

# **Universidade da Beira Interior**

## **Departamento de Informática**



**Departamento de  
Informática**

### **Controlo de Transações**

Elaborado por:

**David Pires M11161**

**Pedro Lemos  
Xavier Campos**

Professor:

**Professor Doutor João Muranho**

8 de novembro de 2020



# Conteúdo

<b>Conteúdo</b>	<b>1</b>
<b>1 Introdução</b>	<b>3</b>
1.1 Introdução . . . . .	3
1.2 Objetivos Gerais . . . . .	3
1.3 Descrição do trabalho prático . . . . .	3
1.4 Objetivos . . . . .	3
1.4.1 Edit . . . . .	4
1.4.2 Browser . . . . .	4
1.4.3 Log Tempo . . . . .	4
1.4.4 Log . . . . .	4
1.5 Organização do Documento . . . . .	4
<b>2 Implementação e Testes</b>	<b>5</b>
2.1 Introdução . . . . .	5
2.2 Transações . . . . .	5
2.2.1 Gestão . . . . .	5
2.2.2 Controlo de Concorrência . . . . .	5
2.2.3 Níveis de isolamento . . . . .	5
2.2.4 Recuperação de Falhas . . . . .	6
2.2.4.1 Tipos de Falhas . . . . .	7
2.2.4.2 Ações Básicas da Recuperação . . . . .	7
2.2.4.3 <i>Checkpoints</i> . . . . .	7
2.2.4.4 Técnicas de Recuperação . . . . .	7
2.3 Tipos de trincos . . . . .	8
2.4 Implementação . . . . .	9
2.4.1 Login . . . . .	9
2.4.2 Edit . . . . .	10
2.4.3 <i>Browser</i> . . . . .	10
2.4.4 Log . . . . .	11
2.4.5 Log Temp . . . . .	12
2.5 Testes e análise de resultados . . . . .	13
2.5.1 Edit . . . . .	13

2.5.1.1	Editar encomenda não confirmada ( <i>uncommitted</i> ) . . . . .	13
2.5.1.2	Dois ou mais utilizadores a editar a mesma encomenda . . . . .	14
2.5.2	<i>Browser</i> . . . . .	14
<b>3</b>	<b>Conclusões</b>	<b>17</b>
3.1	Conclusões Principais . . . . .	17
3.2	Sugestões de melhoria . . . . .	17
3.3	Sugestões de melhoria (para o próximo ano letivo) . . . . .	17
	<b>Bibliografia</b>	<b>19</b>

## **Capítulo**

# **1**

## **Introdução**

### **1.1 Introdução**

Este projeto foi realizado no contexto da unidade curricular de Sistemas de Gestão de Bases de Dados, que se enquadra no primeiro ano de Mestrado em Engenharia Informática da Universidade da Beira Interior, no ano letivo 2020/2021.

### **1.2 Objetivos Gerais**

Para atingirmos um produto final sobre o controlo de transacções, dividiu-se o trabalho por objectivos. Fazer a ligação da Base de Dados à aplicação e desenhar e colocar em ação as quatro aplicações pedidas pelo docente.

### **1.3 Descrição do trabalho prático**

O trabalho prático assenta sobre quatro aplicações que manipulam o sistema de gerenciamento da base de dados. Estas mesmas, têm a competência de controlar transacções, nomeadamente, ao nível do isolamento, tipos de trincos e impasse.

### **1.4 Objetivos**

Nesta secção vamos explicar quais os objectivos de que cada aplicação.

### 1.4.1 Edit

Nesta aplicação pretendemos alcançar a função que nos dá a liberdade para editar tuplos presentes na tabela. Esta aplicação deixa-nos poder alterar a morada do cliente e/ou a quantidade de cada produto que este pediu. Assim que esta esteja de acordo com o utilizador, tem um botão presente para guardar as suas alterações feitas.

### 1.4.2 Browser

Nesta aplicação temos como objectivo termos o poder para consultar e visualizar as encomendas e as suas linhas. Neste caso é apresentado ao utilizador uma janela com duas grelhas, uma onde está presente as encomendas e outra com os produtos de uma encomenda seleccionada.

### 1.4.3 Log Tempo

Com o módulo "Log Tempo" pretende-se mostrar registos do tipo "Outro" numa grelha. Em que cada linha terá a informação da edição de uma encomenda por um certo utilizador.

