**GABRIEL LEONARDO DZUMAN**

**projeto integrado ii – ADS, BaCKEND, WEB, MOBILE, DEVOPS**

Curitiba - PR

2021

**GABRIEL LEONARDO DZUMAN**

**projeto integrado ii – ADS, BaCKEND, WEB, MOBILE, DEVOPS**

Projeto Integrado apresentado como requisito parcial para a obtenção do título de Bacharel Tecnólogo em Análise e Desenvolvimento de Sistemas pela Universidade Norte do Paraná.

Professores: Adriane Aparecida Loper; Gilberto Fernandes Junior; Leonardo Santiago Sidon da Rocha; Vanessa Matias Leite.

Curitiba - PR

2021

**RESUMO**

Neste projeto irei trazer algumas pesquisas sobre Lógica Dedutiva, Ferramentas Case e suas modelagens de dados no mercado, e apresentar Pseudocódigo, que nada mais é do que uma forma genérica de transcrever uma linguagem de programação, tendo algumas regras que é necessário atenção, e ao fim deste código, mostrarei o seu resultado com as variáveis informadas no mesmo.

Abordaremos a especificação de requisitos de software, etapa esta, de extrema importância no desenvolvimento, pois ela pode ser determinante para o sucesso do software, definindo os objetivos e funções que o mesmo precisa executar, e por fim, definiremos o que são requisitos funcionais e não-funcionais, exemplificando cada um deles.

|  |  |
| --- | --- |
|  |  |

**sumário**

|  |  |
| --- | --- |
| **1** | **LÓGICA DEDUTIVA** 01 |
| **2** | **FERRAMENTA CASE** 02 |
| **3** | **Título da seção secundária** 03 |
|  | **CONCLUSÃO** 04 |
|  | **REFERÊNCIAS** 05 |
|  |  |
|  |  |

**1 Lógica dedutiva**

Há uma tradição segundo a qual há dois ramos na lógica: a lógica dedutiva e a indutiva. Mais recentemente, as diferenças entre estas disciplinas tornaram-se tão profundas que a maior parte das pessoas usam hoje em dia o termo “lógica” com o significado de lógica dedutiva, reservando termos como “teoria da confirmação” para abranger pelo menos parte do que se costumava chamar “lógica indutiva”. Irei seguir a prática mais recente, interpretando “filosofia da lógica” como “filosofia da lógica dedutiva”. Nesta secção, irei tentar mostrar as diferenças entre as duas disciplinas, e indicar brevemente as razões pelas quais as pessoas pensam que a lógica indutiva não é realmente lógica.

Uma maneira de as premissas de um argumento constituírem boas razões a favor da sua conclusão se siga das suas premissas. Vamos chamar “válido” a qualquer argumento cuja conclusão se siga das duas premissas. Um teste inicial de validade é o seguinte. Perguntamos: será possível que as premissas sejam verdadeiras mas a conclusão falsa? No caso do argumento “O Gabriel é um dramaturgo e alguns dramaturgo são pobres. Logo, Gabriel é pobre” a resposta é “Sim”. Mesmo que alguns dramaturgos sejam pobres, é possível que outros, talvez até a maioria, sejam ricos, e que Gabriel seja um destes outros. Em geral, um argumento é valido unicamente se for impossível que as premissas sejam todas verdadeiras, mas a conclusão falsa. Poderemos ter a esperança de distinguir a lógica dedutiva da indutiva dizendo que a primeira, mas não a segunda, se ocupa da validade?

1. Todos os homens são mortais. Sócrates é um homem. Logo, Sócrates é mortal.
2. O Sol nasceu todas as manhãs até hoje. Logo (é provável que) nasça amanhã.

O primeiro é um exemplo canónico de um argumento classificado, como válido pela lógica dedutiva. O segundo é um argumento que não é classificado como válido pela lógica dedutiva.

