

**PROJETO INTERDISCIPLINAR**

**Restaurante**

Praticidade e liberdade para o futuro

**Alunos:**

|  |  |
| --- | --- |
| **RGM** | **Nome** |
| 5829440696 | Bruno William Calixto |
| Xxxxxxxx | xxxxxxxxxxxxxxxxxxxxxxxxxxxx |
| Xxxxxxxx | xxxxxxxxxxxxxxxxxxxxxxxxxxxx |
| Xxxxxxxx | xxxxxxxxxxxxxxxxxxxxxxxxxxxx |
| Xxxxxxxx | xxxxxxxxxxxxxxxxxxxxxxxxxxxx |

(Nome em ordem alfabética)

Guarulhos - SP

2022

**UNIVERSIDADE CRUZEIRO DO SUL**

**PROJETO INTERDISCIPLINAR**

**Restaurante**

Praticidade e liberdade para o futuro

## “Trabalho apresentado como parte do requisito para aprovação na Disciplina de Projeto Interdisciplinar do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Cruzeiro do Sul.”

**Orientadores:** Prof. Jéssica e Prof. Gilberto

Guarulhos - SP

2022

**Sumário**

[1. Apresentação: 3](#_Toc48489109)

[1.1 Justificativa e Motivação 3](#_Toc48489110)

[1.2 Dados do Sistema. 3](#_Toc48489111)

[2 Requisitos de Técnica de Desenvolvimento de Algoritmos 3](#_Toc48489112)

[3 Requisitos de Programação Orientada a Objetos 3](#_Toc48489113)

[4 Consideração finais 3](#_Toc48489114)

[5 Referencias 3](#_Toc48489115)

[APENSO 1 – Cronograma de entrega de atividades. 4](#_Toc48489116)

# 1. Apresentação:

## 1.1 Justificativa e Motivação

Nossa equipe, pensou inteiramente em um projeto que fosse baseado em como o usuário poderia simplificar seu padrão em questão de pedido de refeições em um restaurante de sua preferência. Por isso realizamos uma implementação de sistema bastante simples e objetiva, visando não só a satisfação com o cliente, mas também a redução de custos a longo prazo para o restaurante que deseje usufruir de nossos sistemas, assim como a redução de resíduos sólidos na natureza (impressão de pedidos e anotações em comandas) e de custo.

## 1.2 Dados do Sistema.

O sistema persiste em realizar a devida praticidade do que vivenciamos hoje, ou seja, de acordo com os tempos, hoje vivenciamos a praticidade de cardápios via web, onde podemos acessá-lo com QR Code, porém, é incomum solicitarmos um pedido com o auxilio de um sistema integrado ao restaurante. Por isso nos dedicamos inteiramente ao desenvolvimento que estamos prestes à exibir neste projeto. Basicamente, o projeto consiste primeiramente em um sistema de geração de comanda, onde isso além de facilitar a alocação de um colaborador para outras áreas distintas, também se dá à autonomia para o usuário em se satisfazer com a maneira que desejar em relação ao desejo de seu pedido. Sabendo disto, proporcionamos que o indivíduo utilize este sistema com uma breve solicitação de identificação, como exibindo seu nome, CPF e a mesa que está alocado.

Logo em seguida, proporcionamos a experiência ao usuário para o acesso exclusivo de seu menu e sua comanda. Podendo então ser exibido em diferentes etapas de processo.

**1 – Ver cardápio:**

Aqui vemos que o usuário ao digitar a opção número 1 do menu, possa visualizar inteiramente o cardápio que pode ser devidamente atualizado pelo administrador do sistema, e devido sua praticidade, também se incorpora ao benefício de diminuir a gestão de impressões de cardápios, o gasto repetitivo de materiais e atualizações desnecessárias em tabelas de preços.

**2- Selecionar Itens:**

Com essa opção, proporcionamos ao usuário, selecionar os itens do pedido através do ID do pedido, podendo assim, selecionar a quantidade de itens desejados onde o mesmo se endireita ao tópico “Comprar”;

**3 – Comprar:**

Aqui o cliente informa sua opção de pagamento, e o mesmo se gera a comanda e a solicitação de finalização do programa, onde o usuário tem a opção de dar continuidade em seu pedido, ou encerrar o processo;

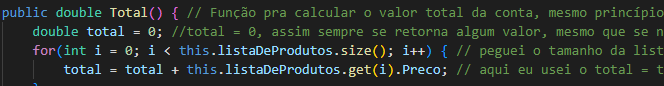
Diante desse menu, irá mostrar todos os dados do cliente como o nome, cpf, número da mesa, forma de pagamento e o total do pedido solicitado e com isso o programa finaliza automaticamente e armazenando futuramente em um banco de dados como histórico ou resgate de pedidos.

