

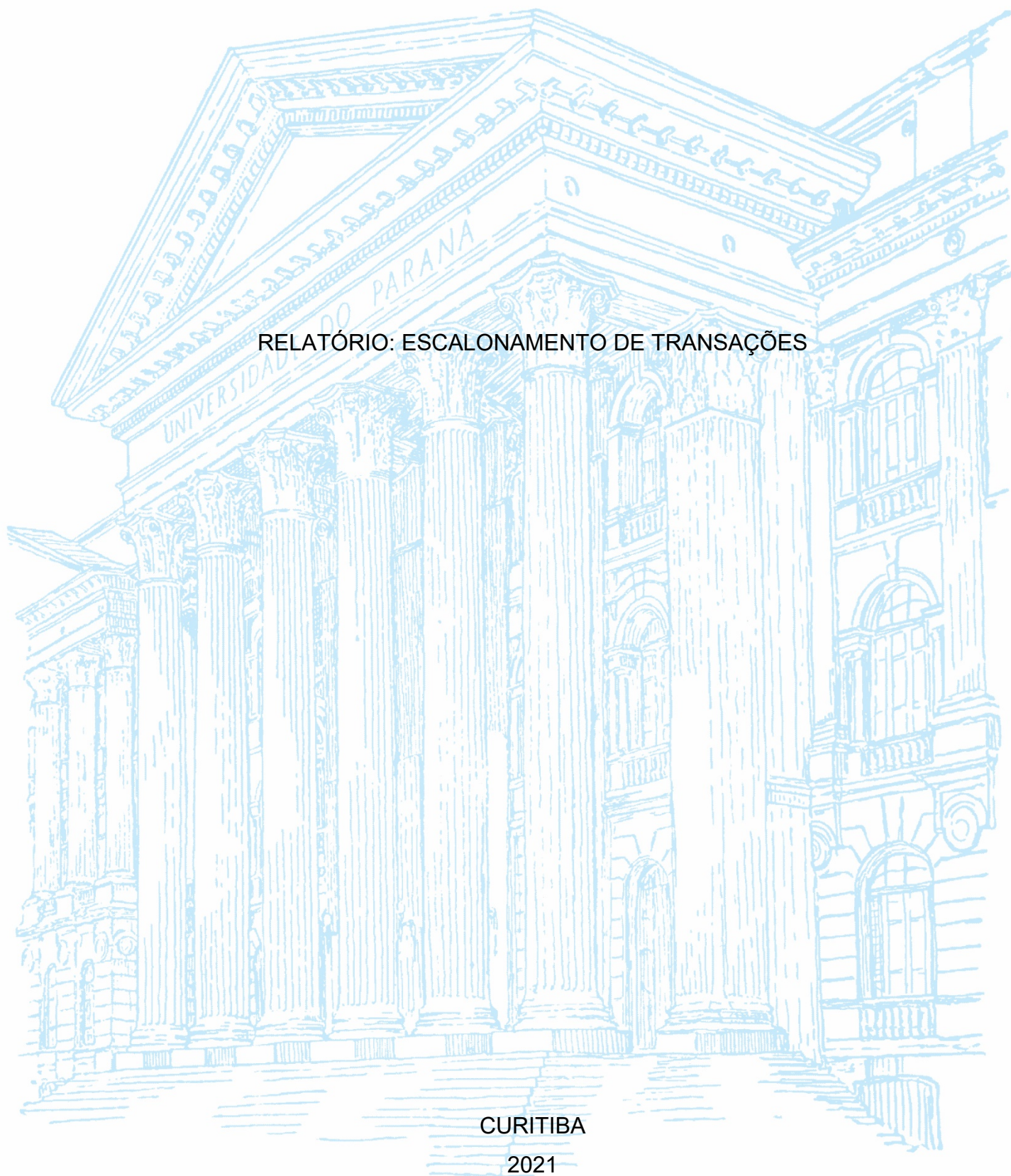
UNIVERSIDADE FEDERAL DO PARANÁ

MATEUS FELIPE DE CÁSSIO FERREIRA

RELATÓRIO: ESCALONAMENTO DE TRANSAÇÕES

CURITIBA

2021



MATEUS FELIPE DE CÁSSIO FERREIRA

RELATÓRIO: IMPACTOS DA BASE DE APRENDIZAGEM

Relatório apresentado como requisito parcial à conclusão da disciplina CI218 – Sistemas de Bancos de Dados, no Curso de Bacharelado em Informática Biomédica, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Orientador: Prof. Dr. Eduardo Almeida