Contudo, o lógico indutivo deve atribuir ao último um estatuto favorável qualquer. Sem dúvida que as razões que as premissas do argumento 2 nos dão a favor da sua conclusão são muito melhores do que as razões dadas pela mesma premissa a favor da conclusão oposta:

1. O Sol nasceu todas as manhãs até hoje. Logo, (é provável que) não nasça amanhã.

Isto pode parecer um argumento tolo, mas aparentemente é qualquer coisa como isto que dá vida a alguns apostadores. A “Falácia de Monte Carlo” consiste na crença de que se o vermelho saiu várias vezes na roleta, é mais provável que da próxima vez saia o preto. O lógico dedutivo contrasta os argumentos 1 e 2 dizendo que o primeiro, mas não o segundo, é válido.

O lógico indutivo irá contrastar os argumentos 2 e 3 — provavelmente sem usar a palavra “válido”, mas dizendo talvez que 2, ao contrário de 3, é “individualmente forte”. As premissas de 2, mas não as de 3, fornecem fortes razões a favor da sua conclusão.

As premissas de 1 também fornecem fortes razões a favor da sua conclusão. Como poderemos distinguir razões dedutivas fortes de razões indutivas fortes? Temos uma sugestão perante nós: a verdade das premissas de um dedutivo válido torna a falsidade da sua conclusão impossível, mas isso não acontece no caso de argumentos indutivamente fortes. Outro modo de colocar esta questão é: as razões dadas por um argumento dedutivamente válido são conclusivas: a verdade das premissas garante a verdade da conclusão. Este modo de traçar o contraste adequa-se a 1 e 2. A verdade da premissa de 2 pode tornar a conclusão provável, mas não a garante: não lhe dá certeza.

A lógica indutiva, como a terminologia da força indutiva sublinha, tem de se ocupar de uma relação que obtém num grau maior ou menor.

Algumas razões não conclusivas são mais fortes do que outras.

Assim, ao contrário da lógica dedutiva, que faz uma dicotomia clara entre argumentos válidos e inválidos, a lógica indutiva irá distinguir um contínuo de casos, no qual o argumento 2 talvez fique com uma alta classificação, ao passo que o 3 fica bastante baixo.

A validade dedutiva é, como dizem os lógicos, *monotónica*. Isto é, se começarmos com um argumento dedutivamente válido, então, independentemente das premissas que acrescentamos, teremos no fim um argumento indutivamente válido. A força indutiva não é monotónica: acrescentar premissas a um argumento indutivamente forte pode transformá-lo um argumento indutivamente fraco. Considere-se o argumento 2, que é considerado um paradigma da força indutiva. Suponha-se que acrescentamos as seguintes premissas: há um meteoro enorme que está a viajar na nossa direção; hoje à noite entrará no sistema solar, onde permanecerá numa órbita estável em torno do Sol; irá ficar entre o Sol e a Terra, de modo que a Terra irá ficar permanentemente na sombra. Quando acrescentamos estas premissas, o argumento que resulta está longe de ser forte. (Admiti uma certa interpretação do que quer dizer que o Sol “nasce”. Seja como for que interpretemos esta expressão, é fácil encontrar premissas cuja adição enfraqueça o argumento.)

**2 FERRAMENTA CASE**

Desde1955, engenheiros mecânicos e elétricos trabalhavam com ferramentas manuais rudimentares como: livros, tabelas, réguas de cálculos, calculadoras mecânicas, lápis, pranchetas e outros objetos que possibilitavam que um engenheiro criasse modelos do produto a ser projetado.

Uma década se passou e este mesmo grupo de engenheiros começou a experimentar a engenharia baseada em computador, surgiram os montadores e compiladores. A partir daí a primeira geração de Engenharia de Software Auxiliada por Computador, ou seja, de ferramentas conhecidas atualmente como Ferramentas CASE foi sendo desenvolvida, surgiram os dicionários de dados (1960), os editores textuais (1970) e os editores gráficos (1980).

