

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**

**ESCOLA POLITÉCNICA**

**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**RELATÓRIO SOBRE PROJETO DE SISTEMA DE COFRE COM  
VETORES**

Felipe Cassante Ganzarolli - RA: 24003492

Felipe Grolla Freitas - RA: 24004846

Gabriel Henrique Pozeti de Faria - RA: 24011960

João Victor Vasconcelos Junqueira Criscuolo - RA: 22024547

Lucas Espica Rezende - RA: 24006575

Campinas – Sp

2024

## **1. INTRODUÇÃO**

O projeto da disciplina PC (Programação de Computadores) tem como propósito a criação de um sistema de controle baseado em senha. Este sistema é projetado para lidar com senhas obtidas no deslizamento de um vetor de 25 dígitos, o qual contém em si os números 0, 2, 4, 6, 8 e 9, distribuídos de maneira randomizada e podendo ser alterados a qualquer momento.

A proposta do projeto é criar um programa que forneça ao usuário 3 tentativas para abrir um cofre que possui senha pré determinada e que pode ser alterada nas configurações do programa a qualquer momento, devendo o projeto em questão ser capaz de validar a senha inserida pelo usuário para garantir o acesso autorizado. Este projeto é um desafio interessante que combina os princípios da engenharia de computação com a necessidade prática de segurança de dados.

## **2. DESCRIÇÃO DO PROBLEMA**

O projeto propõe a criação de um sistema de controle baseado em senha. A complexidade do problema se encontra em vários aspectos como:

### **2.1 Tamanho da Senha**

O sistema deve ser capaz de lidar com informações contidas em um vetor de tamanho 25. Isso implica que o sistema deve ser capaz de processar uma ampla gama de entradas e ainda funcionar de maneira eficiente.

### **2.2 Validação da Senha**

A senha é composta por seis dígitos específicos: 0, 2, 4, 6, 8 e 9. Isso adiciona uma camada de complexidade, pois o sistema deve ser capaz de contar o número de vezes que tais números apareceram durante os deslizamentos, bem como validar se a senha inserida bate com a pré determinada pelo grupo (

composta das rotações 11, 12, 13, 14 e 15 em sequência), rejeitando qualquer entrada que contenha dígitos diferentes.

## **2.3 Interface de Usuário**

O usuário deve ser capaz de alcançar por si os valores que compõem a senha correta. Isso requer uma interface de usuário intuitiva e fácil de usar, que permita ao mesmo inserir sua senha da maneira idealizada pela equipe.

## **2.4 Autorização de Acesso**

Se a senha for validada com sucesso, o sistema deve conceder acesso autorizado ao usuário. Isso implica que o sistema deve ter um mecanismo para permitir ou negar acesso com base nessa validação.

## **3. DIFICULDADES, SOLUÇÕES E DESENVOLVIMENTO DO PROJETO**

O projeto desenvolvido pela equipe incluiu uma lógica de programação que utilizou diversos tipos de loops (while, do while e for), para assegurar que o programa se repetisse durante o número de tentativas e deslizamentos definidos pela atividade; Além disso, foram empregadas estruturas de decisão com if e else, bem como foram utilizadas as bibliotecas `stdio.h`, `locale.h` (para reconhecer caracteres da língua portuguesa) e `stdlib.h` (para formatar a tela do PC). Todos esses aspectos, incluindo a lógica de programação, o desenvolvimento e o funcionamento do sistema, podem ser visualizados na imagem do projeto.

Um desafio enfrentado pelo grupo foi a implementação da barra deslizante, que consiste em definir um intervalo entre dois dígitos, começando em 10 e terminando no limite final determinado pelo usuário.

De início, o grupo criou dois vetores, o primeiro envolvendo os valores de números solicitados na atividade e o segundo utilizado como uma forma de contador para obter a posição real em que se encontravam os números no vetor:

```

#include <stdio.h>
#include <locale.h>
#include <stdlib.h>

int main () {
    setlocale(LC_ALL, "portuguese");
    int vetor[25] = {2,0,4,2,8,6,9,8,9,4,0,2,6,6,8,0,9,0,9,4,2,6,8,8,4};

    int exemplo[25] =
    {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24};

    int contador, tentativas=3, loop,i=10,j, zero, dois, quatro, seis, oito, nove,
    Cofre;
    int senha_correta=632656;

    printf("\t\t Posição | Barra\n");
    printf("\t-----|-----\n");
    for (loop = 0; loop < 25; loop++) {
        printf("\t\t\t %d | %d\n", exemplo[loop], vetor[loop]);
    }
}

```

Contudo, após finalizarem a base do que seria o programa e prosseguirem para a solução de bugs e afins, o grupo se deparou com alguns bugs relacionados a valores digitados pelo usuário que eram maiores ou menores que os limites do vetor, o que ocasionava em utilização de lixo de memória e consequentes problemas no programa.

Tendo isso em vista, o desenvolvimento do código foi reiniciado e, com sucesso, reduziu o número de vetores do programa para apenas um, tendo por consequência disso resolvido os problemas acima mencionados utilizando uma nova lógica de programação.