### 1.4.4 Log

A aplicação terá também uma grelha onde se poderão ser encontrados os  $N$  registos mais recentes feitos na tabela *LogOperations* do tipo "I", "U" e "D".

## 1.5 Organização do Documento

De forma a melhorar a estrutura deste documento, este encontra-se dividido em quatro secções principais:

1. Introdução - nesta secção será feita uma descrição geral do projeto, identificando os objetivos e a descrição do trabalho prático;
2. Implementação e testes - neste capítulo é descrita a solução implementada, bem como alguns testes realizados;
3. Conclusões - Revisão do documento e análise do trabalho realizado. São comparados os objetivos iniciais com os atingidos.

## **Capítulo**

# 2

## ***Implementação e Testes***

### **2.1 Introdução**

Neste capítulo vão ser apresentadas todas as técnicas desenvolvidas, bem como todos os passos que serviram de auxílio ao desenvolvimento do programa final. Serão também apresentados os testes realizados e algumas conclusões.

### **2.2 Transações**

#### **2.2.1 Gestão**

Num Sistema de Gestão de Bases de Dados as transações são executadas de forma intercalada, o que é essencial para o bom desempenho do sistema.

#### **2.2.2 Controlo de Concorrência**

Se a base de dados for acedida por vários utilizadores, é possível que exista transações que vão ler e/ou escrever sobre o mesmo objecto, para tal é preciso haver um controlo sobre estas. A execução concorrente de transações submetidas por vários utilizadores tem de ser organizada de tal modo que cada transacção não interfira com as restantes, pois só assim se pode garantir que não há resultados incorrectos.

#### **2.2.3 Níveis de isolamento**

O padrão SQL define quatro níveis de isolamento de transação em termos de três fenómenos que devem ser evitados entre transações simultâneas. Os

fenómenos não desejados são:

**Dirty Read:** ocorre quando uma transação lê dados que ainda não foram confirmados.

**Nonrepeatable Read:** ocorre quando uma transação lê a mesma linha duas vezes, mas obtém dados diferentes a cada vez.

**Phantom Read:** É uma linha que corresponde aos critérios de pesquisa, mas que não é visto inicialmente e foi adicionada por outra transação.

Nível de Isolamento	<i>Dirty read</i>	<i>Nonrepeatable read</i>	<i>Phantom read</i>
<i>Read Uncommitted</i>	Permitidas	Permitidas	Permitidas
<i>Read Committed</i>	-	Permitidas	Permitidas
<i>Repeatable Read</i>	-	-	Permitidas
<i>Serializable</i>	-	-	-

Tabela 2.1: Níveis de isolamento da transação no SQL [2]

### ***Read Committed***

Não permite que seja recebida informação para a qual ainda não tenha sido executado o *commit*. Este tipo de isolamento adquire *share locks* para todas as páginas onde passará durante a transação.

### ***Read Uncommitted***

É o nível de isolamento mais baixo. Neste nível, uma transação pode ler alterações ainda não *committed* por outra transação, permitindo assim *dirty reads*. Neste nível, as transações não são isoladas umas das outras.

### ***Repeatable Read***

Assegura que *T* apenas lê as alterações realizadas por transações *committed* e nenhum valor escrito ou lido por *T* é alterado por nenhuma outra transação até que este termine.

### ***Serializable***

Este é o nível mais isolado que não permite nenhum tipo de problema (*Dirty Read*, *Nonrepeatable read* e *Phantom Read*).

## **2.2.4 Recuperação de Falhas**

A recuperação de falhas visa recuperar a base de dados para o último estado consistente antes de uma falha. A ideia principal é garantir tanto atomicidade



quanto a durabilidade das transações e cabe ao *recovery manager* do sistema de gestão de base de dados essa responsabilidade.

#### 2.2.4.1 Tipos de Falhas

- Falha de transação
- Falha de sistema
- Falha de meio de armazenamento

#### 2.2.4.2 Ações Básicas da Recuperação

##### **Undo**

Existe quando uma ou mais transações não concluíram todas as suas operações e então as modificações realizadas serão desfeitas.

##### **Redo**

Na ocorrência de uma falha, algumas transações podem ter concluído as suas operações, mas suas ações podem não se ter refletido na base de dados então as modificações realizadas por estas transações são refeitas na base de dados.

#### 2.2.4.3 Checkpoints

Um *checkpoint* é usado para declarar um ponto onde o sistema de gestão de base de dados estava num estado consistente e todas as transações foram committed, sendo guardado num ficheiro *log*.

