Sistemas de Informação Distribuídos

Licenciaturas em Engenharia Informática e Informática e Gestão de Empresas

2017-2018, Segundo Semestre

Monitorização de Culturas em Laboratório

Auditoria e Migração

Identificação do grupo autor da especificação (Etapa A): \_\_\_7\_\_\_

|  |  |  |
| --- | --- | --- |
| Número | Nome | Foto |
| 73051 | Diogo Juvandes |  |
| 73498 | Alexandre Costa |  |
| 73528 | Miguel Penedo |  |
| 78337 | Marcos Teixeira |  |
| 79038 | Diogo Toupa |  |
| 77983 | Tiago Garção |  |
| X  Especificação: PHP Ficheiro | | |

Identificação do grupo autor da implementação (Etapas B e C): \_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| Número | Nome | Foto |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Especificação: PHP Ficheiro  Implementação: PHP Ficheiro | | |

Instruções

Estas instruções são de cumprimento obrigatório. Relatórios que não cumpram as indicações serão penalizados na nota final.

* Podem (e em várias situações será necessário) ser adicionadas novas páginas ao relatório, mas não podem ser removidas páginas. Se uma secção não for relevante, fica em branco, não pode ser removida;
* Todas as secções têm que iniciar-se no topo de página (colocar uma quebra de página antes);
* A paginação tem de ser sequencial e não ter falhas;
* O índice tem de estar actualizado;
* Na folha de rosto (anterior) têm de constar toda a informação solicitada, nomeadamente todas as fotografias de todos os elementos dos dois grupos. É obrigatório que caiba tudo numa única página;
* A formatação das “zonas” (umas sombreadas outras não sombreadas) não pode ser alterada;
* Nas etapas A e B (até secção 1.4 inclusive), o grupo que primeiro edita o documento (Etapa A) **apenas escreve nas zonas não sombreadas**, e o outro grupo apenas escreve nas zonas sombreadas;
* A etapa C é apenas preenchida pelo grupo que recebe o presente documento do outro grupo. Nas secções 2.1, 2.2, 2.3 e 2.6 deve colocar nas zonas não sombreadas a especificação que entregou ao outro grupo (sem alteração, *copy e paste*),
* As restantes secções são preenchidas normalmente pelo grupo que recebe o presente documento do outro grupo.

Índice

Conteúdo

[1 Etapa A e B 11](#_Toc3143865)

[1.1 Esquema relacional da base de Dados Mysql (origem) 11](#_Toc3143866)

[1.1.1 Apreciação Crítica e esquema relacional implementado 15](#_Toc3143867)

[1.2 Utilizadores Base de Dados de Origem 16](#_Toc3143868)

[1.2.1 Apreciação Crítica a Gestão de Utilizadores Base de Dados de Origem 19](#_Toc3143869)

[1.3 Gestão de Logs 20](#_Toc3143870)

[1.3.1 Triggers de suporte à criação de logs Base de Dados de Origem 20](#_Toc3143871)

[1.3.2 Stored Procedures de suporte à criação de logs (**se relevante**) 26](#_Toc3143872)

[1.4 Migração entre Bases de Dados 31](#_Toc3143873)

[1.4.1 Esquema relacional da base de Dados Mysql (destino) 31](#_Toc3143874)

[1.4.2 Forma de Migração 33](#_Toc3143875)

[1.4.3 Gestão de Utilizadores de Suporte à Migração (origem e/ou destino) 42](#_Toc3143876)

[1.4.4 Triggers de suporte à migração de dados (**se relevante**) 44](#_Toc3143877)

[1.4.5 Stored Procedures de suporte à migração de dados 47](#_Toc3143878)

[1.4.6 Eventos de suporte à migração de dados 50](#_Toc3143879)

[1.4.7 PHP suporte à migração de dados (se relevante) 53](#_Toc3143880)

[1.5 Avaliação Global de especificações da Etapa A 56](#_Toc3143881)

[2 Etapa C (Especificação e Implementação do Próprio Grupo) 58](#_Toc3143882)

[2.1 Especificação do Esquema relacional da base de Dados Origem 58](#_Toc3143883)

[2.2 Especificação de Utilizadores 59](#_Toc3143884)

[2.3 Especificação de Gestão de Logs 60](#_Toc3143885)

[2.3.1 Triggers de suporte à gestão de logs 60](#_Toc3143886)

[2.3.2 Stored Procedures de suporte à gestão de logs 61](#_Toc3143887)

[2.4 Avaliação da especificação do próprio grupo Gestão de Logs 62](#_Toc3143888)

[2.5 Implementação Gestão de Logs 63](#_Toc3143889)

[2.5.1 Utilizadores implementados 63](#_Toc3143890)

[2.5.2 Lista de Triggers 64](#_Toc3143891)

[2.5.3 Triggers Implementados 65](#_Toc3143892)

[2.5.4 Lista de Stored Procedures 66](#_Toc3143893)

[2.5.5 Stored Procedures Implementados 67](#_Toc3143894)

[2.6 Especificação de Migração entre Bases de Dados 68](#_Toc3143895)

[2.6.1 Esquema relacional da base de Dados Mysql especificada (destino) 68](#_Toc3143896)

[2.6.2 Forma de Migração Especificada 69](#_Toc3143897)

[2.6.3 Utilizadores Especificados 70](#_Toc3143898)

[2.6.4 Triggers de suporte à migração de dados especificados 71](#_Toc3143899)

[2.6.5 Stored Procedures de suporte à migração de dados especificados 72](#_Toc3143900)

[2.6.6 Eventos de suporte à migração de dados especificados 73](#_Toc3143901)

[2.6.7 PHP de suporte à migração de dados especificado 74](#_Toc3143902)

[2.7 Avaliação das especificações do próprio grupo Migração 75](#_Toc3143903)

[2.8 Implementação da Migração de Dados 76](#_Toc3143904)

[2.8.1 Utilizadores Implementado 76](#_Toc3143905)

[2.8.2 Lista Triggers 77](#_Toc3143906)

[2.8.3 Triggers Implementados 78](#_Toc3143907)

[2.8.4 Lista de Stored Procedures 79](#_Toc3143908)

[2.8.5 Stored Procedures Implementados 80](#_Toc3143909)

[2.8.6 Lista Eventos 81](#_Toc3143910)

[2.8.7 Eventos Implementados 82](#_Toc3143911)

[2.8.8 PHP Implementado 83](#_Toc3143912)

[2.9 Comparação de Implementações (ficheiro versos PHP) 85](#_Toc3143913)

[2.9.1 Eficiência de Migração 86](#_Toc3143914)

[2.9.2 Robustez 87](#_Toc3143915)

[2.9.3 Flexibilidade / Dependência 88](#_Toc3143916)

[2.9.4 Segurança 89](#_Toc3143917)

[2.10 Auditoria de Dados (base de dados origem) 90](#_Toc3143918)

Monitorização de Culturas em Laboratório

Um laboratório de investigação de um departamento de biologia necessita de um sistema para monitorizar a evolução de culturas. Mais concretamente, pretende acompanhar a temperatura e luz a que as culturas estão sujeitas, bem como detectar/antecipar potenciais problemas.

