

Projeto Integrativo



Acadêmicos: André Filho Nunes dos Santos, Clara Luz Lopes Dias da Cruz,

Guilherme Miguel Pereira Oliveira, Kayran Henrique Reis Teixeira, Yuri Peixoto de Souza

Ciclo 1 – 2024/2

Projeto Integrador 2024/2 - Engenharia de Software

1. Estrutura de Dados

Ciclo 1

Descrição da atividade: Estruturar o projeto utilizando alocação dinâmica de memória e armazenado as informações fazendo uso de registros e listas encadeadas.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include <locale.h>
```

```
struct pedido {
```

```
    int id;
```

```
    int status;
```

```
};
```

```
struct pedido* pedidos = NULL;
```

```
int total_pedidos = 0;
```

```
void cria_pedido() {
```

```
    struct pedido novo_pedido;
```

```
while (1) {
```

```
    printf("Digite o id do pedido: ");
```

```
    if (scanf("%d", &novo_pedido.id) != 1) {
```

```
        printf("Erro ao ler o id.\n");
```

```
        while (getchar() != '\n');
```

```
        continue;
```

```
    }
```

```
    printf("Digite o status do pedido (0 - finalizado, 1 - em andamento): ");
```

```
    if (scanf("%d", &novo_pedido.status) != 1 || novo_pedido.status < 0 ||  
novo_pedido.status > 1) {
```

```
        printf("Erro ao ler o status.\n");
```

```
        while (getchar() != '\n');
```

```
        continue;
```

```
    }
```

```
    break;
```

```
}
```

```
total_pedidos++;
```

```
pedidos = (struct pedido*) realloc(pedidos, total_pedidos * sizeof(struct pedido));
```

```
if (pedidos == NULL) {
```

```
    printf("Erro de alocação de memória!\n");
```

```
        exit(1);

    }

    pedidos[total_pedidos - 1] = novo_pedido;

    printf("Pedido criado com sucesso!\n");

}
```

```
void remover_pedido() {

    if (total_pedidos == 0) {

        printf("Não há pedidos para remover.\n");

        return;

    }

    int id;

    printf("Digite o id do pedido a ser removido: ");

    scanf("%d", &id);
```

```


    int encontrado = 0;

    for (int i = 0; i < total_pedidos; i++) {

        if (pedidos[i].id == id) {

            encontrado = 1;

            for (int j = i; j < total_pedidos - 1; j++) {

                pedidos[j] = pedidos[j + 1];
```

```
}  
  
total_pedidos--;  
  
pedidos = (struct pedido*) realloc(pedidos, total_pedidos * sizeof(struct pedido));  
  
printf("Pedido removido com sucesso!\n");  
  
break;  
  
}  
  
}
```

```
if (!encontrado) {  
    printf("Pedido com id %d não encontrado.\n", id);  
}  
}
```

```
void ver_pedidos_em_andamento() {  
    if (total_pedidos == 0) {  
        printf("Não há pedidos em andamento.\n");  
        return;  
    }  
  
    printf("Pedidos em andamento:\n");  
  
    for (int i = 0; i < total_pedidos; i++) {  
        if (pedidos[i].status == 1) {  
            printf("ID: %d\n", pedidos[i].id);  
        }  
    }  
}
```



```
}  
  
}  
  
}  
  
int menu() {  
  
    int opcao;  
  
    printf("\nBEM-VINDO AO SISTEMA DE GERÊNCIA E CONTROLE DE PEDIDOS\n");  
  
    printf("1 - Criar um pedido e adicionar à fila\n");  
  
    printf("2 - Remover um pedido\n");  
  
    printf("3 - Ver pedidos em andamento\n");  
  
    printf("0 - Fechar a execução do programa\n");  
  
    printf("Escolha uma opção: ");  
  
    scanf("%d", &opcao);  
  
    return opcao;  
}  
  
int main() {  
  
    setlocale (LC_ALL, "portuguese");  
  
