

Orientação do PIM IV

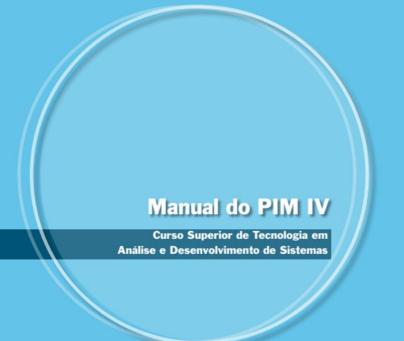
Profa. Dra. Vanessa Lessa

Estrutura

Observações importantes para seu PIM

- A estrutura do PIM está totalmente descrita no manual.
- Deve-se seguir rigorosamente o que está no manual quanto às normas de formatação para que possa ser aceito.
- 3) Procure no calendário do aluno as **datas de postagem** e preste atenção para não perder essa data; caso contrário, não poderá ser atribuída nota.
 - 4) O PIM deve ser inédito.
 - *A capa do manual pode ser diferente.





- A pandemia de 2020 estimulou uma demanda por inovação médica e muitas empresas começaram a investir no desenvolvimento de softwares de saúde.
- Países ao redor do mundo exigem tecnologia e equipamentos para combater o vírus: testes, máscaras respiratórias, equipamentos de proteção, ventiladores, desfibriladores e muito mais. À medida que aprendemos mais sobre a doença, vemos as tecnologias digitais de saúde cada vez mais sendo adotadas nesse contexto.



Fonte: https://pixabay.com/photos/coronavirus-virus-mask-corona-4914026/

 Como o novo coronavírus infecta um número exponencial de pessoas, os métodos convencionais de rastreamento para identificar aqueles que foram diagnosticados com o vírus e limitar a transmissão não são suficientes.

É por isso que governos em todo o mundo têm recorrido ao uso de tecnologia para esse fim.
 Ao rastrear os usuários, as autoridades são capazes de identificar os indivíduos que foram contaminados e, posteriormente, alertar àqueles que podem estar próximos a alguém

com Covid-19.

Fonte: https://pixabay.com/illustrat ions/corona-coronavirus-virus-covid-19-5401250/

O objetivo será desenvolver um sistema em C (utilizando o Code::Blocks – disponível de forma gratuita no seguinte link: http://www.codeblocks.org/) que será utilizado pelos hospitais para cadastrar os pacientes que forem diagnosticados com Covid-19 e carecem de acompanhamento e monitoramento.

🎳 wxauitest.cpp [wxauitest] - Code::Blocks v1.0 View Search Project Build Debug wxSmith Tools Plugins Settings Help 🐧 👔 🔍 💲 🔊 🔊 🐼 Build target: Release Unicode Management Projects Nr Address File Line wxauitest.cpp ☐ Workspace MyFrame::OnCopyPerspectiveCod... wxEvtHandler::ProcessEventIfH... C:\WINDOWS\sy... wxauitest wxEventHashTable::HandleEvent() C:\WINDOWS\sy... wxauitest.cop wxEvtHandler::ProcessEvent() C:\WINDOWS\sy... wxauitest.rc wxEvtHandler::ProcessEvent() C:\WINDOWS\sy... wxFrameBase::ProcessCommand() C:\WINDOWS\sy... myTrama . . Handlafommand() C-\MINDOMS\ ev Save to text file Refresh d b X wxauitest.cpp wxauitest.rc m action window = 0xbaadf(A 849 -m action hintrect 850 m_perspectives_menu->Append(ID_FirstPerspective + m_perspect 851 x = 0m perspectives.Add(m mgr.SavePerspective()); 852 y = 0853 width = 0Open files list 854 void MyFrame::OnCopyPerspectiveCode(wxCommandEvent& event) height = 0 855 Opened Files $m_bover_button = 0x0$ 856 wxString s * m_mgr.SavePerspective(); wxauitest.cpp m_last_hint 857 wxauitest.rc x = 0858 if (wxTheClipboard->Open()) y = 0859 860 wxTheClipboard->SetData(new wxTextDataObject(s)); width = 0CPU Registers 861 wxTheClipboard->Close(); height = 0 862 Integer m last mouse move 863 0x22f8ac 2291884 x = 259864 0xc39940 12818752 y = 15865 void MyFrame::OnRestorePerspective(vxCommandEvent& event) 0x1 m_i hint_wnd = 0x0866 0x1 m hint fadetimer = <incompl 867 m mgr. LoadPerspective(m perspectives. Item(event. GetId()) 0x22f84c 2291788 m_hint_fadeamt = -1163005 4457104 static sm_eventTableEntries 0x22f8d4 2291924 0x22f9dc 2292188 0x4123da 4269018 Compiling: done Breakpoints 0x212 Adding source dir: C:\Devel\codeblocks\wxaui-0.9\sample\ 0x1b Adding file: .\ReleaseUnicode\wxauitest.exe aui-0.9\sample\wxauitest.cpp 856 0x23 Starting debugger: done 0x23 Debugger name and version: GNU gdb 6.3 0x3b At C:/Devel/codeblocks/wxaui-0.9/sample/wxauitest.cpp:856 0x0 At C:/Devel/codeblocks/wxaui-0.9/sample/wxauitest.cpp:858 0x23 At C:/Devel/codeblocks/wxaui-0.9/sample/wxauitest.cpp:858 🚺 Code::Blocks Debug 🔍 Search results 📞 Build log 🥐 Build messages 📞 Debugger C:\Devel\codeblocks\wxaui-0.9\sample\wxauitest.cpp Line 858, Column 1 Insert Read/Write

Fonte: https://www.cod eblocks.org/

 Ao receber o diagnóstico positivo, os profissionais da saúde devem realizar o login no sistema (informando o usuário e a senha) e informar os dados pessoais do paciente que serão salvos em um arquivo.