A segunda geração iniciou-se a partir de 1980, através de ferramentas para documentação e diagramação, sendo seguida pelo surgimento de ferramentas de verificação de análise e projeto, ferramentas de geração automática de código a partir do projeto e ferramentas de automação do projeto. Apenas nos anos 90, com a terceira geração de Ferramentas CASE, desenvolveram-se ferramentas com assistência inteligente, recurso para reutilização e engenharia reversa.

Atualmente, os engenheiros de software dispõem de um número maior e mais variado de ferramentas no mercado, que trazem benefícios ao processo desenvolvimento de produtos de software, a saber: encoraja um ambiente interativo, reduz custos de manutenção, melhora a qualidade do produto de software, agiliza o processo de desenvolvimento, aumenta a produtividade.

Toda ferramenta ou método que auxilie num processo de construção lógica ou física, documentação ou teste pode ser considerada uma Ferramenta CASE.

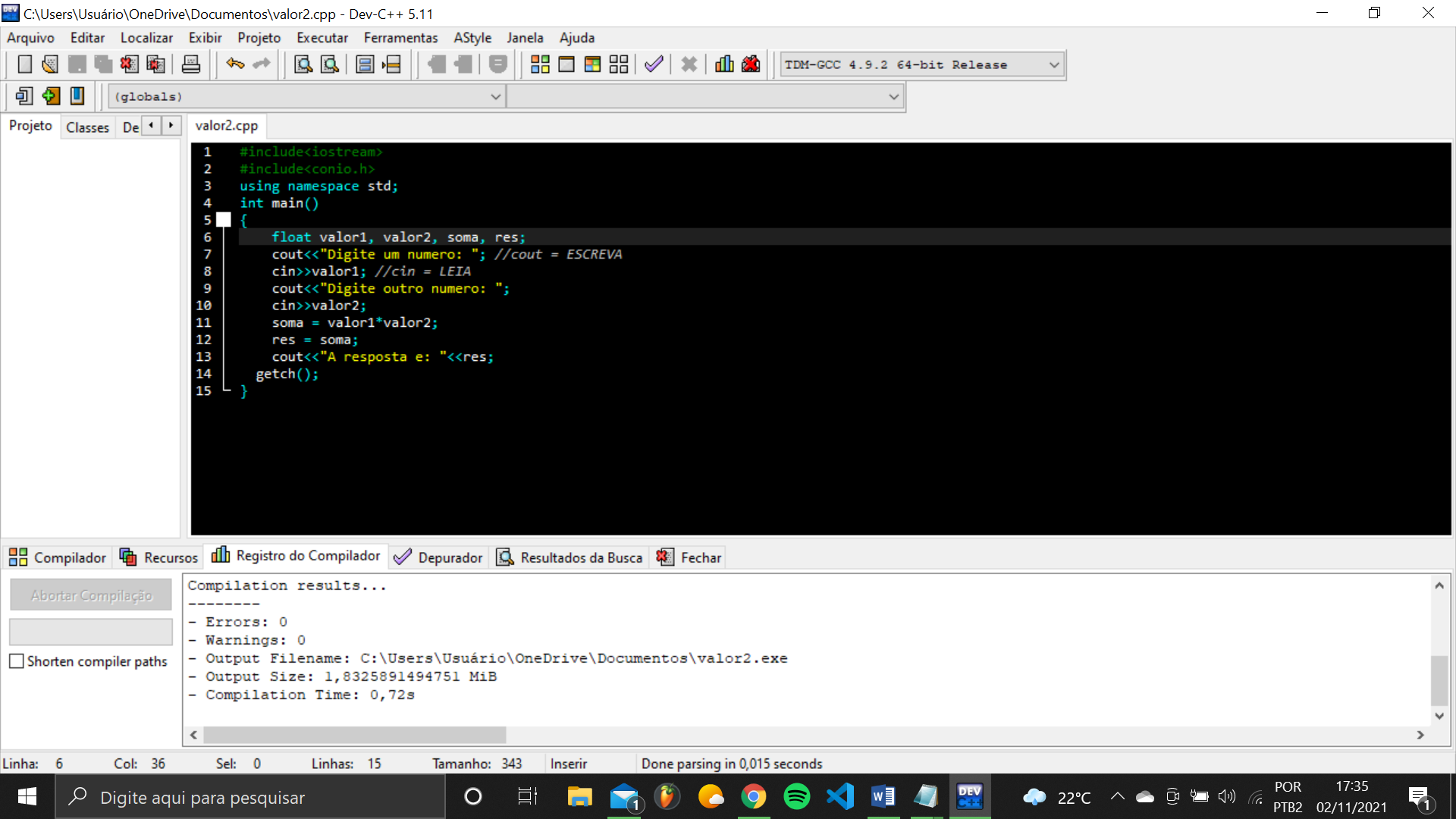
A dissertação presente tem como foco o desenvolvimento de um protótipo de Ferramenta CASE voltada a utilização de usuários finais, apresentado na seção sete, para a verificação de estruturas de base de dados e a partir desta verificação detectar erros com maior eficácia e eficiência. Aproveita-se também para realizar um estudo sobre os diversos tipos de Ferramentas CASE encontradas atualmente no mercado, ressaltando a deficiência no mercado de Ferramentas CASE para usuários finais. Atualmente, a palavra ferramenta está muita associada à ideia de ferramenta de software ou ferramenta automatizada, que é um software feito para auxiliar em alguma tarefa.