    int opcao;  
  
    do {  
  
        opcao = menu();  
  
        switch (opcao) {
```

case 1:

cria_pedido();

break;

case 2:

remover_pedido();

break;

case 3:

ver_pedidos_em_andamento();

break;

case 0:

printf("Encerrando o programa...\n");

break;

default:

printf("Opção inválida!\n");

}

} while (opcao != 0);

free(pedidos);

return 0;

}

Ciclo 2

Descrição da atividade: Estruturar o código agora utilizando listas duplamente encadeadas ou listas circulares.

Insira a atividade aqui!

Comentário adicional: [Opcional]

Ciclo 3

Descrição da atividade: Apresentação do SINACEN e envio do trabalho.

Insira a atividade aqui!

Comentário adicional: [Opcional]

2. Inglês Instrumental

Ciclo 1

Descrição da atividade: Formar grupos de 4 a 5 pessoas e escolher um tema relacionado à Engenharia de Software, como "Desenvolvimento Ágil", "Segurança da Informação", "Inteligência Artificial" ou "User Experience (UX)". Cada grupo realizará uma pesquisa em materiais em inglês, analisando como o tema se conecta a outras disciplinas do curso. Os alunos deverão apresentar um relatório técnico com suas pesquisas e análises.

INTEGRATING USER EXPERIENCE WITH CORE SOFTWARE DESIGN PRINCIPLES

The integration of User Experience (UX) principles with core concepts in software development involves combining Object-Oriented Programming (OOP), database design, operational research, and data structures.

OOP principles such as encapsulation, inheritance, and modularity directly support UX by enabling the creation of flexible, scalable, and reusable software components. This modularity allows developers to create user-friendly interfaces that can easily adapt to changing user requirements.

Well-structured databases ensure responsive and data-driven applications that align with user needs as it provides fast and reliable access to data, which enhances the overall user experience.

Furthermore, operational research techniques such as optimization and decision-making algorithms can be applied to improve system efficiency and user satisfaction. For example, optimizing workflows or navigation paths within an interface can reduce the time and effort users spend completing tasks.

Finally, efficient data structures contribute to smoother user interactions by improving performance and responsiveness, especially in interactive applications where real-time feedback is important, well-designed data structures allow for faster data retrieval and manipulation leading to a seamless user experience.

Comentário adicional: Keywords: User experience; Object oriented programming; Database design; Operational research.

Ciclo 2

Descrição da atividade: Com base na atividade desenvolvida no Ciclo 1, criar um vídeo curto (5-7 minutos) explicando o tema em inglês e um infográfico interativo destacando os principais conceitos e conexões interdisciplinares. Os grupos apresentarão seus trabalhos em uma sessão de "Tech Talks", onde deverão interagir com os colegas em inglês. Após as apresentações, fornecerão feedback construtivo sobre os trabalhos dos outros grupos e produzirão um relatório final em inglês.

Insira a atividade aqui!

Comentário adicional: [Opcional]

Ciclo 3

Descrição da atividade: Apresentação do SINACEN e envio do trabalho.

Insira a atividade aqui!

Comentário adicional: [Opcional]

3. Pesquisa Operacional

Ciclo 1

Descrição da atividade: Formule um modelo matemático de programação linear para otimizar um aspecto do projeto integrador, como minimizar os custos de desenvolvimento ou maximizar a eficiência do uso de recursos. Resolva o problema utilizando o Excel, apresentando as soluções possíveis e discutindo qual seria a melhor estratégia para o projeto.

Problema de gerência de recursos: Uma empresa de jateamento de peças metálicas, trabalha com dois tipos de serviços para atender seus clientes. O serviço A trabalha apenas com peças sem pinturas, este serviço obtém um lucro de 100,00 R\$, com custo de 30 minutos na máquina M1 e 10 minutos na máquina M2. Já o serviço B trabalha apenas com peças que possuem pinturas, este serviço obtém um lucro de 220,00 R\$, com custo de 1 hora e 20 minutos na máquina M1 e 30 minutos na máquina M2. Levando em consideração este contexto e que a empresa dispõe de apenas 12 horas de trabalho diárias da máquina M1 e de 8 horas de trabalho diária na máquina M2. Modele o problema para maximizar o lucro da empresa e respeitando as restrições impostas.

Maximizar $Z = 100x + 220y$			x1	x2	condição	resultado
$30x1 + 80x2 \leq 720$		R1	30	80	\leq	720
$10x1 + 30x2 \leq 480$		R2	10	30	\leq	480
$x1 \geq 0$		FO	100	220		
$x2 \geq 0$						
Variáveis	Resultado	Restrições	Resultado		FO	
x1 =	24	r1	720		2400	
x2 =	0	r2	240			

Podemos concluir que a solução do problema de gerência de recurso é, que a empresa de jateamento deve trabalhar apenas com o serviço A. Pois ao trabalhar apenas com este serviço, a empresa pode jatear 24 peças por dia e obter um lucro de 2400 R\$ por dia, assim maximizando seus lucros diários.

Ciclo 2

Descrição da atividade: Utilizando o modelo matemático desenvolvido no Ciclo 1, aplique o Método Simplex para encontrar a solução ótima do problema. Compare os resultados obtidos com a solução gráfica anterior e discuta as diferenças e implicações para o projeto integrador.

Insira a atividade aqui!

Comentário adicional: [Opcional]

Ciclo 3

Descrição da atividade: Apresentação do SINACEN e envio do trabalho.

Insira a atividade aqui!

Comentário adicional: [Opcional]

4. Programação Orientada a Objetos

Ciclo 1

Descrição da atividade: Desenvolver e Aplicar os Conceitos associados aos pilares do Paradigma de Programação Orientado a Objetos, utilizando estruturas de linguagens de programação Orientadas a Objetos, empregando conceitos de reuso e padrões de projetos na prática em laboratórios por meio dos estudos de casos que orientarão a criação dos Projetos Televisor, Geladeira, Fogão, Automóvel e Objetos Geométricos. Atividade prática utilizando a Linguagem de Programação JAVA em IDE Eclipse.

