



Universidade Federal do Espírito Santo  
Departamento de Informática

## 2ª Trabalho Prático

### Mini-SADE *revival*

Programação II - 2023/1

09 de Novembro de 2023

**IMPORTANTE:** no dia 02/12/2023 diversas atualizações foram realizadas na descrição do trabalho. Verifique o classroom da turma para mais informações.

**Principais alterações:**

- **Busca de pacientes**
  - A busca deve ser realizada apenas pelo nome
  - Não é necessário ordem alfabética para apresentar a lista
  - Foi removida a opção de visualizar paciente
- **O relatório geral foi simplificado**
  - Foi removido o desvio padrão e porcentagem
  - Foi removido a distribuição das lesões
- **Fila de impressão**
  - Foi removido a necessidade de persistir a fila em um arquivo binário (não existe mais banco para fila)
  - Também foi removido o documento visualizacao.txt
- **Telas de menu**
  - Médico são capazes de gerar relatório geral e executar fila de impressão
  - Alterações na tela de busca para refletir as mudanças anteriores
- **Funcionamento do programa**
  - É necessário concatenar a string `saida` no caminho informado via argumento

**Errata 1:** correção do nome dos arquivos de impressão no exemplo da subseção de Tela de Fila Impressão. De (receita\_0007.txt e lista\_busca\_0009.txt) para (receita.txt e lista\_busca.txt)

**Errata 2:** adição de uma observação para esclarecer o acesso de um ADMIN na subseção de Tela de acesso

**Errata 3:** inclusão da solicitação de nome de usuário e senha para cadastro de um médico (Página 14)

**Errata 4:** correção da idade para data de nascimento no item 2 da funcionalidade 2 e inclusão do tipo de medicamento na tela de receita médica

## **Objetivo**

O objetivo deste trabalho é colocar em prática as habilidades de programação na linguagem C adquiridas ao longo do curso. O trabalho foi elaborado para que você exercite diversos conceitos principalmente a construção de TADs opacos e genéricos, alocação e manipulação de memória de maneira dinâmica, e manipulação de arquivos binários. Como consta no plano de ensino da disciplina, o trabalho será realizado em duas etapas (a primeira realizada em casa e a segunda em sala de aula), sendo esta a primeira etapa. O somatório de ambas as etapas **equivale a 50% da nota final do curso**.

## **Introdução**

A extensão universitária é uma atividade complementar à formação acadêmica oferecida pelas universidades e instituições de ensino superior. Ela tem como objetivo estender os conhecimentos gerados no ambiente acadêmico para a sociedade em geral, promovendo a interação entre a instituição de ensino e a comunidade. A UFES possui centenas de projetos de extensão, um deles é o Programa de Assistência Dermatológica (PAD-UFES), que oferece atendimento gratuito à população Capixaba para tratamento de lesões de pele, desde a triagem até a cirurgia, se necessário. Para digitalizar todo o atendimento dos pacientes atendidos pelo PAD, em 2021 foi criado o PADTech, o núcleo de tecnologia do PAD na qual um dos objetivos é desenvolver e manter um sistema de coleta, armazenamento e análise de todos os dados obtidos durante um atendimento do PAD. Atualmente, o PADTech é um projeto de extensão ligado ao departamento de Informática e que possui bolsistas e voluntários que trabalham na criação do Software de Análise Dermatológica (SADE).

Sendo assim, neste trabalho, sua missão **é implementar um protótipo do SADE** (na qual chamaremos de **mini-SADE**) que tem como objetivo realizar algumas das tarefas que são necessárias para habilitar um atendimento dos profissionais de saúde associados ao PAD-UFES. Vale destacar que o trabalho abrange apenas uma pequena parte do sistema que está em produção. Para aqueles que tiverem curiosidade do funcionamento completo, o PADTech funciona como um projeto associado ao Laboratório de Inteligência Artificial em Saúde, no CT-XIII na sala 27.

## **Descrição**

A intenção principal do mini-SADE é gerenciar digitalmente o atendimento de pacientes em uma clínica fictícia. Para isso, deve ser considerado as seguintes premissas:

- A clínica possui três atores principais: secretário, médico e paciente
- Um secretário e/ou médico acessam o sistema mediante login e senha
- Um secretário cadastra pacientes para serem atendidos por um médico
- Somente pacientes pré-cadastrados podem realizar consultas
- Um paciente pode realizar uma ou mais consultas e cada consulta está associado a um médico, que pode ser diferente
- Durante a consulta, informações clínicas do paciente serão preenchidas
- Durante uma consulta podem ser encontradas nenhuma ou várias lesões
- Uma lesão pode ser um caso simples, sem necessidade de um tratamento, ou pode ser encaminhada para crioterapia ou cirurgia
- Todas as informações da clínica devem persistir em um sistema de banco de dados
- Médicos e secretários devem ser capazes de recuperar informações dos bancos de dados
- Diversos documentos podem ser gerados a partir do banco de dados
- Documentos podem ser enviados para uma fila de impressão, que vai imprimir o documento solicitado

Dessa forma, o mini-SADE deve ser capaz de fornecer as seguintes funcionalidades:

1. Cadastro de secretários, médicos e pacientes
2. Realização de uma consulta médica
3. Busca de pacientes
4. Relatório geral
5. Fila de impressão de documentos

Além disso, é esperado que o sistema implemente a persistência dos dados. Isso quer dizer que a base de dados deve ser salva e reutilizada independente da execução do código.

## **Funcionalidade 1: cadastro de secretários, médicos e pacientes**

A primeira funcionalidade do mini-SADE é o cadastro de atores do sistema. Os atores possuem dados em comum e dados específicos. Em comum, todos eles possuem:

- Nome completo (string com no máximo 100 caracteres)
- CPF (no formato: 000.000.000-00)

- Data de nascimento (seguindo padrão dd/mm/aaaa)
- Telefone (formato: (00)00000-0000)
- Gênero (masculino, feminino, outros)

**O CPF é o identificador único de um ator.** O sistema não deve permitir cadastro de atores do mesmo tipo com um mesmo CPF. Por exemplo, não é permitido o cadastro de dois médicos com o mesmo CPF. Porém, o médico também pode ser um paciente. Neste caso, é permitido um cadastro para o papel médico e outro para o papel paciente. Caso seja realizada uma tentativa de cadastro com o mesmo CPF para o mesmo tipo o sistema deve informar: `CPF JA EXISTENTE. OPERACAO NAO PERMITIDA.`

Além das informações acima, médicos e secretários possuem também:

- Nome de usuário (string com no máximo 20 caracteres)
- Senha (string com no máximo 20 caracteres)

Essas informações são utilizadas para acessar o sistema. Um secretário também possui um nível de acesso, que pode assumir duas possibilidades: `USER` ou `ADMIN`. Por sua vez, um médico possui um CRM (string com no máximo 12 caracteres no padrão CRM-0000000), que é a permissão de atuação do médico.