#### 2.2.4.4 Técnicas de Recuperação

As técnicas mais comuns de recuperação são as baseadas em *logs*. Utiliza um arquivo de *log* onde regista sequencialmente as modificações feitas pelas transações à base de dados. Em caso de falhas, este *log* permite ao *recovery manager* desfazer todas as transações que não tenham sido concluídas, bem como refazer todas as transações *committed*.

#### **ARIES**

*Algorithm for Recovery and Isolation Exploiting Semantics* (ARIES) é baseado no protocolo *Write Ahead Log* (WAL) que em cada modificação será escrita num *log* que poderá ser um dos seguintes:

1. *Undo-only log record*:

Só é guardada os dados iniciais, por isso só a operação *undo* poderá ser feita para obter os dados antigos.

2. *Redo-only log record*:

Só os dados finais são guardados, então só poderá ser utilizada a operação *redo*.

3. *Undo-redo log record*:

Ambos os dados são guardados.

O processo de recuperação é dividido em 3 passos:

1. *Análise*:

Identifica páginas “suja” no *buffer (pool)* e transações ativas no momento da falha;

2. *Redo*:

Repete todas as ações, começando do ponto apropriado no *log* e restaura o estado da base de dados idêntico ao momento da falha;

3. *Undo*:

Desfaz as ações das transações que não realizaram o *commit*, de forma que a base de dados reflita apenas as ações das transações que realizaram o *commit*.

[1]

## 2.3 Tipos de trincos

O *locking* é um procedimento que é usado para controlar o acesso simultâneo a dados quando uma transação está a aceder a base de dados. Um *lock* pode negar acesso a outras transações para prevenir resultados incorrectos. Existe dois tipos de *locks*: *Shared Lock* e *Exclusive Lock*.

1. *Shared Lock*:

Se uma transação tem um *shared lock* num dado, pode ler, mas não o consegue atualizar, pode ser atribuído simultaneamente a múltiplos utilizadores

### 2. *Exclusive Lock*:

Se uma transação tem um *exclusive lock* num dado, pode ler e atualizar. Nem sequer permite que qualquer outros clientes possam visualizar a informação.

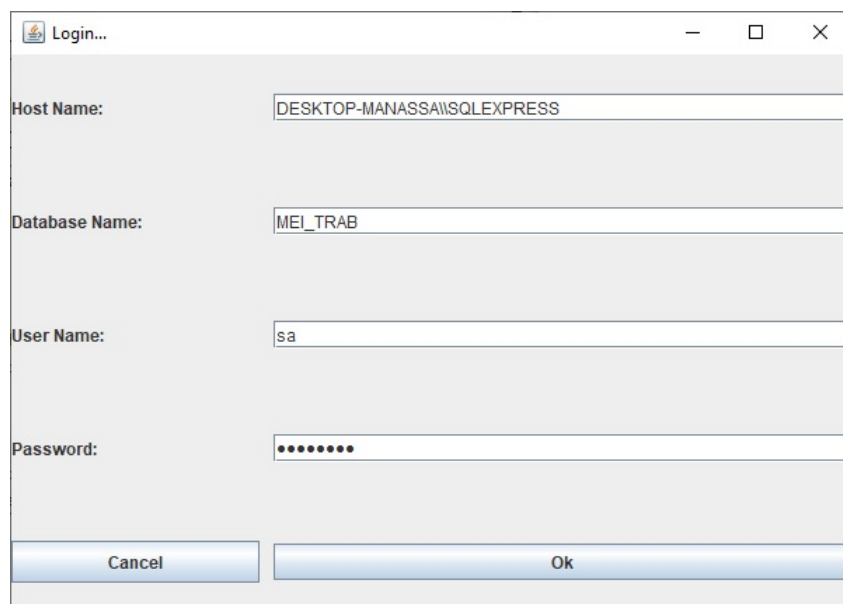
[3]

## 2.4 Implementação

De modo a facilitar a leitura e compreensão do código, este foi separado em vários ficheiros e classes, obtendo uma melhor organização. De seguida será explicado o programa final, bem como deve ser utilizado.

### 2.4.1 Login

Ao iniciar a aplicação, é presente uma janela que dá a possibilidade para conectar ao servidor da base de dados. É solicitado um endereço IP ou *host-name*, o nome da base de dados e as credenciais de um *username*.