CURITIBA

2021

1 INTRODUÇÃO

Este relatório busca apresentar o sistema que contém dois algoritmos de detecção de conflitos de escalonamento de transações concorrentes, que são:

- a) Serialização de Conflito;
- b) Serialização de Visão;

2 DESENVOLVIMENTO

O sistema implementado utiliza um conjunto de funções e bibliotecas. Os arquivos, com um breve resumo do que representam, são:

- escalona.c: implementa a função principal que trata o escalonamento de transações;
- transacoes.h: define as estruturas de dados que serão utilizadas para montar uma transação, bem como as funções utilizadas para a manipulação das transações contidas no arquivo transacoes.c;
- transacoes.c: implementações da biblioteca transacoes.h;
- grafos.h: define as estruturas de dados que serão utilizadas para montar um grafo, bem como as funções utilizadas para a manipulação dessas estruturas contidas no arquivo grafos.c;
- grafos.c: implementações da biblioteca grafos.h;
- Makefile: utilizado para compilar os arquivos e gerar um programa executável chamado “escalona”.

2.1 SERIALIZAÇÃO DE CONFLITO

Para implementação desse algoritmo, o sistema utiliza um conjunto de funções, procedimentos e uma ordem a ser cumprida. Primeiramente, as transações lidas serão inseridas numa estrutura de grafo, onde ligamos arestas para determinadas condições. Após feito isso, buscamos ciclos formados nesse grafo, percorrendo todos os vértices do grafo gerado e fazendo uma busca de profundidade para ajustar os valores de pré e pós do estado de um vértice em uma função recursiva. Feito isso, todas as arestas do grafo gerado são percorridas, bem como a matriz de adjacência, para buscar uma aresta de retorno, que acontece quando $pos[V_i] < pos[V_j]$. Uma vez determinado os ciclos formados, são identificadas

as transações que conflitantes (e, portanto, não são serializáveis) e as transações que não possuem nenhum conflito (e são, portanto, serializáveis).

Caso o grafo gerado não possua ciclos, será impresso um escalonamento serializável por conflito e por visão. Em caso de o grafo possuir algum ciclo, primeiro serão impressos, no primeiro escalonamento, as transações que tiveram um conflito. Em sequência, o escalonamento não serializável por conflito será testado quanto à equivalência por visão. Ao final, se existir, será impresso um escalonamento das transações que não formaram um ciclo e são, portanto, serializáveis.

2.2 SERIALIZAÇÃO DE VISÃO

O algoritmo de serialização de visão é aplicado a um escalonamento em que apresentou um conflito (formou um ciclo no grafo gerado). Primeiramente, buscamos no conjunto de transações alguma que tenha feito uma operação de escrita. Se encontrado, determinamos se houve uma operação de leitura, em uma mesma transação e em um mesmo atributo, antes da operação dessa operação de escrita. Caso isso **não tenha sido** realizado essa leitura, testamos o escalonamento por visão.

A segunda parte do algoritmo de serialização de visão consiste em identificar todos os atributos que são operados nas transações. A ideia é verificar se existe alguma operação de escrita sobre determinado atributo e armazenar as informações de identificador da transação e o tempo de chegada de determinada transação. Ao final, iteramos sobre uma lista de atributos que foi gerado e tentamos identificar duas transações diferentes (portanto, com identificadores diferentes) que operem uma escrita sobre um atributo e que não seja a última escrita de determinada transação. Caso encontremos alguma transação que atende essa condição, identificamos que o escalonamento não é equivalente por visão. Caso contrário, o escalonamento é equivalente por visão.

Por outro lado, antes de entrar na segunda parte do algoritmo, caso identifiquemos alguma transação que tenha realizado uma operação de leitura (antes da operação de escrita encontrada) em um mesmo atributo, o escalonamento não é equivalente por visão.