O relacionamento entre os atores do sistema é o seguinte:

- Somente secretários e médicos possuem acesso ao sistema
- Um secretário do tipo `ADMIN` pode cadastrar outros secretários, médicos e pacientes
- Um secretário do tipo `USER` pode cadastrar apenas médicos e pacientes
- Médicos acessam o sistema para realizar consultas dos pacientes
- Pacientes não possuem acesso ao sistema

## Funcionalidade 2: realização da consulta

Um médico deve ser capaz de realizar uma consulta de um paciente previamente cadastrado no sistema. Não é possível realizar consulta de pacientes sem cadastro. Para isso, a consulta se inicia com o médico digitando um CPF. Em seguida, podem ocorrer duas situações:

1. **O número do CPF não está cadastrado:** neste caso, a consulta não será iniciada e a mensagem `PACIENTE SEM CADASTRO` deve ser exibida na tela.
2. **O número do CPF está cadastrado:** neste caso, o sistema deve exibir na tela o nome e a **data de nascimento** do paciente e a consulta pode ser iniciada.

Uma consulta deve conter a data (seguindo padrão dd/mm/aaaa) em que ela ocorreu, o médico que a realizou e nenhuma ou várias lesões que foram descobertas durante a consulta.

Durante a consulta, o médico coleta informações clínicas dos pacientes:

- Se o paciente possui diabetes (inteiro assumindo 1 ou 0)
- Se o paciente é fumante (inteiro assumindo 1 ou 0)
- Se possui alergia a algum medicamento (inteiro assumindo 1 ou 0)
- Se o paciente possui histórico de câncer (inteiro assumindo 1 ou 0)
- Qual o tipo de pele do paciente
  - Valores possíveis: (I, II, III, IV, V, VI)

Coletada essas informações, o médico deve ser capaz de cadastrar uma ou mais lesões relacionadas ao paciente. Uma lesão possui os seguintes atributos:

- Rótulo (string com no máximo 5 caracteres seguindo o padrão L1, L2, L3, ..., Ln)
  - Deve ser incrementado automaticamente à medida que as lesões são adicionadas ao paciente
- Diagnóstico clínico
  - Valores possíveis: CARCINOMA BASOCELULAR, CARCINOMA ESPINOCELULAR, MELANOMA, CERATOSE ACTINICA, NEVO, CERATOSE SEBORREICA e OUTROS
- Região do corpo
  - Valores possíveis: FACE, ORELHA, COURO CABELUDO, PEITORAL, DORSO, ABDOME, BRAÇO, ANTEBRAÇO, MAO, COXA, CANELA, PE e OUTROS
- Tamanho (inteiro representando o tamanho em milímetros)
- Se a lesão deve ser enviada para cirurgia (inteiro assumindo 1 ou 0)
- Se a lesão deve ser enviada para crioterapia (inteiro assumindo 1 ou 0)

Ao final do cadastro de todas as lesões, o médico deve ser capaz de criar e enviar documentos para fila de impressão. Os documentos possíveis são: **receita médica**, **solicitação de biópsia**, e **encaminhamento**. Informações sobre os documentos são fornecidas na sequência. Todo atendimento deve ser registrado no banco de dados. Mais informações sobre essa parte são fornecidas na seção **Funcionamento do Programa**.

### Documentos relacionados a consulta

Como já mencionado, antes de finalizar a consulta, o médico pode criar e enviar documentos para a fila de impressão. Ele pode enviar:

- Nenhuma ou várias receitas médicas
- Nenhum ou vários encaminhamentos

- Nenhuma ou uma solicitação de biópsia

Todos os documentos estão relacionados à consulta que o médico está realizando no momento. O conteúdo de cada documento é descrito a seguir:

**Receita médica:** documento com uma receita a ser fornecida pelo paciente com as seguintes informações:

- Nome do paciente
- Tipo de uso (ORAL ou TOPICO)
- Nome do medicamento (string com no máximo 50 caracteres)
- Tipo de medicamento (string com no máximo 50 caracteres)
- Quantidade (inteiro)
- Instruções de uso (string com no máximo 300 caracteres)
- Nome do médico, CRM e data

Observe que este documento mescla dados já preenchidos (ex: nome do paciente) com dados que ainda precisam ser preenchidos (ex: tipo de uso).

Exemplo:

Receita médica

RECEITUARIO

NOME: LUCIAN (nome do paciente)

USO ORAL (tipo de uso)

DIPIRONA 500MG (nome do medicamento)

12 (quantidade) COMPRIMIDO(S) (tipo de medicamento)

TOMAR 01 COMPRIMIDO E 6 EM 6 HORAS (instruções de uso)

DR. MUNDO (nome do médico) (CRM-132456) (CRM do médico)

10/10/2023 (data da consulta)

**Solicitação de biópsia:** documento que deve ser gerado para todas as lesões encaminhadas para cirurgia. Contém as seguintes informações:

- Nome do paciente
- CPF do paciente
- Rótulo da(s) lesão(ões)
- Diagnóstico da(s) lesão(ões)
- Tamanho da(s) lesão(ões)

- Nome do médico, CRM e data

Neste caso, o médico não precisa preencher mais nada, todos os dados já foram coletados durante a consulta e devem ser usados para gerar o documento de forma automática (ou seja, ele apenas solicita e o documento é gerado).