The image shows a standard Windows-style login dialog box. The title bar reads 'Login...'. Inside the dialog, there are four labeled text input fields arranged vertically. The first field, 'Host Name:', contains the text 'DESKTOP-MANASSA\SQLEXPRESS'. The second field, 'Database Name:', contains 'MEI\_TRAB'. The third field, 'User Name:', contains 'sa'. The fourth field, 'Password:', contains a series of dots to mask the password. At the bottom of the dialog, there are two buttons: 'Cancel' on the left and 'Ok' on the right.

Figura 2.1: Menu de login.

### 2.4.2 Edit

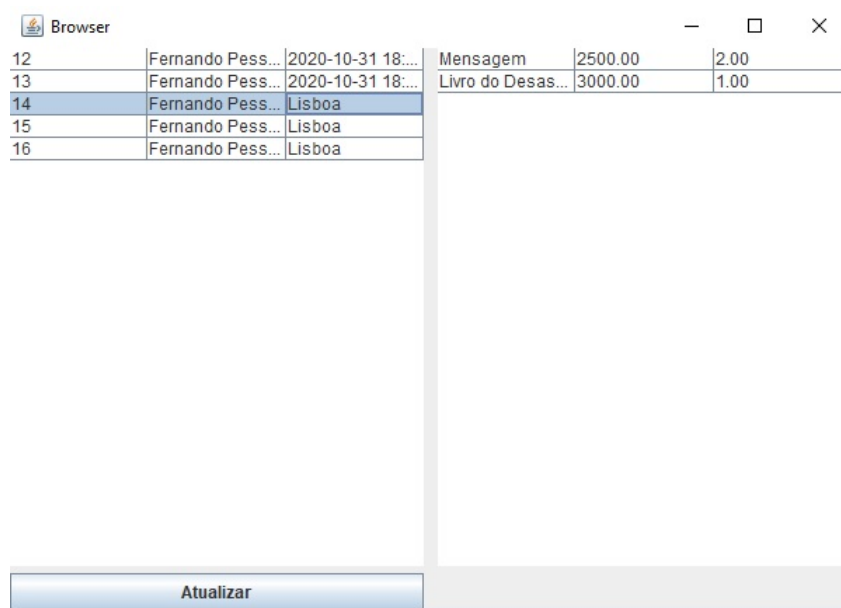
Nesta aplicação, o utilizador indica o ID da encomenda que quer editar/alterar. O programa mostra os dados da encomenda e permite ao utilizador alterar.

Editar encomenda	
Cliente:	Fernando Pessoa
Morada:	Lisboa
Mensagem	2
Livro do Desassossego	1
Produtos:	
<button>Guardar</button>	

Figura 2.2: Aplicação Edit.

### 2.4.3 Browser

Na aplicação *Browser* visualiza as encomendas e as suas linhas. A janela desta aplicação deve conter duas grelhas, uma com as encomendas e a outra com os produtos da encomenda seleccionada.



12	Fernando Pess...	2020-10-31 18:...
13	Fernando Pess...	2020-10-31 18:...
14	Fernando Pess...	Lisboa
15	Fernando Pess...	Lisboa
16	Fernando Pess...	Lisboa

Mensagem	2500.00	2.00
Livro do Desas...	3000.00	1.00

Atualizar

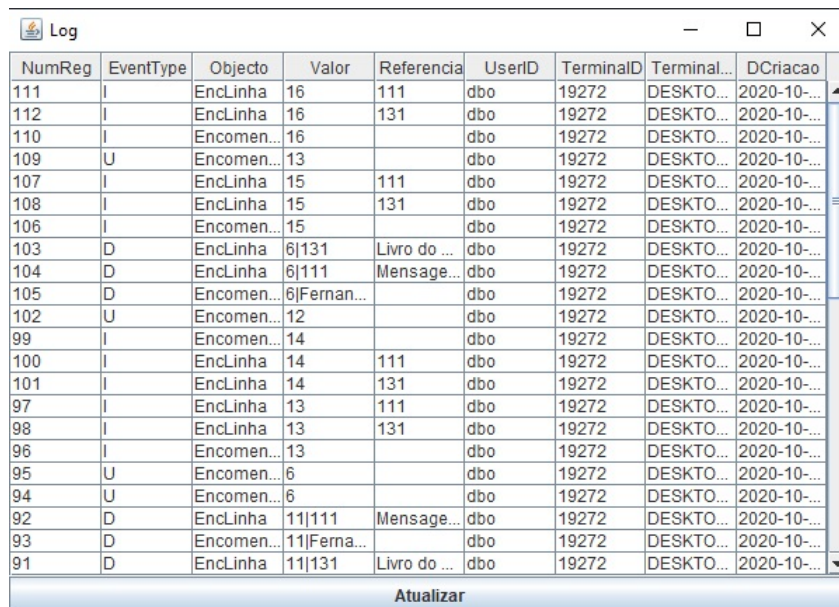
Figura 2.3: Aplicação *Browser*.

