” e a formatação “%d”, pois desta forma não impediria de o usuário informar um caractere alfabético e ser interpretado como seu valor ASCII correspondente.

Para isto foi criado uma função a fim receber uma cadeia de caracteres, a partir disso, seria validado se o valor informado corresponde a um tipo numérico ou não, foi criado também uma função para validar se o usuário informou uma cadeia de caracteres e que em nenhum momento foi informado um caractere especial ou com acentuação.

Então basicamente foi criado uma função para obter cada tipo de valor e realizar sua devida validação. Com isso pode surgir o questionamento “Como foi realizado a conversão do valor de ‘string’ para inteiro, longo ou double?”, bem todas essas conversões se iniciaram a partir do tipo inteiro, primeiro é verificado se a cadeia contém apenas valores do tipo numérico, caso negativo retorna o valor zero como resultado da conversão, em caso positivo, é realizado a conversão de cada caractere da direita para a esquerda seguindo a fórmula abaixo para cada iteração.

*valorAcumulado = valorAcumulado + ((caracteres[indice] - 48) \* (10 ^ qtdeIteracao))*

Abaixo está a imagem que demonstra a função responsável por esta conversão.

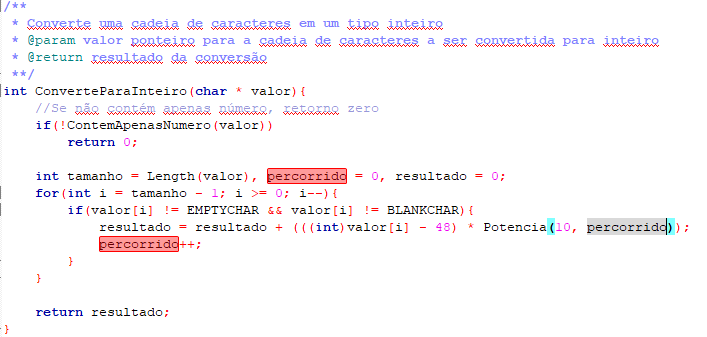


Figura 20 – Função Responsável pela Conversão para Inteiro

Para o tipo longo, a lógica utilizada foi a mesma, já o double foi realizado a quebra deste em duas partes e convertido para inteiro, só depois unidos como double, a parte decimal é dividido pelo resultado da potência de 10 elevado a quantidade de caracteres contidos nesta parte.

# CONCLUSÕES FINAIS

Este projeto demonstrou a importância de aprimoramento de tecnologias para uso na área da saúde a fim de facilitar a distribuição/compartilhamento de informações, ajudar na tomada de decisão, como no caso da Secretaria Municipal da Saúde que ao receber diversas informações de pacientes pertencentes ao grupo de risco teve que planejar as medidas a serem adotadas para evitar a disseminação do vírus.

Foi possível notar também a importância de conhecimento profundo em linguagem específica para que se possa ter melhor desenvolvimento da aplicação desejada. Além disso, para todo projeto é importante uma grande equipe ou pelo menos um bom planejamento, que além de seguido deve ser controlado a fim de realizar melhorias no processo e mudanças ágeis no plano e Análise de Requisitos não acaba na primeira reunião, é constante durante todo o clico de vida do projeto.

# REFERÊNCIAS BIBLIOGRÁFICAS

Engenharia de Software I, Ivanir Costa. – São Paulo: Editora Sol, 2014.

<https://stackoverflow.com/>

Mizrahi, Victorine Viviane. Treinamento em Linguagem C 2ª Edição. Editora Pearson, 2008. [[Biblioteca Virtual (bvirtual.com.br)](https://plataforma.bvirtual.com.br/)]

Manzano, José Augusto N. G. Estudo Dirigido de Linguagem C. Editora Saraiva, 2013. [[Minha Biblioteca](https://integrada.minhabiblioteca.com.br/books/9788536519128)].