Exemplo:

Solicitação de biópsia

PACIENTE: RENATA GLASC (nome do paciente)

CPF: 556.889.664-85 (CPF do paciente)

SOLICITACAO DE BIOPSIA PARA AS LESOES:

L1 (rótulo da lesão) - CARCINOMA ESPINOCELULAR (diagnóstico da lesão) -  
FACE (região da lesão) - 30MM (tamanho da lesão)

L5 - MELANOMA - PERNA - 40MM

DR. MUNDO (nome do médico) (CRM-132456) (CRM do médico)

10/10/2023 (data da consulta)

Observe que somente lesões que foram enviadas para cirurgia podem aparecer na solicitação. É obrigatório ao menos uma lesão. Se nenhuma lesão for enviada para cirurgia, não deve ser possível gerar o documento.

**Encaminhamento:** documento contendo um encaminhamento para outra especialidade. Contém as seguintes informações:

- Nome do paciente
- CPF do paciente
- Especialidade encaminhada (string com no máximo 50 caracteres)
- Motivo do encaminhamento (string com no máximo 300 caracteres)
- Nome do médico, CRM e data

Este também é um documento que precisa de entradas novas para ser produzido.

Exemplo:

Encaminhamento

PACIENTE: RENATA GLASC (nome do paciente)

CPF: 556.889.664-85 (CPF do paciente)

ESPECIALIDADE ENCAMINHADA: CARDIOLOGIA (especialidade a ser encaminhada)

MOTIVO: PACIENTE NECESSITA DE AVALIACAO CARDIOLOGICA PARA REALIZAR UMA CIRURGIA (motivo para encaminhamento)

DR. MUNDO (nome do médico) (CRM-132456) (CRM do médico)  
10/10/2023 (data da consulta)

### Funcionalidade 3: busca de pacientes

O sistema deve permitir a realização de busca de pacientes cadastrados (tanto para médicos, quanto para secretários) na base de dados a partir das seguintes entradas:

- Nome do paciente
  - Retornar todos os pacientes com o nome informado
- Data da consulta
  - Retornar todos pacientes que realizaram consulta nesta data
- Diagnóstico
  - Retornar todos pacientes que possuem uma lesão com este diagnóstico

O usuário pode utilizar uma ou mais entradas para determinar o filtro. Por exemplo, se o usuário utilizar apenas o nome do paciente, o sistema deve buscar todos os pacientes que possuem o nome informado. Porém, se ele usar dois ou mais campos, o sistema deve realizar a busca considerando todos os campos utilizados — para não utilizar um campo basta deixá-lo em branco. Por exemplo, um paciente com nome LUCIAN, consultado na data 10/10/2023 e com diagnóstico OUTROS. O(s) paciente(s) que atenderem a todos os parâmetros devem ser listados na tela em ordem alfabética — incluindo nome e CPF — e com um valor numérico associado a cada um deles. Por exemplo:

O usuário realizará a busca pelo nome e se houver nomes repetidos de pacientes eles devem ser listados seguindo a ordem de inserção no banco. Por exemplo, se existem 3 pacientes de nome LUCIAN eles devem ser listados da seguinte forma:

1 - LUCIAN (556.889.664-85)  
2 - LUCIAN (121.568.114-32)  
3 - LUCIAN (456.987.564-96)

O número na frente do nome reflete a ordem de cadastro no banco, ou seja, o paciente LUCIAN com CPF 556.889.664-85 foi cadastrado antes dos demais.

Na sequência, deve ser possível realizar duas operações:



- Enviar lista para impressão
- ~~Visualizar paciente~~

### Enviar lista para impressão

Ao escolher esta opção, o usuário transforma esta lista em um documento e a envia para impressão. Na sequência ele deve ser capaz de voltar para o menu inicial.

### ~~Visualizar paciente~~

~~Ao escolher a opção de visualizar paciente, o usuário poderá escolher um paciente da lista utilizando o número identificador. Utilizando o exemplo anterior, se ele escolher 2, deve ser exibido na tela as seguintes informações do paciente:~~

~~NOME: LUCIAN~~  
~~DATA DE NASCIMENTO: 19/11/1990 (32 ANOS)~~  
~~GENERO: MASCULINO~~  
~~TELEFONE: (27) 99898 3131~~  
~~DIABETES: SIM~~  
~~FUMANTE: NAO~~  
~~ALERGIA A MEDICAMENTO: SIM~~  
~~HISTORICO DE CANCER: NAO~~  
~~TIPO DE PELE: III~~

~~LESOES:~~  
~~TOTAL: 3~~  
~~ENVIADA PARA CIRURGIA: 1~~  
~~ENVIADA PARA CRIOTERAPIA: 1~~

~~DESCRICAO DAS LESOES:~~  
~~L1 - CARCINOMA ESPINOCELULAR - FACE - 30MM - ENVIADA PARA CIRURGIA~~  
~~L2 - CERATOSE ACTINICA - ANTEBRACO - 90MM - ENVIADA PARA CRIOTERAPIA~~  
~~L3 - CERATOSE SEBORREICA - FACE - 40MM~~

~~Se o paciente não possui lesão cadastrada, toda parte de LESOES e DESCRICAO DAS LESOES devem ser omitidas.~~

~~Após visualizar o paciente, o usuário poderá converter a visualização em um documento e enviar para impressão ou voltar para o menu inicial.~~

## Funcionalidade 4: relatório geral

Esta funcionalidade do mini-SADE deve basicamente criar um relatório geral dos dados coletados ao longo dos atendimentos. Este relatório é consolidação geral do banco de dados e deve fornecer as seguintes informações:

- Número total de pacientes **que possui pelo menos um atendimento**
- Média ~~e desvio padrão~~ da idade dos pacientes
  - Utilizando somente a parte inteira
- ~~Distribuição percentual dos~~ **Quantidade de** pacientes por gênero
  - Na ordem: feminino, masculino e outros
  - Utilizando somente a parte inteira
- Tamanho médio ~~e desvio padrão~~ das lesões
  - Utilizando somente a parte inteira
- Número total de lesões
- Número total de cirurgias
  - ~~◦ Incluindo porcentagem em relação ao número total~~
  - Utilizando somente a parte inteira
- Número total de crioterapias
  - ~~◦ Incluindo porcentagem em relação ao número total~~
  - Utilizando somente a parte inteira
- ~~• Distribuição absoluta e percentual dos diagnósticos clínico~~
  - ~~◦ Utilizando ordenação decrescente de acordo com o número absoluto~~
  - ~~◦ Se houver empate, usar ordem alfabética~~
  - ~~◦ Utilizando somente a parte inteira do percentual~~

### Exemplo

relatório

NUMERO TOTAL DE PACIENTES ATENDIDOS: 1  
IDADE MEDIA: 43  
DISTRIBUICAO POR GENERO:  
- FEMININO: 1  
- MASCULINO: 0  
- OUTROS: 1  
TAMANHO MEDIO DAS LESOES: 10  
NUMERO TOTAL DE LESOES: 1  
NUMERO TOTAL DE CIRURGIAS: 1  
NUMERO TOTAL DE CRIOTERAPIA: 0

Após exibir o relatório, o usuário deve ser capaz de enviar o relatório para impressão ou retornar para o menu principal.