Numa estufa estão colocados dois sensores que medem a temperatura e quantidade de luz ambiente (que afecta todas as culturas existentes na estufa).

Periodicamente os investigadores dirigem-se à estufa para efectuarem manualmente várias medições de variáveis (humidade, ph, etc) e registá-las num computador que está localizado na estufa. Cada cultura tem um único investigador responsável e apenas ele pode criar, actualizar e consultar os dados de medições das suas culturas. Esta *protecção de dados* é um aspecto importante do sistema. Nem todas as variáveis necessitam serem lidas e registadas. Para cada cultura o investigador decide quais delas devem ser lidas, e regista no sistema qual o intervalo de valores que considera “normal” para o par variável/cultura.

Por exemplo, para as culturas hidropónicas de pimento e tomate, fazem-se medições do nível de concentração de mercúrio e chumbo. Mas numa cultura de bactérias onde se adicionaram antibióticos o que faz sentido medir é o índice de concentração das bactérias, não faz sentido medir o nível de concentração de mercúrio e chumbo.

**Alertas**

Existem dois tipos de alertas:

a) alertas resultantes das medições das variáveis. O investigador, quando insere manualmente um valor de uma medição, caso o valor ultrapasse os limites será alertado com um aviso (no próprio computador) e com uma mensagem para o telemóvel (por vezes o investigador pede a um colega para efectuar a medição, sendo por isso aconselhável que o alerta não apareça somente no monitor do computador).

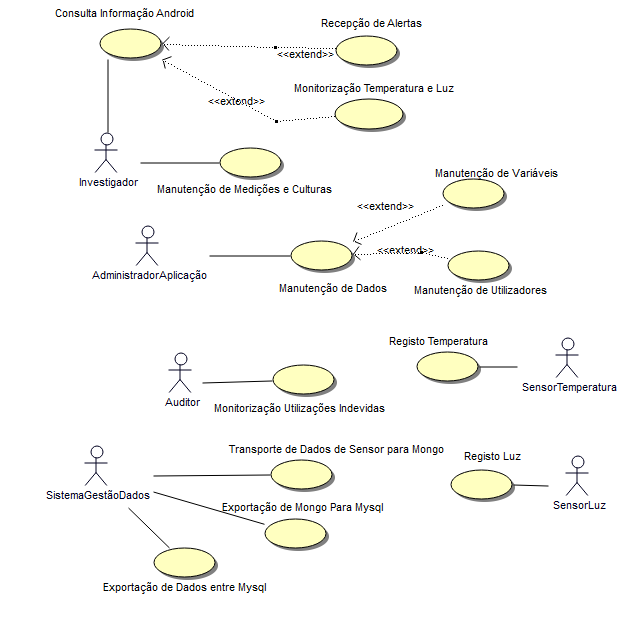
b) Alertas resultantes dos sensores de temperatura e luminosidade. O sistema sabe, para toda a estufa, o intervalo de valores de luminosidade e temperatura adequado (igual para todas as culturas). Se o sensor detectar que os valores vão ser ultrapassados deve notificar por telemóvel o investigador.

Cada investigador deverá ter a possibilidade de, através de um telemóvel, monitorizar a evolução da temperatura e luminosidade (não apenas a última leitura, mas a evolução na última hora ou horas) e receber os dois tipos de alertas.

**Registo de Acessos**

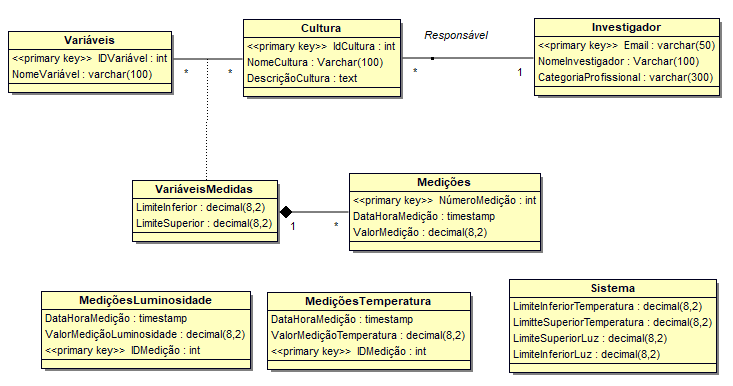
É necessário guardar na base de dados (mysql) o registo de todas as operações de escrita sobre todas as tabelas (quais dados foram alterados/inseridos/apagados, quando e por quem) e o registo de operações de consulta apenas sobre a tabela Medições. Esse registo de alterações (*log) é exportado* incrementalmente(apenas informação nova) e periodicamente para uma base de dados autónoma (também mysql). Através dessa base de dados (apenas de consulta) um auditor pode analisar se ocorreram utilizações abusivas dos dados (por exemplo, quem é que alterou limites de temperatura de uma cultura, etc.).

**Diagrama de Use Case Global**



No presente relatório apenas são contemplados os use case “Exportação Dados entre Mysql”, “Monitorização de Utilizações Indevidas” e “Manutenção de Utilizadores” (apenas a componente Mysql/Privilégios/SP/Triggers)). A componente Java (manutenção de culturas, medições, variáveis e utilizadores) não é especificada neste relatório (diz respeito à UC Eng. Prog II). Nenhum use case pressupõe a programação de formulários.

Diagrama de Classes de Suporte à Base de Dados



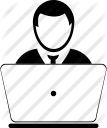
Esquema de Migração

*Auditor de Dados*



Base de Dados

**Mysql**



Exportação Automática, Incremental e periódica

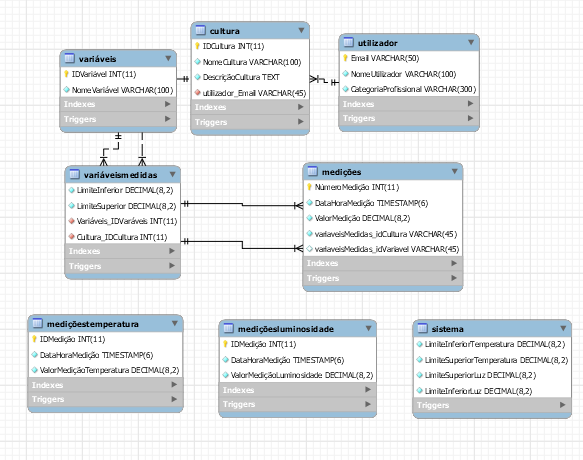
**SQL**

Base de Dados

**Mysql**

# Etapa A e B

## Esquema relacional da base de Dados Mysql (origem)



Todos os atributos são not-null/mandatory (obrigatórios).

Para cada uma das chaves estrangeiras, vamos classificar a relação entre tabelas como cascade ou restrict: cascade - quando é feita uma alteração/delete na variável da tabela mãe as alterações/delete passam para a tabela filha;

restrict - alterações/delete feitas na tabela mãe não passam para a tabela filha).