#### 2.4.4 Log

Na aplicação *Log* contem uma grelha onde se visualiza as 50 linhas mais recentes da tabela *LogOperations*. Esta tabela só mostra eventos do tipo “I”, “U”, “D” como pedido.

A query executada para a obtenção desta tabela é a seguinte:

```
SELECT TOP 50 * FROM LogOperations WHERE EventType  
= 'I' OR EventType = 'U' OR EventType = 'D'  
ORDER BY DCriacao DESC;
```



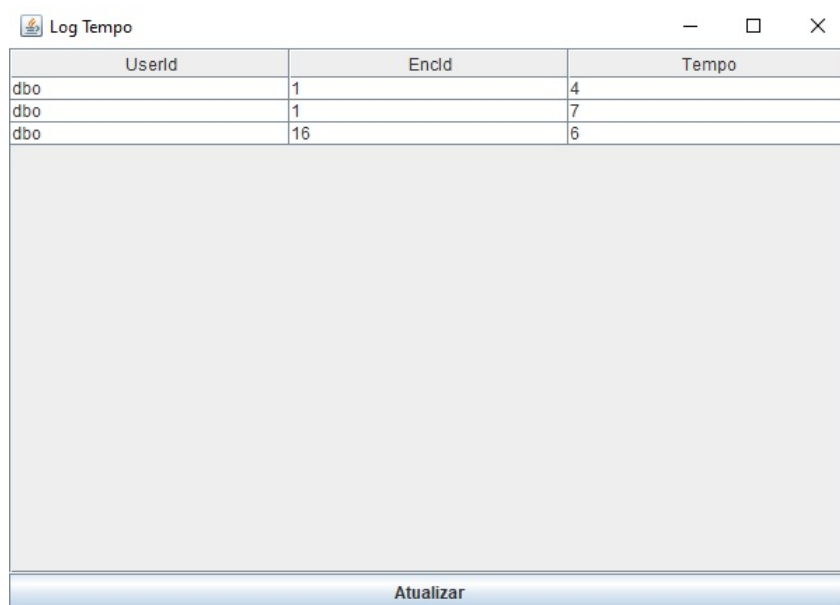
NumReg	EventType	Objecto	Valor	Referencia	UserID	TerminalID	Terminal...	DCriacao
111	I	EncLinha	16	111	dbo	19272	DESKTO...	2020-10-...
112	I	EncLinha	16	131	dbo	19272	DESKTO...	2020-10-...
110	I	Encomen...	16		dbo	19272	DESKTO...	2020-10-...
109	U	Encomen...	13		dbo	19272	DESKTO...	2020-10-...
107	I	EncLinha	15	111	dbo	19272	DESKTO...	2020-10-...
108	I	EncLinha	15	131	dbo	19272	DESKTO...	2020-10-...
106	I	Encomen...	15		dbo	19272	DESKTO...	2020-10-...
103	D	EncLinha	6 131	Livro do ...	dbo	19272	DESKTO...	2020-10-...
104	D	EncLinha	6 111	Mensaje...	dbo	19272	DESKTO...	2020-10-...
105	D	Encomen...	6 Ferna...		dbo	19272	DESKTO...	2020-10-...
102	U	Encomen...	12		dbo	19272	DESKTO...	2020-10-...
99	I	Encomen...	14		dbo	19272	DESKTO...	2020-10-...
100	I	EncLinha	14	111	dbo	19272	DESKTO...	2020-10-...
101	I	EncLinha	14	131	dbo	19272	DESKTO...	2020-10-...
97	I	EncLinha	13	111	dbo	19272	DESKTO...	2020-10-...
98	I	EncLinha	13	131	dbo	19272	DESKTO...	2020-10-...
96	I	Encomen...	13		dbo	19272	DESKTO...	2020-10-...
95	U	Encomen...	6		dbo	19272	DESKTO...	2020-10-...
94	U	Encomen...	6		dbo	19272	DESKTO...	2020-10-...
92	D	EncLinha	11 111	Mensaje...	dbo	19272	DESKTO...	2020-10-...
93	D	Encomen...	11 Ferna...		dbo	19272	DESKTO...	2020-10-...
91	D	EncLinha	11 131	Livro do ...	dbo	19272	DESKTO...	2020-10-...