A Ferramenta CASE é a solução que mais tem causado revolução hoje em dia nos ADS’s (Ambiente de Desenvolvimento de Software), com ela é possível baixar em muito o tempo total de desenvolvimento de software. Sua presença tronou-se vital para o bom funcionamento de um ADS, auxiliando em todo o ciclo de desenvolvimento (Gerência, Análise, Projeto, Implementação e Teste), além de exercer uma grande importância para a manutenção do software.

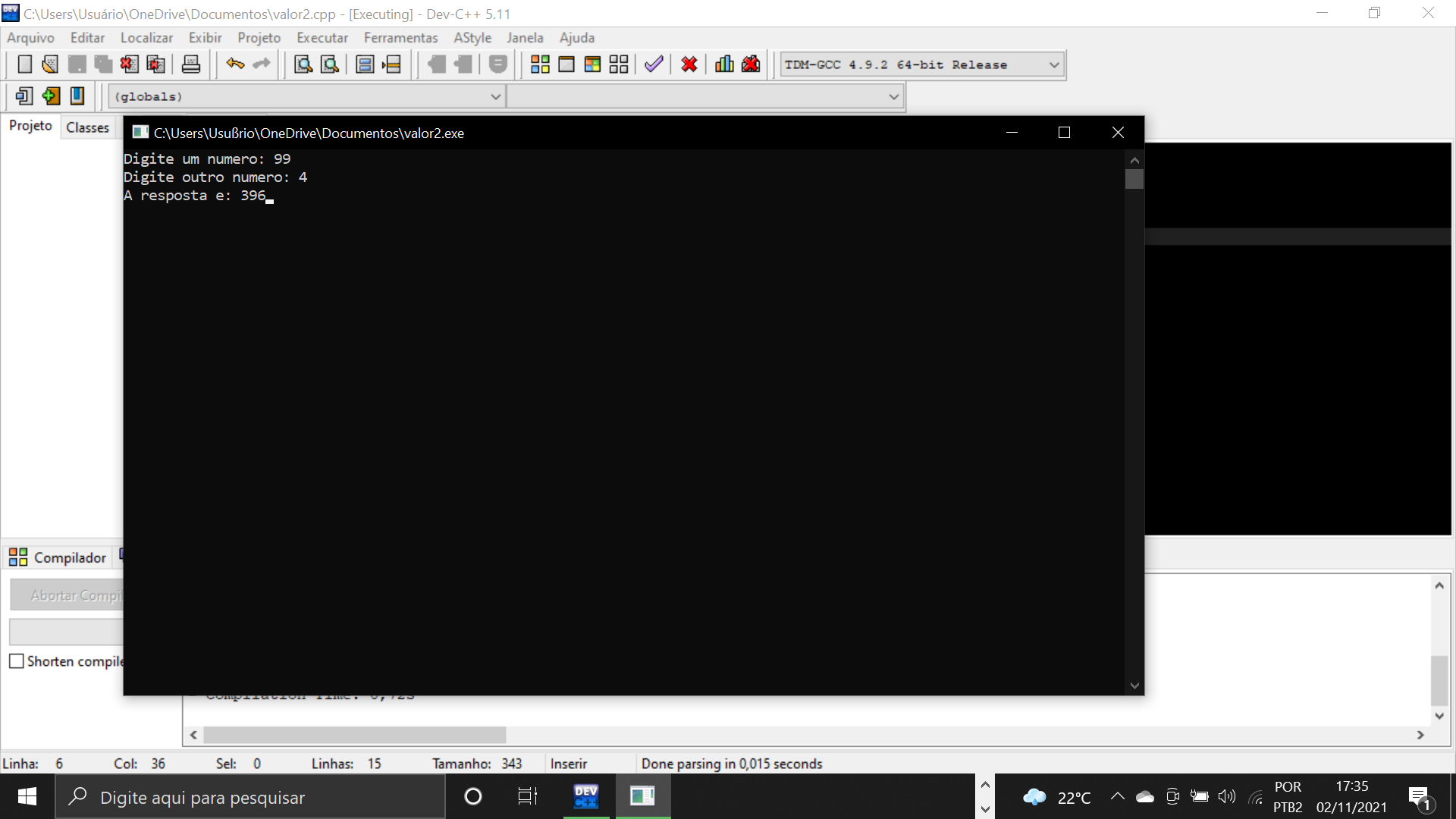
Esta importância também pode estar associada ao apoio realizado pelas Ferramentas CASE às metodologias e métodos que vão surgindo. Sem ferramentas, uma metodologia ou método não terá boa aceitação no mercado, isto ocorreu com diagramas como o DFD (Diagrama de Fluxo de Dados) e o E-R (Entidade — Relacionamento), que só foram amplamente utilizados quando surgiram as primeiras ferramentas para auxiliar na tarefa de diagramação.

**3 PSEUDOCÓDIGO**

**-Estrutura do código e teste**



**-Compilação**



**CONCLUSÃO**

Para concluirmos este Projeto Integrado, abordamos Lógicas Dedutivas, citando exemplos, Ferramentas Case e sua modelagem de dados mais comuns no