Definimos a tabela Pai como a que oferece a chave primária e a tabela Filho como a que recebe a chave anterior como chave estrangeira.

No diagrama da Base de Dados Origem, foram identificadas 5 relações possíveis de classificar a sua integridade. Na relação entre as tabelas Utilizador - Cultura, onde Utilizador dá como chave estrangeira “Email”,

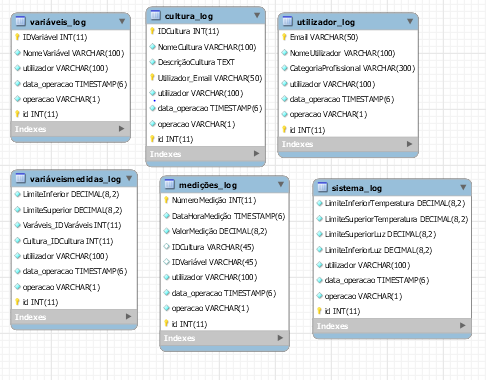
se o valor da chave for alterado, é também alterado na tabela Cultura. Se for eliminado, a tabela Cultura mantém-se, ainda que o utilizador em questão já não conste na base de dados.

Consideramos que todos os Updates das chaves estrangeiras são definidos com Cascade para ser sempre mantida a relação de igualdade entre os valores das tabelas.

No caso do Delete, consideramos que os dados das tabelas Cultura e, por consequência, VariaveisMedidas têm um caráter mais relevante e central à base de dados. Sendo assim, a eliminação de dados de uma destas tabelas deve provocar a eliminação dos mesmos dados nas tabelas descendentes.

Isto não acontece com a eliminação de dados da tabela Variáveis, pois consideramos que mesmo após uma variável ser apagada, deve ser mantido o registo de uma medição associada a uma cultura sobre esta variável.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tabela Pai | Tabela Filho | Chave Estrangeira | Update | Delete |
| Utilizador | Cultura | Email | Cascade | Restrict |
| Variável | VariávelMedida | ID\_Variável | Cascade | Restrict |
| Cultura | VariávelMedida | ID\_Cultura | Cascade | Cascade |
| VariávelMedida | Medição | ID\_Variável | Cascade | Cascade |
| VariávelMedida | Medição | ID\_Cultura | Cascade | Cascade |



As tabelas log são preenchidas por triggers, através das tabelas de dados correspondentes.

A finalidade das tabelas log é registar toda a atividade das tabelas de dados correspondentes, logo, são um duplicado das tabelas originais, mas sem preenchimento obrigatório e com 5 novos campos extra:

ID: Identificador do registo;

Utilizador: Quem realiza a operação;

Operação: Qual a operação realizada (Insert, Update, Delete);

Data: Data em que a operação é realizada.

Todos os registos das tabelas log devem ser mantidos mesmo quando os dados correspondentes nas tabelas de dados são eliminados.

Dá-se a exceção dos logs da tabela Medições. Aqui será necessário guardar a informação de quem consultou esta tabela. Recorrendo a um Stored Procedure, será feita a leitura desta tabela e também o registo de quem executou este SP, guardando no campo `Operação` sobre o valor “S”.

### Apreciação Crítica e esquema relacional implementado

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  Breve Justificação:  Foram feitas alterações? (Sim/Não): \_\_\_\_\_\_\_  **Novo Esquema (assinale e justifique as alterações)**  <Apenas preencher caso tenham procedido a alterações> |

## Utilizadores Base de Dados de Origem

|  |  |  |
| --- | --- | --- |
| **Tabela** | **Administrador** | **Investigador** |
| **Cultura** | - | L,E |
| **Cultura\_Log** | - | L |
| **Medições** | - | L,E |
| **Medições\_Log** | - | L |
| **MediçõesLuminosidade** | L | L |
| **MediçõesTemperatura** | L | L |
| **Sistema** | L,E | L |
| **Sistema\_Log** | L | L |
| **Utilizador** | L,E | L |
| **Utilizador\_Log** | L | L |
| **Variáveis** | L,E | L |
| **Variáveis\_Log** | L | L |
| **VariáveisMedidas** | L,E | L |
| **VariáveisMedidas\_Log** | L | L |

|  |  |  |
| --- | --- | --- |
| **Stored Procedures** | **Administrador** | **Investigador** |
| **select\_all\_medições** | - | X |
| **consulta\_por\_dia** | X | X |
| **consulta\_Email\_por\_CategoriaProfissional** | X | X |
| **consulta\_utilizador\_email\_por\_NomeCultura** | X | X |
| **cria\_utilizador** | X | - |
| **apaga\_utilizador** | X | - |
| **insere\_medição** | - | X |

Em que E=Escrita, L=Leitura, X=Executar e - = sem permissões.

Na base de dados operacional teremos 2 tipos de utilizador: o Administrador e o Investigador.

O Administrador faz a manutenção de variáveis e utilizadores, pelo que tem acesso aos registos de todas as tabelas e permissão de escrita nas tabelas: “Sistema”, “Utilizador”, “Variáveis”, “VariáveisMedidas”.

O Investigador faz a manutenção de culturas e medições, pelo que, também, tem acesso aos registos de todas as tabelas, mas permissão de escrita nas restantes: “Cultura”, “Medições”.

Nenhum dos dois tipos de utilizadores tem permissões de escrita nas tabelas: “MediçõesLuminosidade” e “MediçõesTemperatura” pois o preenchimento desta tabela é feito de forma automática, migrando os valores registados pelos sensores através do MongoDB.

Não são dadas permissões de escrita nas tabelas de logs, pois isto iria comprometer a fiabilidade e legalidade dos registos que poderão ser lidos pelo Auditor

No caso do acesso às tabelas Cultura e Medições, apenas o investigador responsável por uma certa cultura é que pode usufruir das permissões definidas.

A maioria dos Stored Procedures definidos têm função de leitura. São cedidas permissões de leitura a todos os utilizadores para todas as tabelas exceto para a tabela Cultura e Medições, pois apenas investigadores têm acesso a estas.

Os SPs “cria\_utilizador” e “apaga\_utilizador” só podem ser executados por administradores, pois são os responsáveis pela manutenção de utilizadores. O SP “insere\_medição” é apenas executável por Investigadores, pois são os responsáveis pelas medições das suas próprias culturas.

Foi criado uma tabela sistema\_log, pois, apesar de conter dados do sistema, a tabela Sistema tem permissões de escrita dadas ao Administrador, logo, contém dados que estão abertos a serem alterados. Assim, é relevante que exista uma tabela que guarde registos de alterações.

### Apreciação Crítica a Gestão de Utilizadores Base de Dados de Origem

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  **Análise crítica (clareza, completude, rigor):**    **Solução Implementada:** |

## Gestão de Logs

### Triggers de suporte à criação de logs Base de Dados de Origem

Nesta fase o grupo criou triggers after insert, after update e after delete para todas as tabelas cuja alteração fosse necessária registar em tabelas de logs.