- Nome.
- CPF.
- Telefone.
- Endereço (rua, número, bairro, cidade, estado e CEP).
- Data de nascimento.
- E-mail.
- Data do diagnóstico.
- Comorbidade do paciente (diabetes, obesidade, hipertensão, tuberculose, outros).

O que é arquivo?

- As matrizes usam um índice de controle, enquanto os arquivos usam um ponteiro de registro.
- A principal vantagem de um arquivo é que as informações armazenadas podem ser consultadas a qualquer momento. Outra vantagem é o fato de armazenar um número maior de registros do que uma tabela em memória. Está limitado apenas ao tamanho do meio físico para gravação.
 - É um conjunto de registros (que pode ser apenas um registro) que, por sua vez, é um conjunto de campos (que pode ser apenas um campo), sendo cada campo o conjunto de informações.

Fonte: Manzano (2013, p. 164).

O que é arquivo?

Um arquivo em linguagem C é do tipo FILE, uma estrutura formada por elementos do mesmo tipo dispostos de forma sequencial. Seu objetivo é fazer a comunicação entre a memória principal (RAM) e a secundária (meios magnéticos) por meio do programa e do sistema operacional. Esse tipo deve ser definido com a seguinte sintaxe:

- FILE <*variável ponteiro>
- em que:
- <*variável ponteiro> definição e um ponteiro para a estrutura do tipo FILE.
 - Para usar um arquivo (ler ou escrever), é necessário executar duas operações básicas, sendo abertura e fechamento, conseguidas com as instruções fopen() e fclose(), desde que o arquivo exista.

Fonte: Manzano (2013, p. 165).

O que é arquivo?

Tipo de abertura	Descrição
R	Este código permite apenas abrir um arquivo-texto para leitura de seus dados. É necessário que o arquivo esteja presente no disco.
W	Este código permite apenas abrir um arquivo-texto para escrita (gravação). Este código cria o arquivo para ser trabalhado. Caso o arquivo exista, este código recria o arquivo, ou seja, você perde o arquivo criado anteriormente. Deve ser usado com muito cuidado.
A	Este código permite apenas abrir um arquivo-texto para escrita (gravação), permitindo acrescentar novos dados ao final dele. Caso o arquivo não exista, ele será então criado.

Fonte: adaptado de: Manzano (2013, p. 166).

 Após o cadastro, o sistema deverá calcular a idade e verificar se o paciente possui alguma comorbidade e se pertence ao grupo de risco (maior de 65 anos).

 Caso o paciente pertença ao grupo de risco, o sistema deverá salvar em um arquivo de texto o CEP e a idade do paciente para que essa informação possa ser enviada

para a central da Secretaria de Saúde da cidade.

Fonte:
https://www.istockphoto.com/
br/foto/arquivos-e-pastas-derede-gm923752568253553773?utm_source=pix
abay&utm_medium=affiliate&
utm_campaign=SRP_photo_
noresults&referrer_url=https
%3A%2F%2Fpixabay.com%
2Fphotos%2Fsearch%2Farq
uivo%2520texto%2F&utm_te
rm=arquivo+texto



Como salvar um arquivo *.txt utilizando a linguagem C

- O arquivo-texto possibilita a criação de registros armazenados com tamanhos diferentes (o que não ocorre com os outros tipos de arquivo).
- Os arquivos-textos estão capacitados a armazenar caracteres, palavras, frases e também dados numéricos. Os números, entretanto, são armazenados como caracteres alfanuméricos, portanto, ocupam muito mais espaço em disco do que na memória de um computador. A solução para esse detalhe é utilizar funções que usem os números em formato binário, o que será visto mais adiante.

Fonte: Manzano (2013, p. 195).

Como salvar um arquivo *.txt utilizando a linguagem C

Antes de iniciar qualquer operação com arquivo, é necessário criá-lo.

```
/* Criacao de arquivo texto */
#include <stdio.h>
int main(void) {
/* definicao do ponteiro para o arquivo */
FILE *Ponteiro;
Ponteiro = fopen("arquivo.txt", "a");
fclose(Ponteiro);
return 0;
```

Fonte: Manzano (2013, p. 194-195).

O que deve ser entregue?

Ao final do desenvolvimento, a equipe deverá entregar um arquivo ZIP com o seguinte conteúdo:

- 1. Documentação do PIM (arquivo em Word no formato padrão da **ABNT**, com as telas do programa, manual do usuário, manual de instalação, como compilar o programa, como testar etc.);
- 2. Binários do programa (exe, dlls, arquivos de teste, arquivo de configuração);
- 3. Arquivos-fontes (desenvolvido no Code::Blocks).

Referências

 MANZANO, José Augusto N. G. Estudo dirigido de linguagem C. 17. ed. São Paulo: Editora Érica, 2013.

ATÉ A PRÓXIMA!