

**UNIVERSIDADE PAULISTA – UNIP EaD**  
**Projeto Integrado Multidisciplinar**  
**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

**HUGO DE FREITAS BARBOZA RA 2097641**  
**MARCELO APARHYAN DE JESUS RA 0439966**  
**TIAGO DA SILVA FERREIRA NEVES RA 0443889**

**PROJETO INTEGRADO MULTIDISCIPLINAR IV**  
**Sistema em c para cadastrar pacientes diagnosticados com covid-19.**

**RIBEIRÃO PRETO**  
**2021**

**HUGO DE FREITAS BARBOZA RA 2097641**  
**MARCELO APARHYAN DE JESUS RA 0439966**  
**TIAGO DA SILVA FERREIRA NEVES RA 0443889**

**PROJETO INTEGRADO MULTIDISCIPLINAR IV**  
**Sistema em C para cadastrar pacientes diagnosticados com covid-19.**

**Projeto Integrado Multidisciplinar  
para obtenção do título de tecnólogo  
em Curso Superior de Tecnologia  
em Análise e Desenvolvimento de  
Sistemas, apresentado à  
Universidade Paulista – UNIP EaD.**

**Orientador(a): Prof. Marcelo Santos.**

**Ribeirão Preto**  
**2021**

## RESUMO

Deste o final de 2019, o mundo entrou e um alerta devido a transmissão de um novo coronavírus, que ao passar de um animal para uma pessoa e depois de pessoa para pessoas, causando sérios problemas respiratórios. Esse novo coronavírus recebeu o nome de Covid-19. Diante do aumento dos casos de covid-19, os governos de todo o mundo tiveram que tomar providencias para melhorar os meios e as condições para identificar as pessoas que testaram positivos para este vírus. Como um força de conter a propagação do vírus, isolando-as, até que se recuperem, do convívio dos demais. O objetivo deste trabalho é o de desenvolver um projeto em linguagem C, que será utilizado nos hospitais e clinicas para a realização de cadastros de pacientes que tiveram o diagnóstico positivo para a covid-19. As informações cadastradas irão para um banco de dado, que irão para a central da Secretaria de Saúde da cidade. As disciplinas que contribuíram diretamente para o desenvolvimento deste trabalho foram Linguagem e Técnicas de Programação e Engenharia de Software I.

**Palavras-chave:** Covid-19, identificação, notificação, linguagem c.

## **ABSTRACT**

Since the end of 2019, the world has entered an alert due to the transmission of a new coronavirus, which, when passing from an animal to a person and then from person to person, causes serious respiratory problems. This new coronavirus was named Covid-19. With the increase in covid-19 cases, governments around the world have had to take steps to improve the means and conditions to identify people who have tested positive for this virus. As a force to contain the spread of the virus, isolating them, until they recover, from the coexistence of others. The objective of this work is to develop a project in C language, which will be used in hospitals and clinics to carry out records of patients who had a positive diagnosis for covid-19. The registered information will go to a database, which will go to the central office of the city's Health Department. The subjects that contributed directly to the development of this work were Programming Language and Techniques and Software Engineering I.

**Keywords:** Covid-19, identification, notification, language c.

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>5</b>
<b>2. OBJETIVOS.....</b>	<b>6</b>
<b>3. MATERIAIS E MÉTODOS .....</b>	<b>7</b>
<b>4. DESENVOLVIMENTO .....</b>	<b>8</b>
<b>4.1 Tela de acesso ao programa .....</b>	<b>8</b>
<b>4.2 Tela de cadastramento de pacientes .....</b>	<b>8</b>
<b>4.3 Manual de instalação .....</b>	<b>11</b>
<b>4.3.1 Manual do usuários.....</b>	<b>11</b>
<b>4.4 Arquivo fonte e compilação do programa .....</b>	<b>12</b>
<b>4.4.1 Arquivo fonte.....</b>	<b>12</b>
<b>4.5 Teste do programa .....</b>	<b>16</b>
<b>5. CONCLUSÃO .....</b>	<b>17</b>
<b>6. REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>18</b>

## 1. INTRODUÇÃO

Devido à pandemia da Covid-19, houve uma crescente demanda por inovações tecnológicas na área da saúde. Com isso, muitas empresas passaram a investir no desenvolvimento de *softwares* de saúde.