## Funcionalidade 5: Fila de impressão de documentos

A fila de impressão, como o próprio nome sugere, nada mais é do que um *event listener* que fica vigiando se alguma impressão está esperando na fila e executa sua impressão. Basicamente, todas as impressões de documentos solicitadas no sistema são enviadas para um banco de dados.

Quando o usuário ativar a fila de impressão, o programa verificará se existe algum documento **na fila em memória** ~~no banco de dados~~ que deve ser impresso. Se sim, a fila imprime todos na sequência. Os novos documentos devem ser impressos na tela (usando `printf` e seguindo os padrões e já especificados) e, para simular uma impressora, devem ser escritos em arquivos de extensão `.txt` dentro de uma pasta informada como argumento para seu programa (ver próxima seção).

O nome dos arquivos a serem impressos são apresentados a seguir:

- **Receita médica:** `receita.txt`
- **Solicitação de biópsia:** `biopsia.txt`
- **Encaminhamento:** `encaminhamento.txt`
- **Resultado de lista de busca:** `lista_busca.txt`
- ~~**Visualização de paciente:** `visualizacao.txt`~~
- **Relatório geral:** `relatorio_geral.txt`

Todos os documentos devem ser impressos dentro do arquivo que o representa anexando no final do arquivo caso exista algum conteúdo no mesmo. Por exemplo, imagine que foram enviadas 3 receitas para impressão. Todas as receitas serão salvas dentro de `receita.txt`, sendo que a 1ª receita da fila ficará no topo do arquivo e a última no final. As receitas devem ser separadas por duas quebras de linha dentro do arquivo. Se não houver nada a ser impresso, o arquivo deve ser criado vazio.

## Banco de dados

Como já descrito, o sistema deve ser capaz de persistir os dados cadastrados no sistema. Para isso, devem ser utilizados arquivos binários para simular um sistema de banco de dados, que deve conter um arquivo binário para cada entidade do sistema. São eles:

- `secretarios.bin`
- `medicos.bin`
- `pacientes.bin`
- `consultas.bin`
- `lesoes.bin`
- ~~`fila_impressao.bin`~~

O caminho para a pasta de banco de dados será informado como parâmetro ao rodar o programa. O sistema pode ser criado a partir de um banco já existente ou pode criar o banco do zero. Por conta disso, é necessário, ao iniciar o sistema, verificar se o banco de dados deve ser criado ou carregado (basicamente, se o arquivo .bin existe, significa que já existe um banco de dados para a entidade). Sendo assim, o sistema deve tanto ler, quanto escrever no banco de dados, que neste caso, são arquivos binários.

## **Fluxo do programa**

Antes de mostrar as telas de fato seu programa deve solicitar via entrada padrão o caminho para o banco de dados:

```
#####  
DIGITE O CAMINHO DO BANCO DE DADOS:  
#####
```

Este caminho indica o diretório em que o banco de dados pode estar armazenado (ou seja, os arquivos binários descritos na seção anterior). Se o binário existir, significa que já existe um banco de dados para a entidade. Se não, significa que o banco vai começar do zero e você será o responsável por criá-lo dentro do diretório informado. Um exemplo de caminho: /home/documentos/banco\_de\_dados

### Tela de acesso

Ao executar o programa a primeira tela “útil” deve solicitar login e senha do usuário através do seguinte menu:

```
##### ACESSO MINI-SADE #####  
DIGITE SEU LOGIN:  
DIGITE SUA SENHA:  
#####
```

O login e senha deve ser comparado com as credências do secretário ou médico já cadastrados no banco. Se o login falhar (ou seja, não existe o usuário ou a senha está incorreta), o programa deve exibir a mensagem `SENHA INCORRETA OU USUARIO INEXISTENTE` e repetir a tela de acesso. Se o login for bem-sucedido, o programa deve ir para o menu inicial que será descrito na sequência. Se este for o primeiro acesso ao sistema (ou seja, não existe nenhum usuário cadastrado), o sistema deve direcionar para a tela de cadastro de usuário que terá o nível de acesso `ADMIN`.

## Tela de menu principal

Após a tela de acesso, é exibido o menu principal do programa. Este menu oferece acesso às principais funcionalidades do software. Porém, ele depende do tipo de usuário e nível de acesso. O menu completo aparece apenas para secretários com nível de acesso ADMIN. Ele deve ser construído da seguinte forma:

```
##### MENU PRINCIPAL #####
```

ESCOLHA UMA OPCAO:

- (1) CADASTRAR SECRETARIO
- (2) CADASTRAR MEDICO
- (3) CADASTRAR PACIENTE
- (4) REALIZAR CONSULTA
- (5) BUSCAR PACIENTES
- (6) RELATORIO GERAL
- (7) FILA DE IMPRESSAO
- (8) FINALIZAR O PROGRAMA

```
#####
```

**Observação:** ADMINs são capazes de acessar qualquer parte do sistema. Caso ele acesse algum item que necessite de uma informação do médico (ex: solicitar biópsia), essas informações devem conter uma string vazia.

Para um secretário do nível USER, o menu deve ser construído da seguinte forma:

```
##### MENU PRINCIPAL #####
```

ESCOLHA UMA OPCAO:

- (2) CADASTRAR MEDICO
- (3) CADASTRAR PACIENTE
- (5) BUSCAR PACIENTES
- (6) RELATORIO GERAL
- (7) FILA DE IMPRESSAO
- (8) FINALIZAR O PROGRAMA

```
#####
```

Por fim, médicos devem ser capazes de visualizar o menu da seguinte forma:

```
##### MENU PRINCIPAL #####
```

ESCOLHA UMA OPCAO:

- (4) REALIZAR CONSULTA
- (5) BUSCAR PACIENTES
- (6) RELATORIO GERAL
- (7) FILA DE IMPRESSAO
- (8) FINALIZAR O PROGRAMA

```
#####
```

Cada opção do menu pode ser acessada utilizando o número que a identifica. Por exemplo, para acessar a opção INICIAR CONSULTA o usuário deve utilizar o caractere 4.