```
public class Piece {
```

```
    private String name;
```

```
    private boolean hasRust;
```

```
    private boolean hasPaint;
```

```
    public Piece(String name, boolean hasRust, boolean hasPaint) {
```

```
        this.name = name;
```

```
        this.hasRust = hasRust;
```

```
this.hasPaint = hasPaint;

}

public String getName() {

    return name;

}

public boolean hasRust() {

    return hasRust;

}

public void setHasRust(boolean hasRust) {

    this.hasRust = hasRust;

}

public boolean hasPaint() {


    return hasPaint;

}

public void setHasPaint(boolean hasPaint) {

    this.hasPaint = hasPaint;

}
```



@Override

```
public String toString() {  
  
    return "Piece: " + name + " | Rust: " + (hasRust ? "Yes" : "No") + " | Paint: " +  
    (hasPaint ? "Yes" : "No");  
  
}  
  
}
```

```
public class SandBlaster {  
  
    public void cleanPiece(Piece piece) {  
  
        if (piece.hasRust() || piece.hasPaint()) {  
  
            System.out.println("Cleaning piece: " + piece.getName());  
  
            piece.setHasRust(false);  
            piece.setHasPaint(false);  
  
            System.out.println("Piece " + piece.getName() + " has been successfully  
cleaned!");  
  
        } else {  
  
            System.out.println("Piece " + piece.getName() + " is already clean.");  
  
        }  
  
    }  
  
}
```

```
import java.util.Scanner;
```



```
public class Main {  
  
    public static void main(String[] args) {  
  
        Scanner scanner = new Scanner(System.in);  
  