# 2 Requisitos de Técnica de DESENVOLVIMENTO DE ALGORITMOS

À princípio, para uma um breve entendimento sobre o que se foi feito dentro de nosso sistema, utilizamos os comentários e os filtros para que pudéssemos também, utilizar as boas práticas de programação, visando um destaque futuro no mercado, de trabalho, onde aos referidos dentro do projeto, poderão ter acesso aos comentários aqui referidos.

Pelo que vimos e presenciamos diante do aprendizado diante da matéria de Técnica e Desenvolvimento de Algoritmos, entendemos que poderíamos utilizar sempre como um esboço padrão, o método e o modo de como seria desenhado o projeto. Acabamos utilizando como alguns dos seguintes exemplos:

**O uso de operações numéricas com variáveis de números reais e inteiros:**



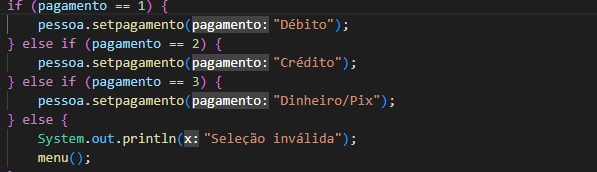
**O uso de variáveis do tipo texto:**

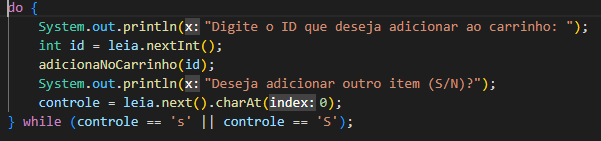


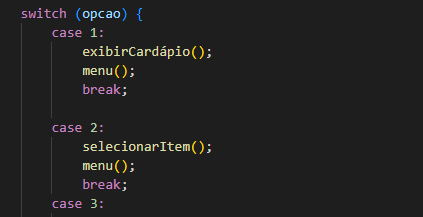
**Comando de escrita e de leitura:**



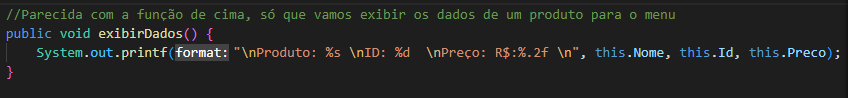
**Estruturas condicionais** **como Se, Senão, Senão se, repetição e escolha de casos:**



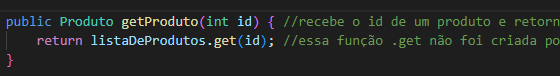




**Aplicações de métodos:**

****

****

****

# 3 Requisitos de Programação Orientada a Objetos

Diante das devidas aulas aplicadas, vimos a importância de aplicações

# 4 Consideração finais

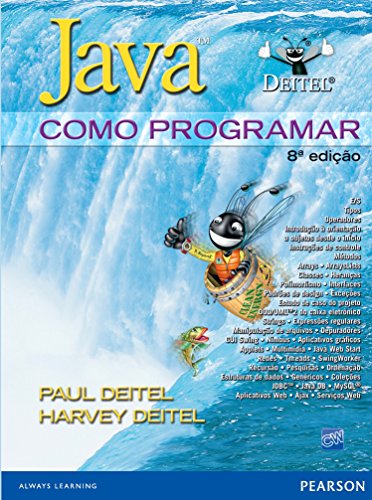
Gostaríamos aqui, de nos expressar pela inteira gratidão que tivemos, inclusive de nossos mentores de aula Jéssica e Gilberto pela excelência do desafio propostos. Afim de agradecer a toda equipe envolvida nos testes do programa, pesquisas de soluções de bugs e afins, e também o grande empenho realizado para a solução de desafios, temos muito o que aprender ainda nos quesitos técnicos de desenvolvimento de sistemas, mas é uma garantia que seguiremos em plena ascensão no desenvolvimento geral. Nossas considerações diante desse desafio, nos proporcionaram a buscar por um profundo conhecimento tanto em nosso desenvolvimento técnico, como também cognitivo. Tivemos também a oportunidade de presenciar grandes dificuldades em questões de interligação de sistemas, ou seja, como eles podiam se conectar com os blocos de programação, visto que escolhemos um tema bem desafiador e distinto do que poderia ser feito com outros temas propostos.

Porém, diante de tantas dificuldades propostas no Projeto Interdisciplinar de nossos cursos, tantas contamos com a ajuda um de nossos grandes amigos, de nome Vinicius, com quem pudemos obter experiências com o desenvolvimento JAVA para justamente sanar e buscar o entendimento de todos os problemas executados em prática.

# 5 BIBLIOGRAFIA

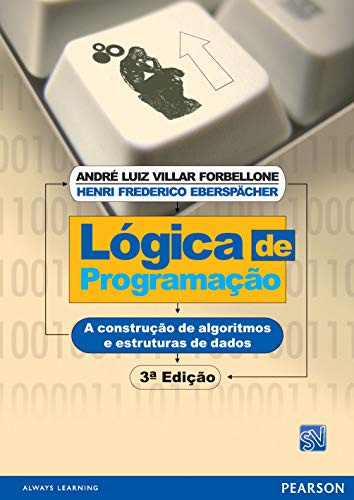
**JAVA:**

DEITEL, Paul; DEITEL, Harvey. Java : como programar. 8.ed. São Paulo: Pearson, 2010.