As tabelas escolhidas foram a da cultura, medições, sistema, utilizador, variáveis e variáveismedidas. A lógica usada para cada uma das tabelas na construção dos triggers é igual para todas. Tomando como exemplo a tabela cultura, o trigger after insert garante que, após a inserção de dados na tabela cultura, será também inserido uma linha na tabela de logs com o devido registo e operação igual a “I” (insert).

Ao realizar-se um update este também é registado na tabela dos logs através do trigger after update, sendo inserido uma nova entrada com os novos valores e operação igual a “U” (update). Caso seja apagado alguma entrada da tabela cultura, o trigger after delete assegurará que na tabela dos logs estará uma nova entrada com a operação igual a “D” (delete).

Para além disto, cada trigger guarda na tabela dos logs a data a que ocorreu a ação, o responsável e um id.

Tendo isto em conta, o grupo não sentiu a necessidade da criação de triggers before visto que não trazia vantagens clara para este modelo.

Exemplo de Trigger After Update (Tabela Utilizador):

CREATE DEFINER=`root`@`localhost` TRIGGER `sql`.`utilizador\_AFTER\_UPDATE` AFTER UPDATE ON `utilizador` FOR EACH ROW

BEGIN

declare new\_id DOUBLE ;

select max(id) + 1 into new\_id from mediçõesluminosidade\_log;

insert into utilizador\_log (Email, NomeUtilizador, CategoriaProfissional, utilizador, data\_operacao, operacao)

values (new.Email, new.NomeUtilizador, new.CategoriaProfissional, current\_user, now(), "U");

if(old.Email<>new.Email) or (old.Email is Null and new.Email is NOT NULL) THEN

update utilizador\_log set Email = new.Email where id = new\_id;

end if;

if(old.NomeUtilizador<>new.NomeUtilizador) or (old.NomeUtilizador is Null and new.NomeUtilizador is NOT NULL) THEN

update utilizador\_log set NomeUtilizador= new.NomeUtilizador where id = new\_id;

end if;

if(old.CategoriaProfissional<>new.CategoriaProfissional) or (old.CategoriaProfissional is Null and new.CategoriaProfissional is NOT NULL) THEN

update utilizador\_log set CategoriaProfissional = new.CategoriaProfissional where id = new\_id;

end if;

END

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Trigger | Tabela | Tipo Operação | Evento |
| Cultura\_AFTER\_INSERT | Cultura | I | A |
| Cultura\_AFTER\_DELETE | Cultura | D | A |
| Cultura\_AFTER\_UPDATE | Cultura | U | A |
| medições\_AFTER\_INSERT | medições | I | A |
| medições\_AFTER\_DELETE | medições | D | A |
| medições\_AFTER\_UPDATE | medições | U | A |
| Sistema\_AFTER\_INSERT | Sistema | I | A |
| Sistema\_AFTER\_DELETE | Sistema | D | A |
| Sistema\_AFTER\_UPDATE | Sistema | U | A |
| Utilizador\_AFTER\_INSERT | Utilizador | I | A |
| Utilizador\_AFTER\_DELETE | Utilizador | D | A |
| Utilizador\_AFTER\_UPDATE | Utilizador | U | A |
| Variaveis\_AFTER\_INSERT | Variáveis | I | A |
| Variaveis\_AFTER\_DELETE | Variáveis | D | A |
| Variaveis\_AFTER\_UPDATE | Variáveis | U | A |
| Variaveis\_AFTER\_INSERT | Variáveis Medidas | I | A |
| Variaveis\_AFTER\_DELETE | Variáveis Medidas | D | A |
| Variaveis\_AFTER\_UPDATE | Variáveis Medidas | U | A |

#### Apreciação Crítica de triggers para gestão de logs

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  Breve Justificação:  **Lista de Triggers (para cada trigger assinalar com x em célula correspondente)**     |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Implementado de Acordo com Especificado | Implementado mas diferente de Especificado | Não Implementado | Não Especificado (criado de novo) | | Nome Trigger  (tal como especificado) |  |  |  |  | | Nome Trigger  (tal como especificado) |  |  |  |  | | Nome Trigger  (tal como especificado) |  |  |  |  | |  |  |  |  |  | |

#### Triggers Implementados para gestão de logs

|  |
| --- |
| 1. Nome Trigger: \_\_\_\_\_  //*Breve Descrição*  *Código*  2. Nome Trigger: \_\_\_\_\_  //*Breve Descrição*  *Código*  3. Nome Trigger: \_\_\_\_\_  //*Breve Descrição*  *Código* |

### Stored Procedures de suporte à criação de logs (**se relevante**)

Nesta fase, foram criados Stored Procedures de modo a poder ser possível visualizar nas tabelas de logs quem fez um select, funcionando por isso como um after select.

Foi realizada a criação de SPs que permitem o select de toda a tabela e o devido registo na tabela de log respetiva com a data da operação, o utilizador que realizou o select, id e operação igual a ‘S’.

Em outros casos foram dados parâmetros de entrada, para que haja uma maior filtragem das linhas da tabela e uma pesquisa mais objetiva, resultando em parâmetros de saída mais específicos e que ajudam o utilizador a selecionar melhor a informação que deseja ver.

Exemplo de SP(select\_Email\_by\_CategoriaProfissional):

CREATE PROCEDURE `select\_Email\_by\_CategoriaProfissional`(in categp varchar(50), out mail varchar(100))

BEGIN

INSERT INTO utilizador\_log (Email, NomeUtilizador, CategoriaProfissional, utilizador, data\_operacao, operacao) VALUES

('','','', CURRENT\_USER, now() , 'S');

Select Email into mail

from utilizador

where categp = CategoriaProfissional;

END

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Procedimento | Parâmetros Entrada | Parâmetros Saída | Breve descrição |
| select\_all\_medições | - | - | Seleciona todas as colunas da tabela medições |
| consulta\_por\_dia | DataHoraMedição | - | Seleciona todas as colunas da tabela medições que tenham como data de medição o valor dado |
| consulta\_Email\_por\_CategoriaProfissional | CategoriaProfissional | Email | Seleciona a coluna Email da tabela utilizador devolvendo as linhas que tenham como CategoriaProfissional o valor dado |
| consulta\_utilizador\_email\_por\_NomeCultura | NomeCultura | Investiigador\_Email | Seleciona a coluna Utilizador\_Email da tabela cultura devolvendo as linhas que tenham como NomeCultura o valor dado |
| cria\_utilizador | Email, NomeUtilizador, CategoriaProfissional, TipoUtilizador |  | Recebe como argumento dados, inseridos manualmente, de um utilizador e insere-os na tabela Utilizador |
| apaga\_utilizador | Email |  | Recebe como argumento o Email de um utilizador e apaga os dados do próprio da tabela Utilizador |
| insere\_medição | ValorMedição |  | Recebe como argumento o valor de uma medição inserida manualmente e insere-a na tabela Medições |