Conforme vão sendo adquiridos melhores conhecimentos sobre as características deste vírus, as tecnologias digitais de saúde vão sendo aprimoradas para melhor preveni-lo e combatê-lo.

Tendo em vista que o novo coronavírus contagia de uma à milhares de pessoas em pouco tempo, os métodos tradicionais de rastreamento para identificar as pessoas que foram acometidas pelo vírus e reduzir as possibilidades de transmissão não são suficientes. Diante disso, os governos de todos os países do mundo passaram a fazer uso de novas tecnologias para alcançar esse objetivo. Realizando o rastreamento adequado destas pessoas, aumentam as possibilidades de identificar pessoas infectadas e informar as pessoas que estão próximas a ela.

Com base no conteúdo das disciplinas Linguagem e Técnicas de Programação e Engenharia de Software I, através do uso de um sistema em C, por meio do codeblocks, será desenvolvido um programa, de uso pelos hospitais, que irá fazer, de maneira gratuita, o cadastramento, o acompanhamento e o monitoramento das pessoas que foram diagnosticadas com covid-19.

Após o cadastro, o sistema deverá calcular a idade e verificar se o paciente possui alguma comorbidade e se pertence ao grupo de risco (maiores 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 da Saúde da cidade.

## 2. OBJETIVOS

Este trabalho tem o objetivo geral de apresentar um sistema em C que será utilizado pelos hospitais para cadastrar os pacientes que forem diagnosticados com covid-19 e carecem de um acompanhamento e monitoramento para que essa informação possa ser enviada para a central da Secretaria da Saúde.

Os objetivos específicos a serem observados são:

- a) desenvolver e aplicar os conhecimentos adquiridos nas disciplinas de Linguagem e técnicas de programação e engenharia de software I;
- b) fomentar o hábito de executar projetos envolvendo múltiplas disciplinas;
- c) desenvolver a capacidade de identificar necessidades e propor soluções técnicas;
- d) elencar, argumentar e justificar sobre metodologias referentes ao desenvolvimento de um sistema para auxiliar a área da saúde;
- e) aplicar as normas ABNT para a produção de trabalhos acadêmicos.

### **3. MATERIAIS e MÉTODOS**

Com base nas disciplinas Linguagem e Técnicas de Programação e Engenharia de Software I, para o desenvolvimento de um programa em linguagem C, utilizou-se um modelo incremental, em virtude de um tempo curto para o seu desenvolvimento.

O método de desenvolvimento baseou-se no levantamento das necessidades, análise das possibilidades, esquematização do projeto, desenvolvimento, implementação e manutenção do programa.



## 4. DESENVOLVIMENTO

### 4.1 Tela de acesso ao programa

O programa desenvolvido neste trabalho possui uma tela inicial, conforme as orientações disponibilizadas para o desenvolvimento deste Projeto Integrado Multidisciplinar IV, conforme pode ser observado na Figura 1.

Figura 1 – Tela de Login.



Fonte: Elaborado pelos alunos, 2021.

Nesta primeira tela, os profissionais de saúde deverão realizar o cadastro no sistema dos pacientes que foram diagnosticados com covid-19. No entanto, antes disso é necessário realizar o login.

### 4.2 Tela de cadastramento de pacientes

Na figura 2, consta o Menu do Sistema, que é onde o usuário as opções que melhor atenderão às necessidades do momento do acesso.

Figura 2 – Menu do Sistema.



Fonte: Elaborado pelos alunos, 2021.

Na Figura 3, encontra-se a tela relacionado ao cadastro de um no paciente. Conforme pode ser observado, o cadastro do paciente tem início informando a idade do paciente, para em seguir abrir os demais campos cadastrais para que todas as características relevantes do paciente sejam incluídas no banco de dados do sistema.

Figura 3 – Tela de cadastro de um novo paciente.



Fonte: Elaborado pelos alunos, 2021.

Na figura 4, mostra os campos relacionados com as características única de cada paciente.