**TDA:**

**• FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de Programação: A Construção de Algoritmos e Estrutura de Dados. 3. ed. São Paulo: Pearson Prentice Hall, 2008**

****

# Apêndice 1 – Cronograma de entrega de atividades.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Descrição** | **Data** | | **Prazo do cronograma em semanas** | | | | | | | | | | | | | | | | | |
| **Início** | **Término** |  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| **1** | Definição do tema e planejamento inicial | **20/09/22** | **21/09/22** |  | **P** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **R** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** | Definindo Métodos de aplicação | **28/09/22** | **06/10/22** |  |  | **P** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | **R** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3** | Colocando em Pratica o projeto | **01/10/22** | **02/10/22** |  |  | **P** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | **R** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4** | Primeira semana de projeto | **08/10/22** | **10/10/22** |  |  |  | **P** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | **R** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **5** | Segunda semana de projeto | **10/10/22** | **18/10/22** |  |  |  | **P** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | **R** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **6** | Definições finais do projeto | **19/10/22** | **26/10/22** |  |  |  |  |  | **P** |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | **R** |  |  |  |  |  |  |  |  |  |  |  |  |
| **7** | Entrega do projeto final e apresentação |  |  | **P** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **R** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Desenvolvimento de Aprendizado das aulas de POO

**•** TIPOS DE DADOS E VARIÁVEIS

São espaços ou alocações nas quais os dados são armazenados. A declaração de uma variável instrui o programa a reservar um espaço na memória, para que seja possível armazenar um dado de determinado tipo.

**•** ENTRADA E SAÍDA

Indica uma chamada para o método showMessageDialog da classe JOptionPane,exige dois argumentos, separados por vírgula.

**•** CONVERSORES DE TIPO

String para outros tipos

**•** CONSTRUTORES

São trechos de código invocados automaticamente quando um objeto é instanciado.

**•** MODELO DE MEMÓRIA

O runtime do Java (JRE – ambiente em tempo de execução) separa a memória em dois espaços chamados de Stack e Heap, as variáveis de tipos primitivos são armazenadas na Stack.

**•** METODOS

São conjuntos de instruções que cumprem uma tarefa específica e devolvem ou não um valor ao programa que o chamou.

**•** ESTRUTURAS DE DECISÃO

As estruturas condicionais servem para impor uma condição para que algo seja executado.

**•** ESTRUTURA SWITCH-CASE

É uma forma simples para se definir diversos desvios no código a partir de uma única variável.

**•** MÉTODOS OBJETOS RECEBENDO E RETORNANDO

Um método pode receber por parâmetro ou retornar qualquer tipo de informação, inclusive um objeto.

**•** SOBRECARGA DE MÉTODOS

Podemos criar vários métodos com o mesmo nome em uma classe, as ações são parecidas, mas os parâmetros necessários

devem ser diferentes.

**•** ESTRUTURAS DE REPETIÇÃO

Também conhecidas como laços (loop), utilizadas para executar repetidamente uma instrução ou um bloco de instruções

enquanto uma determinada condição for verdadeira.

**•** ENCAPSULAMENTO DE DADOS

O encapsulamento é as vezes referido como ocultamento de informações. Os usuários dos objetos não conhecem sua

constituição e os utilizam através dos métodos públicos.

**•** MODIFICADORES DE ACESSO

Há três modificadores de acesso: public, protected e private;

Nesse projeto usamos o modificador de acesso public.

**•** HERANÇA

Como o próprio nome sugere, na orientação a objetos o termo herança se refere a algo herdado, em Java, a herança ocorre

quando uma classe passa a herdar características (atributos e métodos) definidas em uma outra classe, especificada como

sendo sua ancestral ou superclasse. A técnica da herança possibilita o compartilhamento ou reaproveitamento de recursos

definidos anteriormente em uma outra classe. A classe fornecedora dos recursos recebe o nome de superclasse e a receptora

dos recursos de subclasse.

**•** CLASSES ABSTRATAS

São criadas quando não se pretende criar objetos a partir delas, são normalmente usadas como superclasses,

**•** MÉTODOS ABSTRATOS

Os métodos abstratos definidos em uma classe abstrata devem obrigatoriamente implementados em uma classe concreta,

mas se uma classe abstrata herdar outra classe abstrata, a classe que herda não precisa implementar os métodos abstratos.

**•** INTERFACES

São estruturas similares às classes abstratas, que definem a especificação de funcionalidades, sem a implementação das mesmas.

**•** POLIMORFISMO

O compilador só conhece o tipo de referência dos objetos (superclasse), verifica os métodos chamados somente na classe da referência.