#### Apreciação Crítica de Stored Procedures de suporte à criação de logs

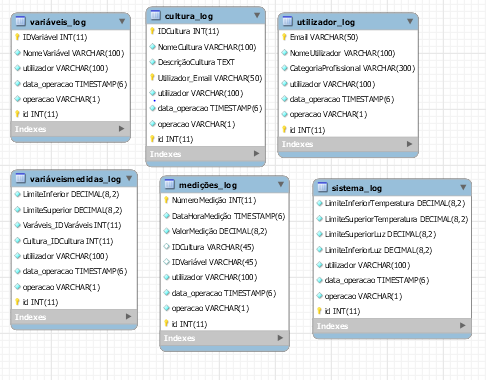
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  Breve Justificação:  **Lista de SP (para cada SP assinalar com x em célula correspondente)**     |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Implementado de Acordo com Especificado | Implementado mas diferente de Especificado | Não Implementado | Não Especificado (criado de novo) | | Nome SP  (tal como especificado) |  |  |  |  | | Nome SP  (tal como especificado) |  |  |  |  | | Nome SP  (tal como especificado) |  |  |  |  | |  |  |  |  |  | |

#### Stored Procedures Implementados de suporte à criação de logs

|  |
| --- |
| 1. Nome SP: \_\_\_\_\_  //*Breve Descrição*  *Código*  2. Nome SP: \_\_\_\_\_  //*Breve Descrição*  *Código*  3. Nome SP: \_\_\_\_\_  //*Breve Descrição*  *Código* |

## Migração entre Bases de Dados

### Esquema relacional da base de Dados Mysql (destino)



#### Apreciação Crítica e esquema relacional implementado

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  Breve Justificação:  Foram feitas alterações? (Sim/Não): \_\_\_\_\_\_\_  **Novo Esquema (assinale e justifique as alterações)**  <Apenas preencher caso tenham procedido a alterações> |

### Forma de Migração

Requisitos para migração:

Para o processo de migração será necessária a instalação do programa Xampp, para o uso dos serviços Apache e MySQL do próprio. Isto criará um servidor de base de dados para o MySQL. Será também utilizado o Task Scheduler do Windows.

Tipo de Ficheiro:

Foi considerada a utilização de dois tipos de ficheiros para este processo: xml ou csv.

As grandes vantagens do CSV sobre o XML é que o primeiro é legível e fácil de editar manualmente, o que por sua vez torna-se mais simples de implementar e analisar que o segundo.

Para além disso, o CSV ocupa menos espaço que o XML, o que o torna mais fácil de gerar e manter. Quanto ao XML, a sintaxe é redundante quando se trata de dados tabulares o que irá afetar a eficiência do processo.

Apesar do XML suportar Unicode (permite qualquer linguagem humana escrita seja comunicada), de poder representar estruturas de dados de ciência da computação (registos, listas e árvores) e de ter requisitos rigorosos de sintaxe, estas características não acrescentariam benefícios em relação ao CSV, tendo em conta as nossas necessidades.

XML contribuiria para uma leitura facilitada a nível do utilizador, mas considerando que será uma máquina a processar o ficheiro e a traduzi-lo para a base de dados, este fator também se torna irrelevante.

Considerando todos os fatos e necessidades, o csv torna-se mais benéfico e eficiente.

Exportação:

A exportação será realizada sobre todas as tabelas de Logs: cultura\_log; medições\_log; sistema\_log; utilizador\_log; variaveis\_log; variaveismedidas\_log.

Sendo assim, existirá um Stored Procedure para cada tabela de logs, responsável pela exportação da própria tabela. Por exemplo: SP exporta\_cultura\_log() exportará apenas a informação na tabela cultura\_log.

Tendo isto em conta, cada SP irá gerar um ficheiro .csv para a tabela a ele atribuída. Por exemplo: SP export\_cultura\_log() gera o ficheiro “n\_cultura\_log.csv”.

Um ficheiro .csv para cada tabela exportada.

No SP, estará declarada uma variável que guarda o último id exportado da tabela respetiva. Assim, apenas será exportada informação com id maior que o valor da variável, evitando envio de entradas previamente exportadas. Isto pode ser representado pelo exemplo de código:

declare ultimoid int default 0;

select \* from cultura\_log where id\_log> @ultimoid

. . .

set @ultimoid := (select max(id\_log) from cultura\_log);

O procedimento começa com a verificação da existência de um id na tabela maior que o último id exportado. Isto para saber se existiu alguma inserção de dados desde a última exportação. Caso não existam dados novos, não há necessidade do procedimento continuar.

No caso de existirem dados novos:

- é selecionada toda a informação com um id maior que o guardado na variável;

- os dados selecionados são exportados para um ficheiro .csv;

- verifica-se o valor do id do último dado exportado e guarda-se este valor na variável.

O nome de cada ficheiro será criado de forma dinâmica - “n\_NomeTabela.csv” - em que “NomeTabela” representa a tabela de logs que está a ser exportada e “n” representa o id do ficheiro que será incrementado automaticamente. A importância deste id será explicada no ponto Importação.

A geração do nome pode ser feita da seguinte forma:

declare numero int;

set @filename= '\_CulturaLog.csv"';

set @comando = 'select \* from `cultura\_log` into outfile "';

set @numero := @numero+1;

SET @statement =

CONCAT(@comando, @numero, @filename);

select @statement;

PREPARE s1 FROM @statement;

EXECUTE s1;

DROP PREPARE s1;

Este código consiste na criação de duas variáveis de texto - uma com o comando de exportação, outra com o nome do ficheiro - e outra variável INT, que guarda o numero associado ao ficheiro. A função CONCAT junta todas estas variveis numa string. É necessária a utilização de Prepared Statements para executar o que estiver presente na string anteriormente criada.

É pretendido que todos estes passos sejam executados de forma periódica e automática. Logo, é necessário a criação de um evento MySQL. Aqui é declarado o período de tempo entre cada execução e é chamada a execução de cada SP. Foi decidido que um período de 24h é o mais indicado, tendo em conta que, desta forma, estaremos a exportar os dados recolhidos de cada dia de trabalho.

O evento pode ser criado com o seguinte exemplo de código:

create event export

on schedule every 24 hour

do

call\_sp()

Importação:

Esta fase da migração será baseada na utilização de um ficheiro .bat e na criação de uma tarefa Windows no Task Scheduler, que auxiliará na execução automática e periódica deste ficheiro .bat. Neste ficheiro existem 3 tipos de comandos para a execução deste processo. Cada tipo de comando terá uma iteração para cada ficheiro importado. A execução seguirá a seguinte lógica (seguindo o exemplo da importação do ficheiro “CulturaLog.csv”):

-O primeiro comando identifica todos os ficheiros com dados da tabela CulturaLog e junta-os num único ficheiro. Neste passo nota-se a importância de cada ficheiro ter um nome dinâmico, como descrito no ponto Exportação (e.g. “3\_CulturaLog.csv”). Para cada exportação, dá-se uma importação. Mas na eventualidade de existir uma falha na importação, o objetivo é preservar o ficheiro não importado, deixando-o disponível para importação na próxima tentativa. Imaginando o caso em que há uma falha a importar o ficheiro “3\_CulturaLog.csv”. A próxima exportação irá gerar o ficheiro “4\_CulturaLog.csv” na mesma diretoria que o anterior. Na próxima tentativa de importação, este comando irá juntar os dois ficheiros previamente mencionados, gerando o ficheiro CulturaLog.csv. Isto pode ser feito com o código exemplo (comando Windows):

c:\directory> copy \*CulturaLog.csv CulturaLog.csv

-O segundo comando executará um script com linguagem MySQL com o simples propósito de importar os dados no ficheiro para a tabela presente na Base de dados Destino respetiva e a formatação necessária aos dados, tendo em conta que é um ficheiro csv. Um exemplo de código útil para esta implementação seria:

LOAD DATA INFILE(“C:\directoria\CulturaLog.csv”)

INTO TABLE cultura\_log

FIELDS TERMINATED BY ','

ENCLOSED BY '\''

LINES TERMINATED BY '\n'

-Terceiro e último comando presente no ficheiro .bat servirá para eliminar os ficheiros já importados, completando assim o processo. Este passo é necessário por várias razões. Se estes não fossem apagados, a informação já importada seria enviada de novo, junta com os novos dados.