Figura 4 – Campos de cadastro do paciente.

```

----- CADASTRAR NOVO PACIENTE! -----
Nome: Teste
CPF: 000.000.000-00
Telefone: (11)99999-9999
Email: teste@teste.com

Data de nascimento
Ex:< DD.MM.AAAA >: 01.01.1001

Data do Diagnostico
Ex:< DD.MM.AAAA >: 02.02.2002

Endereco
Rua: Teste
Numero: 0
Bairro: Teste III
Cidade: Testolandia
Estado: Teste
CEP: 00000-000

Comorbidades <S> ou <N>: S
Informe a Idade do paciente: 1020

! Paciente pertence ao Grupo de Risco !
! O caso sera enviado pra Secretaria de Saude!

Gerar codigo do paciente: 0001

```

Fonte: Elaborado pelos alunos.

Nota-se que na figura 5, encontra-se as informações cadastrais mais relevantes do paciente, que são: nome, cpf, telefone, email dados residenciais e se possui alguma comorbidade que o classifique como pertencendo ao Grupo de Risco.

Na figura 5 mostra o tela, após a conclusão dos dados cadastrais do paciente, dando ao usuário a opção de incluir um novo cliente ou de voltar ao menu inicial que contém outras opções.

Figura 5 – Tela de registro concluído.

```

=====
! Registro Concluido com Sucesso! !
=====

-----SISCOV19-----
1-Novo cadastro
2-Voltar ao Menu

Entre com a Opção: _

```

Fonte: Elaborado pelos alunos.

Após o cadastro dos pacientes, caso o usuário deseje lista-los, o programa permite que isso seja realizado, conforme aparecer na figura 6:

Figura 6 – Tela Lista de pacientes.



Fonte: Elaborado pelos alunos.

### 4.3 Manual de instalação

Manual de instalação.

1. Abra o arquivo ZIP "siscov19.zip";
2. Para executar o programa, abra a pasta "SISCOV19";
3. Dentro dela encontre a pasta "bin";
4. Abra a pasta "Debug";
5. E clique 2 vezes sobre o arquivo EXE "SISCOV19.exe";
6. O programa será aberto no CMD do computador;
7. Quando o programa abrir siga o "Manual de Usuário".

#### 4.3.1 Manual do usuário

Manual do Usuário

1. Com o programa aberto no CMD insira os dados de "login" (admin) e "senha" (123456);

2. Após efetuar o "login com sucesso" escolha a opção que deseja no "Menu".
3. Caso escolha "Adicionar paciente" deve pressionar a opção 1 e preencher do formulário "Novo Cadastro";
4. Para "Alterar dados do paciente deve pressionar opção 2 iinformar código do paciente alterar os dados solicitados na tela;
5. Para ter acesso a "Lista de Pacientes" pressione a opção 3 e os pacientes cadastrados apareceram na tela, de acordo com sua urgência;
6. Para "Excluir Pacientes" pressione a opção 4 e informe o paciente a ser excluído do sistema;
7. Para "Encerrar o Programa" pressione a opção 0 seguido de Enter, dessa forma ele encerra o programa e fechar a janela do CMD.

## **4.4 Arquivo fonte e compilação do programa**

### **4.4.1 Arquivo fonte**

Como encontrar o "Arquivo Fonte"?

1. Abra a pasta "SISCOV19";
2. Encontre o arquivo "main.c";
3. Para abrir o "código fonte" clique 2 vezes no arquivo;
4. O IDE irá abrir arquivo e mostrar o "código fonte" na tela.

### **4.4.2 Compilação do programa**

Como Compilar e Testar o Programa?

1. Com o "código fonte" aberto no IDE Code:Blocks;
2. Acesse a "aba de ferramentas" Build > Cumpile current file ou pressione CRTLT+SHIFT+F9;
3. Para testar acesse a "aba de ferramentas" Build > Build and run ou pressione F9.