mercado; Pseudocódigo que é uma maneira genérica de escrever uma linguagem de programação e vimos sobre a especificação de requisitos de software é uma etapa muito importante do desenvolvimento, pois ela pode ser determinante para o sucesso de um software. Nesta fase, é definido os objetivos e funções que um software precisa executar, bem como as que ele não pode ter (restrições) sabendo disso, defina o que são requisitos funcionais e requisitos não- funcionais, dando exemplos para cada um deles.

**REFERÊNCIAs**

<https://jus.com.br/artigos/65069/a-logica-dedutiva-indutiva-e-abdutiva-o-que-isso-tem-a-ver-com-as-novas-questoes-do-direito-e-com-voce#:~:text=A%20l%C3%B3gica%20dedutiva%20busca%20analisar,segunda%20gaveta%20de%20um%20arm%C3%A1rio>.

<https://pt.wikipedia.org/wiki/M%C3%A9todo_dedutivo>

<http://www.each.usp.br/amartins/Aula%203.pdf>

<https://www.devmedia.com.br/ferramentas-case-conhecendo-algumas-boas-opcoes/32034>

<https://datasus.saude.gov.br/glossario/ferramenta-case/#:~:text=Uma%20ferramenta%20CASE%20(do%20ingl%C3%AAs,modelagem%20at%C3%A9%20programa%C3%A7%C3%A3o%20e%20testes>.

-MATERIAIS COMPLEMENTARES UNOPAR.