Atualizar

Figura 2.4: Aplicação Log.

### 2.4.5 Log Temp

Na aplicação *Log Temp* contem uma grelha onde se visualiza quem fez edições do tipo 'O', outro. Tem presente um refresh de 10 em 10 segundos.



Userid	EncId	Tempo
dbo	1	4
dbo	1	7
dbo	16	6

Atualizar

Figura 2.5: Aplicação Log temp.

## 2.5 Testes e análise de resultados

Após elaborada as várias aplicações, como requerido, foram realizados vários testes e analisados os seus resultados. Podemos concluir que os resultados obtidos dependem sempre do nível de transacção definido no momento dos testes. Aqui vamos dar alguns casos práticos e explicar os resultados obtidos.

### 2.5.1 Edit

#### 2.5.1.1 Editar encomenda não confirmada (*uncommitted*)

Para o caso de *Read Uncommitted*:

Query 1

```
begin tran
update encomenda set ClienteId=Pedro where
    EncId=1
waitfor delay '00:00:15'
rollback
```

Query 2

```
set transaction isolation level read
uncommitted
select ClienteId from encomenda where EncId
=1
```

Ao correremos as duas *queries* ao mesmo tempo o *output* será de "Pedro", mas esse não é o output correto porque a *query* 2 leu valores *uncommitted* e como a transação deu *rollback* para o estado inicial, o output correto seria o inicial.

Para os outros níveis de isolamento, não haverá problemas pois nenhum deles utiliza *dirty reads*.

#### 2.5.1.2 Dois ou mais utilizadores a editar a mesma encomenda

- *Read Uncommitted*  
Neste caso os utilizadores, ambos podem editar por isso neste caso, poderá acontecer erros.
- *Read Committed*  
Neste caso os utilizadores conseguem editar sobre as transações que já foram *committed*.
- *Repeatable Read*  
Neste caso os utilizadores terão de esperar que um dos utilizadores acabe a edição.
- *Serializable*  
Neste caso se um dos utilizadores está a fazer uma edição terão de esperar que um dos utilizadores acabe a edição ficando a tela em branco, ou seja fica à espera.

#### 2.5.2 Browser

Aqui vamos considerar diversas taxas de *refresh*, e os resultados obtido com níveis de isolamento diferentes.

- *Read Uncommitted*  
Neste caso os utilizadores, podem observar a tabela sobre transações que ainda não foram *committed*.



- *Read Committed*  
Neste caso os utilizadores conseguem observar a tabela sobre transações que já foram *committed*.
- *Repeatable Read*  
Neste caso se um dos utilizadores está a fazer uma edição, quem tiver a fazer *browse* consegue observar os dados. Trata de um *select*, logo não tem *lock*.
- *Serializable*  
Neste caso se um dos utilizadores está a fazer uma edição, quem tiver a fazer *browse* consegue observar os dados. Trata de um *select*, logo não tem *lock*.



## **Capítulo**

# 3

## **Conclusões**

### **3.1 Conclusões Principais**

Com a realização deste trabalho foi nos possível consolidar os conteúdos lecionados tanto nas aulas assim como ganhar mais experiência a trabalhar com uma base de dado e a sua interacção através de aplicações. Obteve-se conhecimento sobre os níveis de isolamento e o seu impacto na leitura e escrita sobre os dados.

### **3.2 Sugestões de melhoria**

O objetivo do trabalho foi atingido mas poderíamos ter conseguido:

- Melhorar a interface;
- Melhor tratamento dos erros;

### **3.3 Sugestões de melhoria (para o próximo ano letivo)**

O trabalho já é muito completo, mas poderia haver uma aplicação para adicionar e apagar dados.



## ***Bibliografia***

- [1] ARIES. [Online] <https://www.geeksforgeeks.org/algorithm-for-recovery-and-isolation-exploiting-semantics-aries>.
- [2] Nivel de Isolamento. [Online] <https://docs.microsoft.com/pt-pt/sql/connect/jdbc/understanding-isolation-levels?view=sql-server-ver15>.
- [3] Trincos. [Online] <http://www.di.ubi.pt/~hugomcp/bd2/t04.pdf>.