Consta na figura 7, as telas dos códigos fontes do programa.





```

93 fclose(sint);
94 ~
95 adicionar(&Arv, aux);
96 ~
97 aux->prox = tmp->head;
98 tmp->head = aux;
99 tmp->qnt++;
100 printf("1-Novo cadastro\n2-Voltar ao Menu\n");
101 printf("Opção: ");
102 scanf("%d", &op_cadastro_paciente);
103 } while (op_cadastro_paciente != 2);
104 }
105 ~
106 //Faz a listagem dos pacientes
107 void listar_pacientes(listas *tmp){
108 int op = 0;
109 printf("\t-----PACIENTES\n");
110 paciente *aux;
111 printf("\n\n");
112 printf("\t-----\n");
113 for(aux = tmp->head; aux != NULL; aux = aux->prox){
114 printf("Nome: %s\nCPF: %s\nSexo: %s\nIdade: %d\nTelefone: %d\nComorbidades: %s\nCodigo: %d\n", aux->nome, aux->cpf, aux->sexo, aux->idade, aux->telefone, aux->comorbidades, aux->codi);
115 printf("\t-----\n\n\n");
116 }
117 ~
118 ~
119 printf("\t-----\n\n\n");

```

```

120 printf("\t\t\tExistem %d pacientes cadastrados\n", tmp->qnt);
121 free(aux);
122 printf("\n\n");
123 printf("1-Voltar ao menu\n2-Encerrar o programa\n");
124 scanf("%d", &op);
125 if (op == 2){
126 exit(0);
127 }
128 }
129 void adicionar(arvore **Arv, paciente *pac){
130 arvore *temp = NULL;
131 if (*Arv == NULL){
132 temp = (arvore*)malloc(sizeof(arvore));
133 temp->Esq = NULL;
134 temp->Dir = NULL;
135 temp->No = pac;
136 *Arv = temp;
137 }
138 else {
139 if (pac->urg < ((*Arv)->No->urg)){
140 adicionar(&(*Arv)->Esq, pac);
141 }
142 else {
143 adicionar(&(*Arv)->Dir, pac);
144 }
145 }
146 }
147 //Em Ordem (Do menor pro maior)
148 void Mostrar(arvore *Arv){
149 if (Arv != NULL){
150 Mostrar(Arv->Esq);
151 printf("Nome: %s\nCPF: %s\nSexo: %s\nIdade: %d\nTelefone: %d\nComorbidades: %s\nUrgencia: %d\nCodigo: %d\n", Arv->No->nome, Arv->No->cpf, Arv->No->sexo, Arv->No->idade, Arv->No->telefone, Arv->No->comorbidades, Arv->No->urg, Arv->No->codi);

```

```

Arv->No->codi);
152 Mostrar(Arv->Dir);
153 }
154 }
155 //Alterando os dados dos pacientes
156 void alterar_paciente(listas *lista_escolhida){
157 char nome_substituto[50];
158 char cpf_substituto[50];
159 int codigo, idade_substituto, telefone_substituto;
160 paciente* atual = lista_escolhida->head;
161 system("cls");
162 ~
163 printf("\tCodigo do paciente a ser alterado: ");
164 scanf("%d", &codigo);
165 //Verificando se o código é válido
166 while(atual != NULL && atual->codi != codigo){
167 atual = atual->prox;
168 }
169 ~
170 ~
171 if(atual != NULL){
172 printf("\n\tNovo nome: ");
173 scanf("%s", nome_substituto);
174 strcpy(atual->nome, nome_substituto); //Faz a mudança de nome
175 ~
176 printf("\n\tNova idade: ");
177 scanf("%d", &idade_substituto);
178 atual->idade = idade_substituto; //Altera a idade
179 ~
180 printf("\n\tNovo CPF: ");
181 scanf("%s", cpf_substituto);
182 strcpy(atual->cpf, cpf_substituto); //Altera o CPF
183 printf("\n\tNovo telefone: ");
184 scanf("%d", &telefone_substituto);

```

```

185 atual->telefone = telefone_substituto
    ;//Altera o Telefone.
186 ~
187 ~
188 printf("\n\tDados alterados com sucesso
    .\n\tVoltando ao menu...");
189 }else{
190 printf("\n\tpaciente não
    encontrado\n\tVoltando ao menu...");
191 }
192 Sleep(3000);
193 ~
194 }
195 ~
196 listas* excluir_paciente(listas*
    lista_escolhida){
197 //Ponteiro para saber o elemento anterior
    ao elemento atual da lista.
198 paciente *anterior = NULL;
199 //Ponteiro para percorrer a lista sem
    perder o primeiro elemento da lista.
200 paciente *atual = lista_escolhida->head;
201 int codigo;
202 system("cls");
203 //Requisitando e lendo o código do paciente
    a ser excluído.
204 printf("\n\nCódigo do paciente a ser
    excluído: ");
205 scanf("%d", &codigo);
206 //Procurando o paciente na lista.
207 while(atual != NULL && atual->codi !=
    codigo){
208 anterior = atual;
209 atual = atual->prox;
210 }
211 ~
212 //Mensagem caso o paciente não seja
    encontrado.
213 if(atual == NULL){
214 printf("\n Paciente não encontrado.");
215 Sleep(1500);