  
        Piece piece1 = new Piece("Piece 1", true, true);  
  
        Piece piece2 = new Piece("Piece 2", false, true);  
  
        Piece piece3 = new Piece("Piece 3", true, false);  
  
  
        SandBlaster sandBlaster = new SandBlaster();  
  
        System.out.println(piece1);  
        System.out.println(piece2);  
        System.out.println(piece3);  
  
  
        sandBlaster.cleanPiece(piece1);  
  
        sandBlaster.cleanPiece(piece2);  
  
        sandBlaster.cleanPiece(piece3);  
  
  
        System.out.println("\nStatus after cleaning:");  
  
        System.out.println(piece1);  
  
        System.out.println(piece2);  
  
        System.out.println(piece3);  
    }  
}
```

```
scanner.close();
```

```
}
```

```
}
```

Ciclo 2

Descrição da atividade: Desenvolver e Aplicar os Conceitos associados aos pilares do Paradigma de Programação Orientado a Objetos, utilizando estruturas de linguagens de programação Orientadas a Objetos, empregando conceitos de reuso e padrões de projetos na prática em laboratórios por meio dos estudo de casos que orientarão a criação do Projeto Banco e Contas Bancarias, reforçando os padrões de Projeto Singleton , Abstract Factory, Factory Method e Facade. Atividade prática utilizando a Linguagem de Programação JAVA em IDE Eclipse.

Insira a atividade aqui!

Comentário adicional: [Opcional]

Ciclo 3

Descrição da atividade: Apresentação do SINACEN e envio do trabalho.

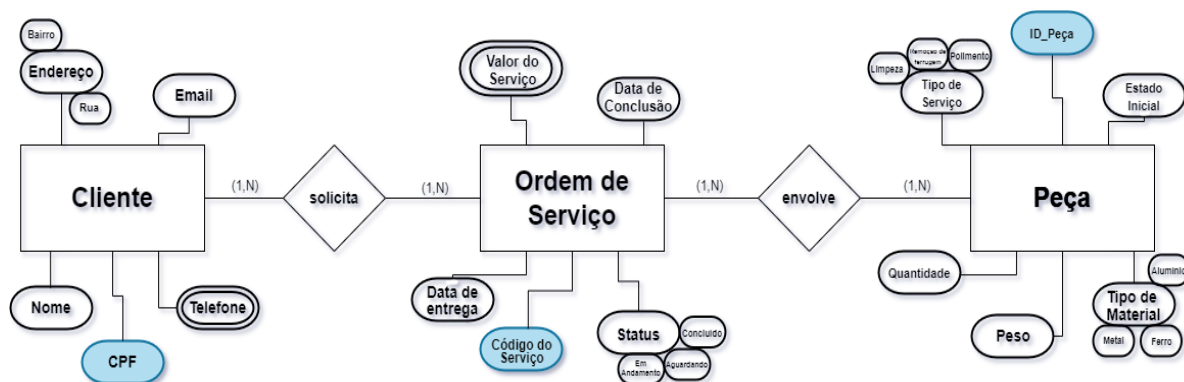
Insira a atividade aqui!

Comentário adicional: [Opcional]

5. Projeto de Banco de Dados

Ciclo 1

Descrição da atividade: Esboce um MER referente a sua aplicação. Utilize os conceitos discutidos em aula, como entidades, relacionamentos e atributos. Além disso, aplique cardinalidade nos referidos relacionamentos.



Ciclo 2

Descrição da atividade: crie um diagrama entidade relacionamento (DER). Nele deverá conter as tabelas e seus respectivos relacionamentos. Além disso, deverá ser declarada em cada atributo o tipo de dado que será inserido.

Insira a atividade aqui!

Comentário adicional: [Opcional]

Ciclo 3

Descrição da atividade: Apresentação do SINACEN e envio do trabalho.

Insira a atividade aqui!

Comentário adicional: [Opcional]