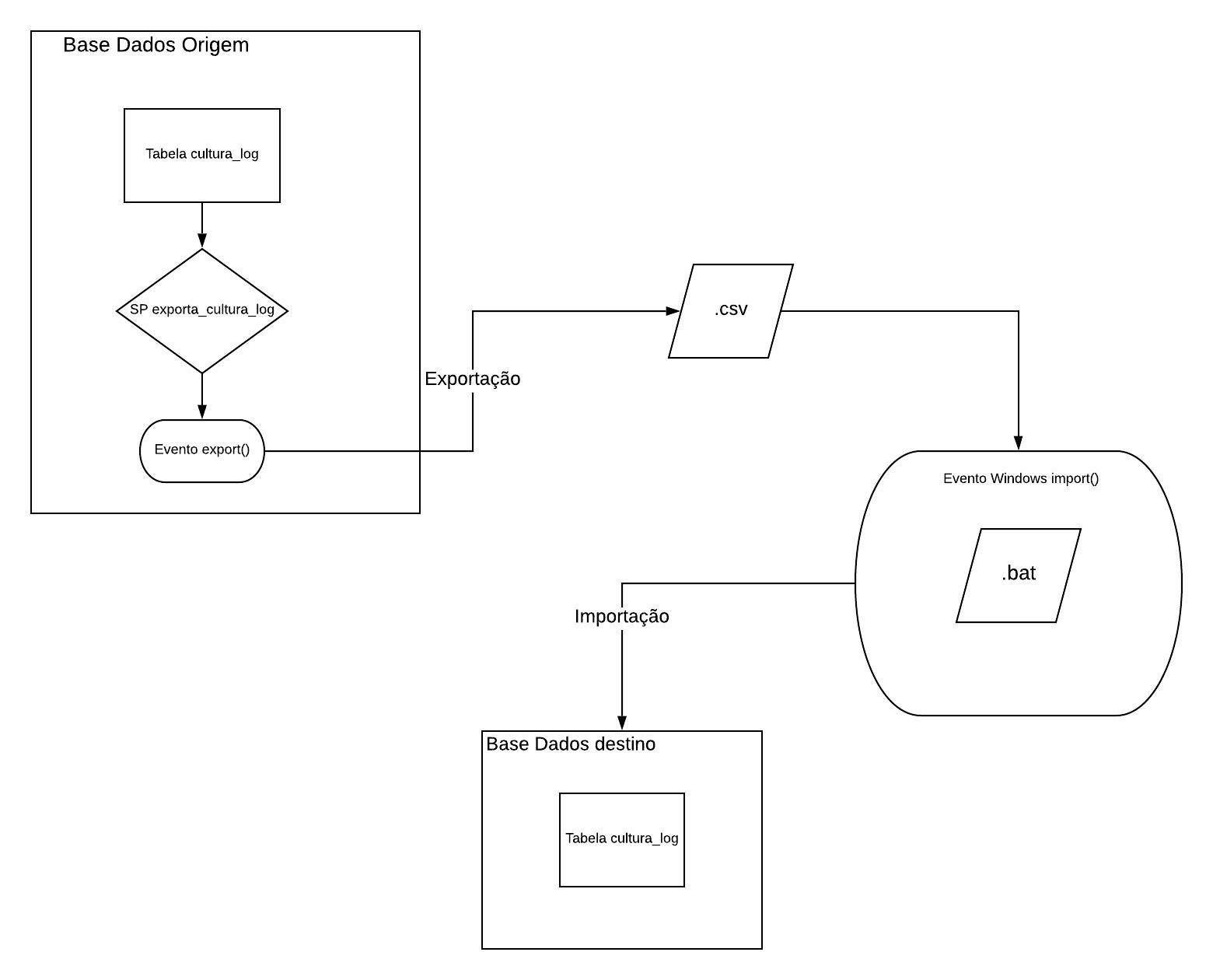
Isto também providencia uma maior segurança de dados, visto que seria perigoso existirem ficheiros com estes dados que poderiam ser acedidos sem qualquer tipo de restrições. Por último, isto também liberta espaço em disco, sendo que seria inútil manter os dados em ficheiro, sabendo que estes mesmos dados já se encontram onde se devem encontrar.

Será criada uma tarefa no Windows Task Sheduler. Esta tarefa executará o ficheiro .bat com a mesma periodicidade de 24h que tem o evento de exportação. De notar que esta tarefa só deve ser executada após o processo de exportação. Também reparar que para cada ficheiro importado, é aplicada a mesma lógica de comandos, como descrito no exemplo em cima.

Reparos gerais:

Todos estes passos garantem um processo automático, com um período bem definido. A eliminação de ficheiros após importação oferece uma maior segurança e privacidade de dados. Este processo é também eficiente, pois garante a chegada de todos os dados à BD destino, sempre enviando apenas dados novos.

Diagrama do processo de Migração (seguindo o exemplo da tabela “cultura\_log”):



#### Apreciação Crítica à especificação da forma de migração

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  **Análise crítica (clareza, completude, rigor):** |

### Gestão de Utilizadores de Suporte à Migração (origem e/ou destino)

|  |  |
| --- | --- |
| Tabela | Auditor |
| Cultura\_Log | L |
| Medições\_Log | L |
| Sistema\_Log | L |
| Utilizador\_Log | L |
| Variáveis\_Log | L |
| VariáveisMedidas\_Log | L |

Em que E=Escrita, L=Leitura, X=Executar e - = sem permissões.

A Gestão de utilizadores na base de dados destino, incide apenas sobre um tipo de utilizador, o Auditor.

O Auditor tem permissões para consultar qualquer registo em todas as tabelas log, de forma a certificar a conformidade dos dados.