```

```

214 printf("\n Paciente não encontrado.");
215 Sleep(1500);
216 return lista_escolhida;
217 }
218 //Excluindo o primeiro paciente da lista.
219 if(anterior == NULL){
220 printf("\n Paciente excluído com sucesso."
    );
221 lista_escolhida->head = atual->prox;
222 Sleep(1500);
223 return lista_escolhida;
224 //Excluindo um paciente do meio da lista.
225 }else{
226 printf("\n Paciente excluído com sucesso."
    );
227 anterior->prox = atual->prox;
228 Sleep(1500);
229 return lista_escolhida;
230 }
231 //Faz a animação do "Entrando..."
232 void entrando(){
233 int carregando;
234 for(carregando=1;carregando<=3;carregando
    ++){
235 printf("\n\n\t\t\tCarregando.");
236 Sleep(100);
237 system("cls");
238 printf("\n\n\t\t\tCarregando..");
239 Sleep(100);
240 system("cls");
241 printf("\n\n\t\t\tCarregando....");
242 Sleep(100);
243 system("cls");
244 }
245 }
246 //Menu principal
247 void menu(){
248 ~
249 int op_menu;
250 listas *lista_paciente = inicializarLista
    ();

```

```

249 ~
250 void menu(){
251 ~
252 ~
253 do{
254 printf("\t\t-----HOSPITAL
    UNIP COVID
    -----\\n\\n");
255 printf("\\n\\n");
256 printf("\tEscolha a opcao desejada\\n");
257 printf("\t[1] - Adicionar paciente\\n");
258 printf("\t[2] - Alterar dados do
    paciente\\n");
259 printf("\t[3] - Exibir pacientes [Por
    chegada]\\n");
260 printf("\t[4] - Exibir pacientes [Por
    urgencia]\\n");
261 printf("\t[5] - Excluir pacientes\\n");
262 printf("\t[0] - Encerrar o programa\\n");
263 printf("\n\tOpcao: ");
264 scanf("%d", &op_menu);
265 switch(op_menu){
266 case 1:-
267 system("cls");
268 cadastrar_paciente(lista_paciente, Arv);
269 break;
270 case 2:-
271 system("cls");
272 alterar_paciente(lista_paciente);
273 break;
274 case 3:-
275 system("cls");
276 listar_pacientes(lista_paciente);
277 break;
278 case 4:-
279 system("cls");
280 printf("-- Pacientes por ordem de urgencia