### Telas de cadastro (opções 1, 2 e 3)

As telas de cadastro de secretário, médico e paciente funcionam de maneira similar. Todas elas devem ler as entradas via entrada padrão (ou seja, utilizando `scanf` ou similar) e os dados devem ser lidos na sequência (ou seja, uma informação por linha). Os dados que são comuns aos 3 atores devem ser lidos primeiro. Sendo assim, a tela de cadastro de secretário deve ser realizada da seguinte forma:

```
##### CADASTRO SECRETARIO #####
```

```
NOME COMPLETO: <inserir dado via teclado>
```

```
CPF: <inserir dado via teclado>
```

```
DATA DE NASCIMENTO: <inserir dado via teclado>
```

```
TELEFONE: <inserir dado via teclado>
```

```
GENERO: <inserir dado via teclado>
```

```
NOME DE USUARIO: <inserir dado via teclado>
```

```
SENHA: <inserir dado via teclado>
```

```
NIVEL DE ACESSO: <inserir dado via teclado>
```

```
CADASTRO REALIZADO COM SUCESSO. PRESSIONE QUALQUER TECLA PARA  
VOLTAR PARA O MENU INICIAL
```

```
#####
```

As telas de cadastro de paciente e médico seguem o mesmo padrão, são elas:

```
##### CADASTRO PACIENTE #####
```

```
NOME COMPLETO: <inserir dado via teclado>
```

```
CPF: <inserir dado via teclado>
```

```
DATA DE NASCIMENTO: <inserir dado via teclado>
```

```
TELEFONE: <inserir dado via teclado>
```

```
GENERO: <inserir dado via teclado>
```

```
CADASTRO REALIZADO COM SUCESSO. PRESSIONE QUALQUER TECLA PARA  
VOLTAR PARA O MENU INICIAL
```

```
#####
```

```
##### CADASTRO MEDICO #####
```

```
NOME COMPLETO: <inserir dado via teclado>
```

```
CPF: <inserir dado via teclado>
```

```
DATA DE NASCIMENTO: <inserir dado via teclado>
```

TELEFONE: <inserir dado via teclado>  
GENERO: <inserir dado via teclado>  
CRM: <inserir dado via teclado>  
NOME DE USUARIO: <inserir dado via teclado>  
SENHA: <inserir dado via teclado>

CADASTRO REALIZADO COM SUCESSO. PRESSIONE QUALQUER TECLA PARA VOLTAR PARA O MENU INICIAL

#####

Em todos os casos, ao digitar o último valor, o programa deve registrar a informação no banco de dados e retornar para o menu principal após o usuário pressionar qualquer tecla.

### Tela de realizar consulta médica

A tela para realizar uma consulta inicia com a solicitação do CPF do paciente para verificar se o mesmo está cadastrado:

##### CONSULTA MEDICA #####

CPF DO PACIENTE: <inserir dado via teclado>

#####

Se o paciente não foi cadastrado, deve ser exibido na tela a mensagem PACIENTE SEM CADASTRO e esperar que o usuário aperte qualquer tecla para retornar para o menu principal:

##### CONSULTA MEDICA #####

CPF DO PACIENTE: 128.456.485-98

PACIENTE SEM CADASTRO

PRESSIONE QUALQUER TECLA PARA VOLTAR PARA O MENU INICIAL

#####

Se o paciente for encontrado, deve ser exibido o nome e a data de nascimento do paciente e a consulta poderá prosseguir. Neste caso, deve ser coletada todas as informações descritas anteriormente seguindo, inclusive, a mesma ordem:

##### CONSULTA MEDICA #####

CPF DO PACIENTE: 121.568.114-32

- - -

- NOME: LUCIAN

- DATA DE NASCIMENTO: 19/11/1990

- - -

DATA DA CONSULTA: <inserir dado via teclado>

POSSUI DIABETES: <inserir dado via teclado>

FUMANTE: <inserir dado via teclado>

ALEGIA A MEDICAMENTO: <inserir dado via teclado>

HISTORICO DE CANCER: <inserir dado via teclado>

TIPO DE PELE: <inserir dado via teclado>

#####

Ao inserir a última informação, o médico deve ser capaz de cadastrar uma ou mais lesões, gerar os documentos ou finalizar o atendimento. Isso é realizado com o seguinte menu:

##### CONSULTA MEDICA #####

ESCOLHA UMA OPCAO:

- (1) CADASTRAR LESAO
- (2) GERAR RECEITA MEDICA
- (3) SOLICITACAO DE BIOPSIA
- (4) ENCAMINHAMENTO
- (5) ENCERRAR CONSULTA

#####

Ao escolher a opção de cadastrar lesão, deve aparecer a seguinte tela solicitando as informações previamente descritas (lembre-se que uma consulta pode gerar nenhuma ou várias lesões):

##### CONSULTA MEDICA #####

CADASTRO DE LESAO:

DIAGNOSTICO CLINICO: <inserir dado via teclado>

REGIAO DO CORPO: <inserir dado via teclado>

TAMANHO: <inserir dado via teclado>

ENVIAR PARA CIRURGIA: <inserir dado via teclado>

ENVIAR PARA CRIOTERAPIA: <inserir dado via teclado>

LESAO REGISTRADA COM SUCESSO. PRESSIONE QUALQUER TECLA PARA  
RETORNAR AO MENU ANTERIOR

#####

Ao inserir a última informação, a lesão deve ser salva e o programa espera o usuário digitar qualquer tecla para voltar para o submenu anterior. Neste caso, o médico pode inserir uma nova lesão ou escolher alguma das outras opções. Se a opção escolhida for receita médica, o programa deve ser direcionado para a seguinte tela:



##### CONSULTA MEDICA #####

RECEITA MEDICA:

TIPO DE USO: <inserir dado via teclado>

NOME DO MEDICAMENTO: <inserir dado via teclado>

TIPO DE MEDICAMENTO: <inserir dado via teclado>

QUANTIDADE: <inserir dado via teclado>

INSTRUÇÕES DE USO: <inserir dado via teclado>

RECEITA ENVIADA PARA FILA DE IMPRESSAO. PRESSIONE QUALQUER TECLA PARA  
RETORNAR AO MENU ANTERIOR

#####

Ao inserir a última informação, a receita deve ser enviada para fila de impressão e o programa espera o usuário digitar qualquer tecla para voltar para o submenu anterior. Se o médico escolher a opção de solicitação de biopsia, nenhuma informação a mais é necessária (tudo já foi coletado). Neste caso, o programa deve exibir a mensagem a seguir e esperar o usuário apertar qualquer tecla para voltar para o submenu.