#### Apreciação Crítica à especificação da Gestão de Utilizadores

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  **Análise crítica (clareza, completude, rigor):**    **Solução Implementada:** |

### Triggers de suporte à migração de dados (**se relevante**)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nome Trigger | Tabela | Tipo de Operação (I,U,D) | Evento  (A,B) | BD  (Origem ou Destino) | Notas (apenas indicar aquilo que não será óbvio) |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### Apreciação Crítica de triggers de suporte à migração de dados

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  Breve Justificação:  **Lista de Triggers (para cada trigger assinalar com x em célula correspondente)**     |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Implementado de Acordo com Especificado | Implementado mas diferente de Especificado | Não Implementado | Não Especificado (criado de novo) | | Nome Trigger  (tal como especificado) |  |  |  |  | | Nome Trigger  (tal como especificado) |  |  |  |  | | Nome Trigger  (tal como especificado) |  |  |  |  | |  |  |  |  |  | |

#### Triggers Implementados de suporte à migração de dados

|  |
| --- |
| 1. Nome Trigger: \_\_\_\_\_  //*Breve Descrição*  *Código*  2. Nome Trigger: \_\_\_\_\_  //*Breve Descrição*  *Código*  3. Nome Trigger: \_\_\_\_\_  //*Breve Descrição*  *Código* |

### Stored Procedures de suporte à migração de dados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome Procedimento | Parâmetros Entrada | Parâmetros Saída | BD | Descrição |
| exporta\_cultura\_log |  |  | Origem | Exporta dados da tabela `cultura\_log` para o ficheiro respetivo |
| exporta\_utilizador\_log |  |  | Origem | Exporta dados da tabela `utilizador\_log` para o ficheiro respetivo |
| exporta\_variaveis\_log |  |  | Origem | Exporta dados da tabela `variaveis\_log` para o ficheiro respetivo |
| exporta\_variaveismedidas\_log |  |  | Origem | Exporta dados da tabela `variaveismedidas\_log` para o ficheiro respetivo |
| exporta\_medições\_log |  |  | Origem | Exporta dados da tabela `mediçoes\_log` para o ficheiro respetivo |
| exporta\_sistema\_log |  |  | Origem | Exporta dados da tabela `sistema\_log` para o ficheiro respetivo |

#### Apreciação Crítica de Stored Procedures de suporte à migração de dados

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  Breve Justificação:  **Lista de SP (para cada SP assinalar com x em célula correspondente)**     |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Implementado de Acordo com Especificado | Implementado mas diferente de Especificado | Não Implementado | Não Especificado (criado de novo) | | Nome SP  (tal como especificado) |  |  |  |  | | Nome SP  (tal como especificado) |  |  |  |  | | Nome SP  (tal como especificado) |  |  |  |  | |  |  |  |  |  | |

#### Storedd Procedures Implementados de suporte à migração de dados

|  |
| --- |
| 1. Nome SP: \_\_\_\_\_  //*Breve Descrição*  *Código*  2. Nome SP: \_\_\_\_\_  //*Breve Descrição*  *Código*  3. Nome SP: \_\_\_\_\_  //*Breve Descrição*  *Código* |

### Eventos de suporte à migração de dados

|  |  |  |
| --- | --- | --- |
| Nome Evento | Local Execução | Descrição |
| exporta\_geral | Origem | Executa todos os Stored Procedures responsáveis pela exportação de cada tabla |
| importa\_geral | Sistema Operativo | Executa o ficheiro .bat responsável pela importação |

#### Apreciação Crítica de Eventos

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  Breve Justificação:  **Lista de Eventos (para cada evento assinalar com x em célula correspondente)**     |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Implementado de Acordo com Especificado | Implementado mas diferente de Especificado | Não Implementado | Não Especificado (criado de novo) | | Nome Evento  (tal como especificado) |  |  |  |  | | Nome Evento  (tal como especificado) |  |  |  |  | | Nome Evento  (tal como especificado) |  |  |  |  | |  |  |  |  |  | |

#### Eventos Implementados

|  |
| --- |
| 1. Nome Evento: \_\_\_\_\_  //*Breve Descrição*  *Código*  2. Nome Evento: \_\_\_\_\_  //*Breve Descrição*  *Código*  3. Nome Evento: \_\_\_\_\_  //*Breve Descrição*  *Código* |

### PHP suporte à migração de dados (se relevante)

<Nesta secção deverá especificar a lógica subjacente ao programa PHP de suporte à migração>

#### Apreciação Crítica ao PHP especificado

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  Breve Justificação: |

#### PHP Implementado

|  |
| --- |
| *Código* |

## Avaliação Global de especificações da Etapa A

<Texto avaliativo da qualidade e clareza das especificações recebidas. Referir a coerência, completude, nível de rigor e detalhe. Convém exemplificar afirmações>

**Avaliação Global da Qualidade das Especificações recebidas**

|  |
| --- |
| Avaliação (A,B,C,D,E) : \_\_\_\_\_\_\_\_\_\_\_\_\_  Utilize a seguinte escala:  A: - 1 – 5 valores B: 6 – 9 valores C: 10 – 13 Valores D: 14 – 17 valores E: 18 – 20 valores |

**Três principais deficiências de especificação que tiveram impacto mais negativo na qualidade da implementação**

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Resumo de Avaliações de Qualidade Anteriores (para cada linha assinalar com x em célula correspondente)**   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Fraco | Razoável | Bom | Muito Bom | | BD Origem |  |  |  |  | | Triggers Log |  |  |  |  | | SP Log |  |  |  |  | | Utilizadores Log |  |  |  |  | | BD Destino |  |  |  |  | | Forma Migração |  |  |  |  | | Triggers Migração |  |  |  |  | | SP Migração |  |  |  |  | | Eventos Migração |  |  |  |  | | Utilizadores Migração |  |  |  |  | | PHP Migração |  |  |  |  | |

# Etapa C (Especificação e Implementação do Próprio Grupo)

## Especificação do Esquema relacional da base de Dados Origem

## Especificação de Utilizadores

## Especificação de Gestão de Logs

### Triggers de suporte à gestão de logs

### Stored Procedures de suporte à gestão de logs

## Avaliação da especificação do próprio grupo Gestão de Logs

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  Justificação:  <fazer um resumo dos principais pontos fracos e fortes.  Depois de ler esta secção o leitor deve ter uma visão sobre que secções estavam mais fracas (triggers? Base de dados?)> |

## Implementação Gestão de Logs

### Utilizadores implementados

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **Tabela** | **Tipo de Utilizador** | | | | Tipo 1 | Tipo 2 | … | | T1 | E | - |  | | T2 | L | E |  | | … |  |  |  | | **Stored Proc.** |  |  |  | | SP1 | X | - |  | | … |  |  |  | |

### Lista de Triggers

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Lista de Triggers (para cada trigger assinalar com x em célula correspondente)**     |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Implementado de Acordo com Especificado | Implementado mas diferente de Especificado | Não Implementado | Não Especificado (criado de novo) | | Nome Trigger  (tal como especificado) |  |  |  |  | | Nome Trigger  (tal como especificado) |  |  |  |  | | Nome Trigger  (tal como especificado) |  |  |  |  | |  |  |  |  |  | |

### Triggers Implementados

|  |
| --- |
| 1. Nome Trigger: \_\_\_\_\_  //*Breve Descrição*  *Código*  2. Nome Trigger: \_\_\_\_\_  //*Breve Descrição*  *Código*  3. Nome Trigger: \_\_\_\_\_  //*Breve Descrição*  *Código* |