```



```

281 Mostrar(Arv);
282 system("pause>null");
283 break;
284 case 5:
285     system("cls");
286     printf("\tQual paciente deseja excluir?\n");
287     lista_paciente = excluir_paciente
        (lista_paciente);
288     break;
289 case 0:
290     exit(0);
291     break;
292 default:
293     printf("\n\n\n");
294     printf("Opcao invalida\n");
295     Sleep(1500);
296     printf("\n\n\n");
297 }
298 system("cls");
299 } while (op_menu != 0);
300 }
301 //Função principal
302 int main(){
303     -
304     while(1)
305     {
306         -
307         char string[5];
308         char enter[2];
309         -
310         printf("\n Digite a senha : ");
311         string[0] = getch();
312         printf ("**");
313         string[1] = getch();
314         printf ("**");
315         string[2] = getch();
316         printf ("**");
317         string[3] = getch();
318         printf ("**");
319         string[4] = getch();
320         -
321         system("color F1");
322         system("cls"); //limpa a tela
323         printf ("\n\n\t\tAcesso
            permitido\n\n\t\tParabens!\n\n\t\tTecle
            Enter para Acessar o
            Sistema!\n\n\n\n\n\n");
324         getch(); // quando pressionar o "ENTER"
            continue o programa
325         system("cls");
326         system("cls"); //limpa a tela
327         printf ("\n\n\t\tAcesso
            permitido\n\n\t\tParabens!\n\n\t\tTecle
            Enter para Acessar o
            Sistema!\n\n\n\n\n\n");
328         getch(); // quando pressionar o "ENTER"
            continue o programa
329         system("cls"); //limpa a tela
330     }
331     else
332     {
333         system("color F1");
334         system("cls"); //limpa a tela
335         printf ("\n\n\t\tAcesso negado!\n\n\t\tVoce
            não tem permissao para acesso! Tecle
            Enter Para Tentar
            Novamente!\n\n\n\n\n\n\n\n\n\n");
336         system("pause>null");
337         system("cls"); //limpa a tela
338         -
339         return main();
340     }
341     -
342     system("color F1"); //Muda a cor do programa
343     setlocale(LC_ALL, "Portuguese");
344     entrando(); //chama a função responsável por
            realizar a animação
345     menu(); //chama o menu principal
346     system("pause>null");
347     }
348 }
349 -
350 -
351     system("color F1"); //Muda a cor do programa
352     setlocale(LC_ALL, "Portuguese");
353     entrando(); //chama a função responsável por
            realizar a animação
354     menu(); //chama o menu principal
355     system("pause>null");
356     }
357 }

```

Fonte: Elaborado pelos alunos.

## 4.5 Teste do programa

Como encontrar o "Arquivo Fonte"?

1. Abra a pasta "SISCOV19";
2. Encontre o arquivo "main.c";
3. Para abrir o "código fonte" clique 2 vezes no arquivo;
4. O IDE irá abrir arquivo e mostrar o "código fonte" na tela.

## 5. CONCLUSÃO

Diante do risco de saúde iminente ocasionado pela pandemia do covid-19, os governos globais tiveram que buscar novos recursos para ter um melhor entendimento do problema e sua amplitude.

Para atender a estas novas exigências governamentais, o setor de tecnologia tem procurado oferecer programas, sistemas e ferramentas de grande utilidade no combate a este vírus.

Este trabalho atendeu ao objetivo que era o de desenvolver um programa em linguagem C, para ser utilizado em hospitais e clínicas, no cadastramento dos pacientes que foram diagnosticados com a covid-19.

Por meio deste programa desenvolvido, o usuário poderá utilizá-lo, fazendo uso das informações contidas em seu banco de dados, para consultas, planejamentos, análises e projeções. Além de ser uma fonte segura de informações que poderão ser utilizadas pelas secretarias das cidades, que poderão ser utilizadas no desenvolvimento de políticas públicas de combate desta pandemia.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

MATERIAIS DIDÁTICOS DISPONIBILIZADOS PELA UNIP.

BRASIL. **Ciência e tecnologia frente à pandemia, 2020.** Disponível em: <<https://www.ipea.gov.br/cts/pt/central-de-conteudo/artigos/artigos/182-corona>>. Acesso em 14 out. 2021.

MATERIAIS DIDÁTICOS DISPONIBILIZADOS PELA UNIP.

LOLE, A.; STAMPA, I.; GOMES, R. L. R. **Para além da quarentena: reflexões sobre crise e pandemia, 2020.** Disponível em: <<https://morula.com.br/wp-content/uploads/2020/06/ParaAlemDaQuarentena.pdf>>. Acesso em 28 out. 2021.

MANZANO, J. A. N. G. **Estudo dirigido de linguagem.** São Paulo: Editora Érica, 2013.

MORAES, P. S. de. **Curso básico de lógica de programação, 2020.** Disponível em: <<https://ftp.unicamp.br/pub/apoio/treinamentos/logica/logica.pdf>>. Acesso em 14 out. 2021.

RODRIGUES, J. M. C.; DOS SANTOS, P. M. G. **Reflexões e desafios das novas práticas docentes em tempo de pandemia, 2020.** Disponível em: <<http://www.ccta.ufpb.br/editoraccta/contents/titulos/educacao/reflexoes-e-desafios-das-novas-praticas-docentes-em-tempos-de-pandemia/reflexoes-e-desafios-das-novas-praticas-docentes-em-tempos-de-pandemia.pdf>>. Acesso em 05 nov. 2021.