##### CONSULTA MEDICA #####

SOLICITACAO DE BIOPSIA ENVIADA PARA FILA DE IMPRESSAO. PRESSIONE  
QUALQUER TECLA PARA RETORNAR AO MENU ANTERIOR

#####

Porém, lembre-se que só é possível solicitar biópsia se o paciente possui ao menos uma lesão que foi encaminhada para cirurgia. Se o médico solicitar a biópsia e o paciente não cumpre o pré-requisito, o programa deve apresentar a seguinte mensagem:

##### CONSULTA MEDICA #####

NAO E POSSIVEL SOLICITAR BIOPSIA SEM LESAO CIRURGICA. PRESSIONE  
QUALQUER TECLA PARA RETORNAR AO MENU ANTERIOR

#####

Ao escolher o encaminhamento, o funcionamento é similar e ao inserir a última informação solicitada, o programa espera o usuário digitar algo para voltar para o menu inicial.

##### CONSULTA MEDICA #####

ENCAMINHAMENTO:

ESPECIALIDADE ENCAMINHADA: <inserir dado via teclado>

MOTIVO: <inserir dado via teclado>

ENCAMINHAMENTO ENVIADO PARA FILA DE IMPRESSAO. PRESSIONE QUALQUER  
TECLA PARA RETORNAR AO MENU ANTERIOR

#####

Por fim, se o médico escolher a opção de encerrar a consulta, o programa deve voltar para o menu principal.

### Tela de busca de paciente

Ao selecionar a opção de buscar pacientes o programa deve apresentar a seguinte tela:

```
##### BUSCAR PACIENTES #####
NOME DO PACIENTE: <inserir dado via teclado>
DATA DA CONSULTA:<inserir dado via teclado>
DIAGNOSTICO:<inserir dado via teclado>
#####
```

Ao inserir a última informação, se nenhum paciente for encontrado, o programa deve oferecer a seguinte tela:

```
##### BUSCAR PACIENTES #####
NENHUM PACIENTE FOI ENCONTRADO. PRESSIONE QUALQUER TECLA PARA
RETORNAR AO MENU ANTERIOR
#####
```

Porém, se o programa encontrar um ou mais pacientes, ele deve retornar a seguinte tela:

```
##### BUSCAR PACIENTES #####
PACIENTES ENCONTRADOS:
1 - LUCIAN (556.889.664-85)
2 - LUCIAN (121.568.114-32)
3 - LUCIAN (456.987.564-96)

SELECIONE UMA OPÇÃO:
  (1) ENVIAR LISTA PARA IMPRESSAO
  (2) RETORNAR AO MENU PRINCIPAL
#####
```

Se o médico escolher a opção de enviar a lista para impressão, deve ser apresentada a seguinte tela:

```
##### BUSCAR PACIENTES #####
LISTA ENVIADA PARA FILA DE IMPRESSAO. PRESSIONE QUALQUER TECLA PARA
RETORNAR AO MENU PRINCIPAL
#####
```

~~Se o médico escolher a opção de visualizar paciente, deve ser apresentado a seguinte tela:~~

~~##### BUSCAR PACIENTES #####  
DIGITE O NUMERO DO PACIENTE: <inserir dado via teclado>  
#####~~

~~Após digitar o número do paciente (e esse número está relacionado a lista que foi apresentada), deve ser apresentada uma tela com as informações do paciente e o médico possui 2 opções, enviar para impressão ou voltar para o menu principal:~~

~~##### BUSCAR PACIENTES #####  
DADOS DO PACIENTE:  
NOME: LUCIAN  
DATA DE NASCIMENTO: 19/11/1990 (32 ANOS)  
GENERO: MASCULINO  
TELEFONE: (27)99898-3131  
DIABETES: SIM  
FUMANTE: NAO  
ALERGIA A MEDICAMENTO: SIM  
HISTORICO DE CANCER: NAO  
TIPO DE PELE: III~~

~~LESOES:  
TOTAL: 3  
ENVIADA PARA CIRURGIA: 1  
ENVIADA PARA CRIOTERAPIA: 1~~

~~DESCRICAO DAS LESOES:  
L1- CARCINOMA ESPINOCELULAR- FACE- 30MM- ENVIADA PARA CIRURGIA  
L2- CERATOSE ACTINICA- ANTEBRACO- 90MM- ENVIADA PARA CRIOTERAPIA  
L3- CERATOSE SEBORREICA- FACE- 40MM~~

~~SELECIONE UMA OPÇÃO:  
(1)- ENVIAR PARA IMPRESSAO  
(2)- RETORNAR AO MENU PRINCIPAL~~

~~#####~~

~~Se o médico escolher enviar para impressão, deve ser apresentada a seguinte tela:~~

~~##### BUSCAR PACIENTES #####~~

DADOS ENVIADO PARA FILA DE IMPRESSAO. PRESSIONE QUALQUER TECLA PARA  
RETORNAR AO MENU PRINCIPAL

#####

Observe que em todos os casos, ao finalizar a opção, o programa deve voltar para o  
**menu principal.**

### Tela de relatório geral

Ao selecionar a opção de relatório geral deverá ser exibido na tela o relatório (como  
descrito na seção anterior) e o usuário deve ser capaz de enviar o mesmo para a  
impressão. Dessa forma, deve ser criado a seguinte tela:

##### RELATORIO GERAL #####

NUMERO TOTAL DE PACIENTES ATENDIDOS: 1

IDADE MEDIA: 43

DISTRIBUICAO POR GENERO:

- FEMININO: 1

- MASCULINO: 0

- OUTROS: 1

TAMANHO MEDIO DAS LESOES: 10

NUMERO TOTAL DE LESOES: 1

NUMERO TOTAL DE CIRURGIAS: 1

NUMERO TOTAL DE CRIOTERAPIA: 0

SELECIONE UMA OPÇÃO:

(1) ENVIAR PARA IMPRESSAO

(2) RETORNAR AO MENU PRINCIPAL

#####

Se o usuário escolher enviar para impressão, deve ser apresentada a seguinte tela:

##### RELATORIO GERAL #####

RELATÓRIO ENVIADO PARA FILA DE IMPRESSAO. PRESSIONE QUALQUER TECLA  
PARA RETORNAR AO MENU ANTERIOR

#####

### Tela de fila de impressão

Ao selecionar a opção de fila de impressão deve ser apresentada a seguinte tela:

##### FILA DE IMPRESSAO MEDICA #####

ESCOLHA UMA OPCAO:

- (1) EXECUTAR FILA DE IMPRESSAO
- (2) RETORNAR AO MENU ANTERIOR

#####

Se o usuário escolher executar a fila de impressão, todos os documentos que estiverem salvos no banco de dados (que neste caso é o arquivo fila\_impressao.bin) serão impressos em sequência. Para isso, tomando como exemplo que exista dois documentos na fila (*receita.txt* e *lista\_busca.txt*), a tela a ser criada deve ser a seguinte:

##### Fila de Impressao Medica #####

EXECUTANDO Fila de Impressão:

RECEITUARIO

NOME: LUCIAN

USO ORAL

DIPIRONA 500MG

12 COMPRIMIDO(S)

TOMAR 01 COMPRIMIDO E 6 EM 6 HORAS

DR. MUNDO (CRM-132456)

10/10/2023

1 - JARVAN IV

2 - TEEMO

PRESSIONE QUALQUER TECLA PARA VOLTAR PARA O MENU PRINCIPAL

#####

## Funcionamento do programa

Após compilado, seu programa deve receber um argumento de entrada através da linha de comando (para mais informações, [leia este tutorial](#)). Este argumento vai indicar o caminho para pasta em que seu programa deve escrever as impressões de documentos.

Exemplo de chamada do programa:

```
./trab2 /home/user/caso_1
```

A entrada de dados do programa será realizada utilizando a entrada padrão. Isso significa que a leitura pode ser realizada tanto via teclado quanto usando o terminal do Linux com o comando `<`. Exemplo de chamada do programa:

```
./trab2 /home/user/caso_1 < dados_entrada
```

Para facilitar a implementação, será fornecido em cada caso de teste o arquivo `dados_entrada.txt`, para simular entradas possíveis dentro do programa.

**Observação:** para o *script* funcionar corretamente é necessário realizar os seguintes passos:

1. Concatenar o caminho da pasta do caso recebido na linha de comando com o caminho do banco de dados lido na entrada padrão
2. Concatenar a string saída no final do caminho da pasta do caso

Exemplo: se o caminho recebido via linha de comando é **Resultados/apacheco/main/Casos/1** e o caminho lido do banco de dados é `../..` se programa deve usar os seguintes caminhos:

- Caminho final do banco de dados:  
**Resultados/apacheco/main/Casos/1/../../**
- Caminho final do caso de teste:  
**Resultados/apacheco/main/Casos/1/saida**

## Informações gerais

Alguns arquivos de entrada e respectivos arquivos de saída serão fornecidos para o aluno. O aluno deverá utilizar tais arquivos para testes durante a implementação do trabalho. É de responsabilidade do aluno criar novos arquivos para testar outras possibilidades do programa e garantir seu correto funcionamento. O trabalho será corrigido usando, além dos arquivos dados, outros arquivos (específicos para a correção e não disponibilizados para os alunos) seguindo a formatação descrita neste documento. Em caso de dúvida, entre em contato com os professores e/ou monitores. O uso de arquivos com formatação diferente poderá acarretar em incompatibilidade durante a correção do trabalho e consequentemente na impossibilidade de correção do mesmo (sendo atribuído a nota zero). Portanto, siga estritamente o formato estabelecido.

## Implementação e verificação de resultados

A implementação deverá seguir o template de código disponibilizado no repositório do trabalho dentro da organização da disciplina no Github:

- Link para o repositório: <https://github.com/prog-II-ufes/template-TP2-etapa-1>

A utilização do template segue os mesmos princípios dos exercícios disponibilizados ao longo da disciplina. Você deve utilizar os arquivos .h disponibilizados e, obrigatoriamente, implementar todas as funcionalidades requisitadas para construção destes TADs. Todavia, vocês são livres para implementar outros TADs e/ou bibliotecas que julgarem necessárias. Além disso, você já possui acesso ao script de correção automática de exercícios e trabalhos. Ele será utilizado para corrigir o seu trabalho. Desta forma, é extremamente recomendado que você avalie seu trabalho utilizando o script e os casos de teste disponibilizados.

Para fins de depuração, você também pode utilizar softwares de comparação de arquivos, como diff e o Meld (alternativa gráfica do diff). Diferenças na formatação poderão impossibilitar a comparação e consequentemente impossibilitar a correção do trabalho. O programa será considerado correto se gerar a saída esperada idêntica à fornecida com os casos de teste.

## Pontuação e Processo de Correção

**Pontuação:** a primeira etapa do trabalho será pontuada de acordo com sua implementação e seguindo os pontos apresentados na tabela a seguir. A pontuação da tabela será utilizada para calcular a nota para cada caso de teste. A pontuação final será a média de pontos considerando todos os casos. Haverá o mesmo número de casos visíveis e ocultos (que serão divulgados durante a segunda etapa do trabalho em sala).

É importante observar que os pontos descritos na tabela não são independentes entre si, isto é, alguns itens são pré-requisitos para obtenção da pontuação dos outros. Observe também que modularização e gerenciamento de memória são pré-requisitos do trabalho. Se você não os cumprir, será descontado uma porcentagem da pontuação obtida (ver próxima sub-seção). Outros erros gerais de funcionamento ou lógica serão tratados de maneira individual.