### Lista de Stored Procedures

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Lista de SP (para cada SP assinalar com x em célula correspondente)**     |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Implementado de Acordo com Especificado | Implementado mas diferente de Especificado | Não Implementado | Não Especificado (criado de novo) | | Nome SP  (tal como especificado) |  |  |  |  | | Nome SP  (tal como especificado) |  |  |  |  | | Nome SP  (tal como especificado) |  |  |  |  | |  |  |  |  |  | |

### Stored Procedures Implementados

|  |
| --- |
| 1. Nome SP: \_\_\_\_\_  //*Breve Descrição*  *Código*  2. Nome SP: \_\_\_\_\_  //*Breve Descrição*  *Código*  3. Nome SP: \_\_\_\_\_  //*Breve Descrição*  *Código* |

## Especificação de Migração entre Bases de Dados

### Esquema relacional da base de Dados Mysql especificada (destino)

### Forma de Migração Especificada

### Utilizadores Especificados

### Triggers de suporte à migração de dados especificados

### Stored Procedures de suporte à migração de dados especificados

### Eventos de suporte à migração de dados especificados

### PHP de suporte à migração de dados especificado

## Avaliação das especificações do próprio grupo Migração

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): \_\_\_\_\_\_\_\_\_\_\_\_  Justificação:  <fazer um resumo dos principais pontos fracos e fortes.  Depois de ler esta secção o leitor deve ter uma visão sobre que secções estavam mais fracas (SP? Forma de Migração Base de dados?)> |

## Implementação da Migração de Dados

### Utilizadores Implementado

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **Tabela** | **Tipo de Utilizador** | | | | Tipo 1 | Tipo 2 | … | | T1 | E | - |  | | T2 | L | E |  | | … |  |  |  | | **Stored Proc.** |  |  |  | | SP1 | X | - |  | | … |  |  |  | |

### Lista Triggers

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Lista de Triggers (para cada trigger assinalar com x em célula correspondente)**     |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Implementado de Acordo com Especificado | Implementado mas diferente de Especificado | Não Implementado | Não Especificado (criado de novo) | | Nome Trigger  (tal como especificado) |  |  |  |  | | Nome Trigger  (tal como especificado) |  |  |  |  | | Nome Trigger  (tal como especificado) |  |  |  |  | |  |  |  |  |  | |

### Triggers Implementados

|  |
| --- |
| 1. Nome Trigger: \_\_\_\_\_  //*Breve Descrição*  *Código*  2. Nome Trigger: \_\_\_\_\_  //*Breve Descrição*  *Código*  3. Nome Trigger: \_\_\_\_\_  //*Breve Descrição*  *Código* |

### Lista de Stored Procedures

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Lista de SP (para cada SP assinalar com x em célula correspondente)**     |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Implementado de Acordo com Especificado | Implementado mas diferente de Especificado | Não Implementado | Não Especificado (criado de novo) | | Nome SP  (tal como especificado) |  |  |  |  | | Nome SP  (tal como especificado) |  |  |  |  | | Nome SP  (tal como especificado) |  |  |  |  | |  |  |  |  |  | |

### Stored Procedures Implementados

|  |
| --- |
| 1. Nome SP: \_\_\_\_\_  //*Breve Descrição*  *Código*  2. Nome SP: \_\_\_\_\_  //*Breve Descrição*  *Código*  3. Nome SP: \_\_\_\_\_  //*Breve Descrição*  *Código* |

### Lista Eventos

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Lista de Eventos (para cada evento assinalar com x em célula correspondente)**     |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Implementado de Acordo com Especificado | Implementado mas diferente de Especificado | Não Implementado | Não Especificado (criado de novo) | | Nome Evento  (tal como especificado) |  |  |  |  | | Nome Evento  (tal como especificado) |  |  |  |  | | Nome Evento  (tal como especificado) |  |  |  |  | |  |  |  |  |  | |

### Eventos Implementados

|  |
| --- |
| 1. Nome Evento: \_\_\_\_\_  //*Breve Descrição*  *Código*  2. Nome Evento: \_\_\_\_\_  //*Breve Descrição*  *Código*  3. Nome Evento: \_\_\_\_\_  //*Breve Descrição*  *Código* |

### PHP Implementado

|  |
| --- |
| *Código* |

Avaliação Global da Qualidade das Especificações do próprio grupo

|  |
| --- |
| Avaliação (A,B,C,D,E) : \_\_\_\_\_\_\_\_\_\_\_\_\_  Utilize a seguinte escala:  A: - 1 – 5 valores B: 6 – 9 valores C: 10 – 13 Valores D: 14 – 17 valores E: 18 – 20 valores |

**Três principais deficiências de especificação que tiveram impacto mais negativo na qualidade da implementação**

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Resumo de Avaliações de Qualidade Anteriores (para cada linha assinalar com x em célula correspondente)**   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Fraco | Razoável | Bom | Muito Bom | | BD Sybase |  |  |  |  | | Triggers Log |  |  |  |  | | SP Log |  |  |  |  | | Utilizadores Log |  |  |  |  | | BD Mysql |  |  |  |  | | Forma Migração |  |  |  |  | | Triggers Migração |  |  |  |  | | SP Migração |  |  |  |  | | Eventos Migração |  |  |  |  | | Utilizadores Migração |  |  |  |  | | PHP Migração |  |  |  |  | |

## Comparação de Implementações (ficheiro versos PHP)

<Resumo da analise das diferenças entre as duas abordagens, indicando vantagens e desvantagens de ambas. Nas secções seguintes as diferenças deverão ser fundamentadas e, quando relevante, suportadas por testes efectuados de forma rigorosa. Os testes deverão ser descritos de modo a poderem ser replicados por outras pessoas.>

### Eficiência de Migração

<Apresentar gráficos e quadros resumo de valores. Cada grupo decide que gráficos e quadros apresenta, mas é importante que se fique com uma noção clara das diferenças de tempos face às quantidades de dados, para cada fase do processo.

Os grupos deverão tentar explicar as diferenças de valores encontradas.>

### Robustez

<Deverá ser analisado e discutido o comportamento das migrações em situações de ruptura: falha de energia, erro de software, etc.>

### Flexibilidade / Dependência

<Deverá ser analisado e discutido o comportamento das migrações em termos de

1. Flexibilidade: facilidade de efectuar alterações, (por exemplo, alterar a periodicidade de ruptura) por pessoas não técnicas;
2. Dependência: de que forma o mau comportamento de uma base de dados afecta a outra base de dados.>

### Segurança

<Deverá ser analisado e discutido as eventuais diferenças em termos de segurança dos dois processos de migração (por exemplo, menor ou maior exposição de informação>

## Auditoria de Dados (base de dados origem)

<Deverá ser criada uma interface HTML onde, através de php, o auditor após se autenticar e selecionar uma tabela, poderá visualizar as acessos à mesma. Fica a cabo do grupo a definição da flexibilidade/usabilidade da visualização.>