Outro ponto do projeto foi o sistema de verificação de senha, um componente crucial do nosso sistema de autenticação, que tendo como base uma senha pré-definida armazenada no sistema, ao obter a senha formada pelo input do usuário, compara cada dígito da senha inserida com a senha pré-definida.

Para realizar essa comparação, a equipe desenvolveu uma lógica complexa que utiliza estruturas de bifurcação (if / else) e operadores lógicos (como || e &&).

Essa lógica é responsável por verificar se cada dígito da senha formulada corresponde ao dígito correspondente da senha pré-definida.

A figura abaixo ilustra esse processo detalhadamente, mostrando como os dígitos são comparados utilizando as variáveis (zero, dois, quatro, etc...) utilizadas para contagem de repetição e como a lógica de verificação é aplicada. Além disso, para fornecer uma melhor experiência ao usuário, implementamos um contador de tentativas, que informa ao usuário quantas chances restam para tentar inserir a senha correta. Cada vez que o usuário erra a senha, uma chance é subtraída do contador. Quando o contador atinge zero, o sistema sai do loop de verificação,

encerrando o processo de autenticação. Isso é feito usando a função break para interromper o loop.

```
printf("\n\tSenha: %d%d%d%d%d", zero, dois, quatro, seis, oito, nove);  
    if (zero==6 && dois==3 && quatro==2 && seis==6 && oito==5 && nove==6){  
        printf("\n\tSenha Correta");  
        break;  
    }  
    else{  
        tentativas--;  
        printf("\n\tSenha Incorreta");  
        printf("\n\tTentativas Restantes: %d", tentativas);  
    }  
}while(tentativas>0);
```

O novo código desenvolvido também tem como principal diferença a utilização da função switch case, para contagem das repetições dos números para composição da senha.

De início, a contagem dos números era feita via o comando if, tendo cada número possível uma condicional própria.

Tal utilização era possível e funcional, contudo, buscando uma melhor otimização de seu programa, o grupo adotou tal mudança.

Como pode ser observado na figura abaixo:

```

while(tentativas>0){
do{
printf("\n\t Número: ");
scanf("%d", &f);
copla=i;
if(f<0){
break;
}
else if(f>=25){
printf("\n\tValor inválido\n\tDigite um valor entre 0 e 24\n");
contador--;
f=copia;
}
if( i>0 && i<f){
for(i; i<=f; i++){
printf("\t%d\n", vetor[i]);
switch (vetor[i]){
case 0:
zero++;
break;

case 2:
dois++;
break;

case 4:
quatro++;
break;

case 6:
seis++;
break;

case 8:
oito++;
break;

case 9:
nove++;
break;
}
}
}
}
}

```

Por fim, o restante do tempo do grupo foi utilizado para deixar a experiência do usuário final mais convidativa, bem como para tentar elaborar auxílios visuais para embelezar o programa:

```

printf("\n\n\n");
printf("\t*****\n");
printf("\t*                               *\n");
printf("\t* *****                               *\n");
printf("\t*      *                               *\n");
printf("\t*      *                               *\n");
printf("\t*      *                               *\n");
printf("\t*      *                               *\n");
printf("\t*      *                               *\n");
printf("\t*      *      *                               *\n");
printf("\t*      *      *      *      *      *\n");
printf("\t*      *      *      *      *      *\n");
printf("\t*      *      *      *      *      *\n");
printf("\t*      *      *      *      *      *\n");
printf("\t*      *      *      *      *      *\n");
printf("\t*****\n");
printf("\t*                               *\n");
printf("\t*****\n");
printf("\n\n\n");

```

(Tentativa de confecção de cofre fechado no início do programa)

```

printf("\n\tSenha Correta!\n");
printf("\n\tO cofre foi aberto!\n\tParabéns!\n");
printf("\t*****\n");
printf("\t*                               *\n");
printf("\t* *****                               *\n");
printf("\t*      *                               *\n");
printf("\t*      *                               *\n");
printf("\t*      *                               *\n");
printf("\t*      *                               *\n");
printf("\t*      *$                               *\n");
printf("\t*      *$$                               $*\n");
printf("\t*      *$$$                               $$$*\n");
printf("\t*      *$$$$                               $$$$$*\n");
printf("\t*      *$$$$$$$$                               $$$$$$*\n");
printf("\t*      *&$$$$$$$$                               $$$$$$*\n");
printf("\t*      *****                               *\n");
printf("\t*                               *\n");
printf("\t*****");

```

(Tentativa de confecção de cofre aberto com a presença de cédulas de dinheiro caso a senha seja corretamente digitada)

#### 4. REFERÊNCIAS

GUIMARÃES, Lúcia Filomena de Almeida. **Atividade\_Avaliativa\_1\_T0102. 2024.**

Disponível em

<<https://puc-campinas.instructure.com/courses/51258/assignments/159945>

?module\_item\_id=361221.>. Acesso em 01 de abril de 2024.