Item	Quesitos	Ponto
------	----------	-------

Receita médica	Gerar o documento de receita médica com sucesso	10%
Solicitação de biópsia	Gerar o documento de solicitação de biópsia com sucesso	10%
Encaminhamento	Gerar o documento de encaminhamento com sucesso	10%
Resultado de lista busca	Gerar o documento de resultado de lista de busca com sucesso	30%
<del>Visualização de paciente</del>	<del>Gerar o documento de visualização de paciente com sucesso</del>	<del>25%</del>
Relatório geral	Gerar o documento de relatório geral com sucesso	40%

**Descontos:** a pontuação obtida de acordo com a tabela anterior será utilizada para calcular a nota final do seu trabalho. O gerenciamento de memória e modularização são itens obrigatórios. Sendo assim, não basta apenas acertar a saída, é necessário seguir os templates de código disponibilizados e gerenciar corretamente o uso da memória. Logo, considere que seu trabalho seja capaz de gerar todos os arquivos corretamente e obtenha pontuação igual a 10. A partir desta pontuação a sua nota final será calculada da seguinte maneira:

(1) primeiramente será verificado se você seguiu corretamente o template de código fornecido. Na tabela a seguir são descritos todos templates disponibilizados e o peso de cada um deles:

Item	Descrição	Peso
tFila.h	TAD relacionado a fila	10%
tDocumento.h	TAD relacionado a documento	10%
tReceita.h	TAD relacionado a receita	5%
main.c + restante	Função principal do programa + restante do código necessário	75%

Se você implementou todos os templates corretamente não haverá nenhum desconto na sua pontuação nesta fase. Porém, se você deixar de implementar algum dos templates, sua pontuação será descontada de acordo com o peso do template.



(2) após a verificação dos templates, será verificado o gerenciamento de memória através do Valgrind. Caso o seu programa apresente vazamento de memória e/ou erro de memória será **descontado 40%** da pontuação que chegar até essa fase.

Sendo assim, é de **suma importância que seu trabalho siga a risca os templates e gerencie corretamente a memória** (lembre-se, esses são os principais objetivos desta disciplina). Todo esse processo de pontuação será realizado por meio do script de correção que vocês terão acesso. Observe que a compilação da pasta completo (dentro de resultados) no script não é utilizado para computar a pontuação. Porém, pode ser utilizado para depuração de erros no seu programa.

**Processo de correção do trabalho:** como consta no plano de ensino da disciplina, os trabalhos práticos são divididos em duas etapas. A primeira etapa é o desenvolvimento do projeto seguindo as regras e descrições apresentadas neste documento. A segunda etapa consiste na correção, atualização e/ou incremento de funcionalidades implementadas na etapa anterior. Esta última etapa é realizada em sala de aula de acordo com cronograma da disciplina. A nota final do trabalho é a média entre as duas etapas, ou seja, cada etapa vale 50% da nota.

É importante ressaltar que ao iniciar a etapa 2 vocês terão acesso à nota e aos casos de teste ocultos da etapa 1. Portanto, vocês poderão corrigir possíveis erros cometidos durante o desenvolvimento da etapa 1.

## Prazo e submissão

Todas as informações referentes a prazo de entrega e forma de submissão estão disponíveis na atividade **Trabalho Prático 2 - Etapa 1** no classroom da disciplina.

## Regras gerais

Para todas as etapas do trabalho, considere as seguintes regras:

- Trabalhos entregues após o prazo **não** serão corrigidos (recebendo a nota zero)
- O trabalho deverá ser feito **individualmente** e pelo próprio aluno, isto é, o aluno deverá necessariamente conhecer e dominar **todos** os trechos de código criados. Cada aluno deverá trabalhar independente dos outros, não sendo permitido a cópia ou compartilhamento de código.
  - Haverá verificação automatizada de plágio. Trabalhos identificados como iguais, em termos de programação (por exemplo, mudar nomes de variáveis e funções entre outros não faz dois trabalhos serem diferentes), **serão penalizados com a nota zero e poderão ser**

**submetidos para penalidades adicionais em instâncias superiores.**

Isso também inclui a pessoa que forneceu o trabalho, sendo portanto, de sua obrigação a proteção de seu trabalho contra cópias ilícitas. Proteja seu trabalho e não esqueça cópias do seu código nas máquinas de uso comum.

- Nenhuma entrada dados nos casos de teste vai possuir acentos ou caracteres especiais.
- Em todos os arquivos de saída você deve escrever sempre com letras maiúsculas. Por exemplo, se o paciente se chama Joao, na saída dele deve constar como JOAO.
- Todos os arquivos de entrada vão seguir sempre o mesmo padrão, incluindo o nome dos arquivos.
- Todas as entradas de dados serão fornecidas de acordo com a descrição. Por exemplo, se o padrão do CPF é 000.000.000-00, todos CPFs fornecidos vão estar neste padrão.
- Para dados de ponto flutuante, use sempre precisão simples. Para a escrita destes dados, use apenas duas casas decimais.
- Para calcular a idade dos pacientes considere a data de referência como sendo 09/11/2023 (dia em que o trabalho foi divulgado)
- Para dados multidimensionais no qual o tamanho é desconhecido, é obrigatório o uso de alocação dinâmica de memória. Liberar a memória alocada é de sua responsabilidade.
  - O script de correção utiliza o Valgrind. Haverá descontos significativos da nota para erros e vazamentos de memória apontados pela ferramenta
- Modularização e organização são fundamentais na construção do código. Você é obrigado a implementar os TADs solicitados via o template dos .h. Todavia, isso não impede a criação de outros.
- Todos os trabalhos serão corrigidos utilizando o sistema operacional Linux. **Garanta que seu código funcione nos computadores do Labgrad.** Qualquer discrepância de resultado na correção por conta de Sistema Operacional, os computadores do Labgrad serão utilizados para solucionar o problema.
- **Todos os TADs implementados no trabalho devem ser opacos**
- Possíveis problemas na descrição do trabalho serão solucionados e comunicados via Classroom. Don't Panic!

## **Considerações finais**

Esse documento descreve de maneira geral as regras de implementação do trabalho. É de responsabilidade do aluno garantir que o programa funcione de maneira correta e amigável com o usuário. Qualquer alteração nas regras do trabalho será comunicada em sala e no portal do aluno. É responsabilidade do aluno frequentar as aulas e se manter atualizado em relação às especificações do trabalho. Caso seja notada qualquer tipo de inconsistência nos arquivos de testes disponibilizados, comunique imediatamente ao professor para que ela seja corrigida e reenviada para os alunos.