Sistemas de Informação Distribuídos

Licenciaturas em Engenharia Informática e Informática e Gestão de Empresas

2019-2020, Segundo Semestre

Deteção de Intrusão e Incêndio em Museus

Auditoria e Migração

Identificação do grupo autor da especificação (Etapa A): **Grupo 20**

|  |  |  |
| --- | --- | --- |
| Número | Nome | Foto |
| 20687 | Pedro Santiago | C:\Users\pmsan\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\D03016C7.tmp |
| 82286 | Bin Guan | C:\Users\pmsan\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\948FF36D.tmp |
| 79142 | Tomás Ferreira | C:\Users\pmsan\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\D9744503.tmp |
| 82652 | Alexandre Ferreira | C:\Users\pmsan\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\952698FF.tmp |
| 82453 | José Pedro Fernandes | C:\Users\pmsan\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\B4500909.tmp |
| 82608 | Francisco Barros |  |
| Especificação: PHP Ficheiro  X | | |

Identificação do grupo autor da implementação (Etapas B e C): **Grupo 23**

|  |  |  |
| --- | --- | --- |
| Número | Nome | Foto |
| 82493 | Miguel Diaz Gonçalves |  |
| 83380 | Gonçalo Dias do Amaral |  |
| 82361 | André Freitas |  |
| 82946 | Pedro Jones |  |
| 74278 | Dmytro Astashov |  |
| 73788 | Vitor Manuel Figueira Canhão |  |
| Especificação: PHP Ficheiro  Implementação: PHP Ficheiro  X  X | | |

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;
* 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 (Etapa B) 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 (*copy e paste*),
* As restantes secções são preenchidas normalmente pelo grupo que recebe o presente documento do outro grupo.

Índice

[1 Etapa A e B 5](#_Toc25168883)

[1.1 Esquema relacional da base de Dados Mysql Origem 5](#_Toc25168884)

[1.2 Apreciação Crítica e esquema relacional implementado 6](#_Toc25168885)

[1.3 Esquema relacional da base de Dados Mysql Destino 7](#_Toc25168886)

[1.4 Apreciação Crítica e esquema relacional implementado 8](#_Toc25168887)

[1.5 Migração entre Bases de Dados 9](#_Toc25168888)

[1.5.1 Forma de Migração 9](#_Toc25168889)

[1.5.2 Apreciação Crítica à especificação da forma de migração 10](#_Toc25168890)

[1.6 Utilizadores Base de Dados de Origem 11](#_Toc25168891)

[1.7 Apreciação Crítica a Gestão de Utilizadores Base de Dados de Origem 11](#_Toc25168892)

[1.8 Utilizadores Base de Dados de Destino 12](#_Toc25168893)

[1.9 Apreciação Crítica a Gestão de Utilizadores Base de Dados de Destino 12](#_Toc25168894)

[1.10 Triggers de suporte à criação de logs e migração 13](#_Toc25168895)

[1.10.1 Apreciação Crítica de triggers para gestão de logs e migração 14](#_Toc25168896)

[1.10.1.1 Triggers Implementados para gestão de logs e migração 15](#_Toc25168897)

[1.11 Stored Procedures de suporte à criação de logs e migração 16](#_Toc25168898)

[1.11.1 Apreciação Crítica de Stored Procedures 17](#_Toc25168899)

[1.11.2 Stored Procedures Implementados 18](#_Toc25168900)

[1.12 Eventos de suporte à migração de dados 19](#_Toc25168901)

[1.12.1 Apreciação Crítica de Eventos 20](#_Toc25168902)

[1.12.2 Eventos Implementados 21](#_Toc25168903)

[1.13 PHP suporte à migração de dados (se relevante) 22](#_Toc25168904)

[1.13.1 Apreciação Crítica ao PHP especificado 23](#_Toc25168905)

[1.13.2 PHP Implementado 24](#_Toc25168906)

[1.14 Avaliação Global de especificações da Etapa A 25](#_Toc25168907)

[2 Etapa C (Especificação e Implementação do Próprio Grupo) 27](#_Toc25168908)

[2.1 Especificação do Próprio Grupo (versão compactada) 27](#_Toc25168909)

[2.1.1 Especificação do Esquema relacional da base de Dados Origem 27](#_Toc25168910)

[2.1.2 Especificação do Esquema relacional da base de Dados Destino 27](#_Toc25168911)

[2.1.3 Forma de Migração Especificada 27](#_Toc25168912)

[2.1.4 Especificação de Utilizadores 27](#_Toc25168913)

[2.1.5 Triggers de suporte à gestão de logs e migração 27](#_Toc25168914)

[2.1.6 Stored Procedures de suporte à gestão de logs e migração 27](#_Toc25168915)

[2.1.7 Eventos de suporte à migração de dados especificados 27](#_Toc25168916)

[2.1.8 PHP de suporte à migração de dados especificado 27](#_Toc25168917)

[2.2 Avaliação Global da Qualidade das Especificações do próprio grupo 28](#_Toc25168918)

[2.3 Implementação do Próprio Grupo 29](#_Toc25168919)

[2.3.1 Utilizadores implementados Base de Dados Origem 29](#_Toc25168920)

[2.3.1 Utilizadores implementados Base de Dados Destino 29](#_Toc25168921)

[2.3.2 Lista de Triggers 30](#_Toc25168922)

[2.3.3 Triggers Implementados 31](#_Toc25168923)

[2.3.4 Lista de Stored Procedures 32](#_Toc25168924)

[2.3.5 Stored Procedures Implementados 33](#_Toc25168925)

[2.3.6 Lista Eventos 34](#_Toc25168926)

[2.3.7 Eventos Implementados 35](#_Toc25168927)

[2.3.8 PHP Implementado 36](#_Toc25168928)

[3 Comparação de Implementações (ficheiro versos PHP) 37](#_Toc25168929)

[3.1 Eficiência de Migração 38](#_Toc25168930)

[3.2 Robustez 39](#_Toc25168931)

[3.3 Flexibilidade / Dependência 40](#_Toc25168932)

[3.4 Segurança 41](#_Toc25168933)

Deteção de Intrusão e Incêndio em Museus

# Etapa A e B

## Esquema relacional da base de Dados Mysql Origem

<Tem de incluir as tabelas e atributos que suportam a auditoria de dados. Deverá conter as regras de integridade das chaves estrangeiras e a indicação de atributos Unique e obrigatórios.

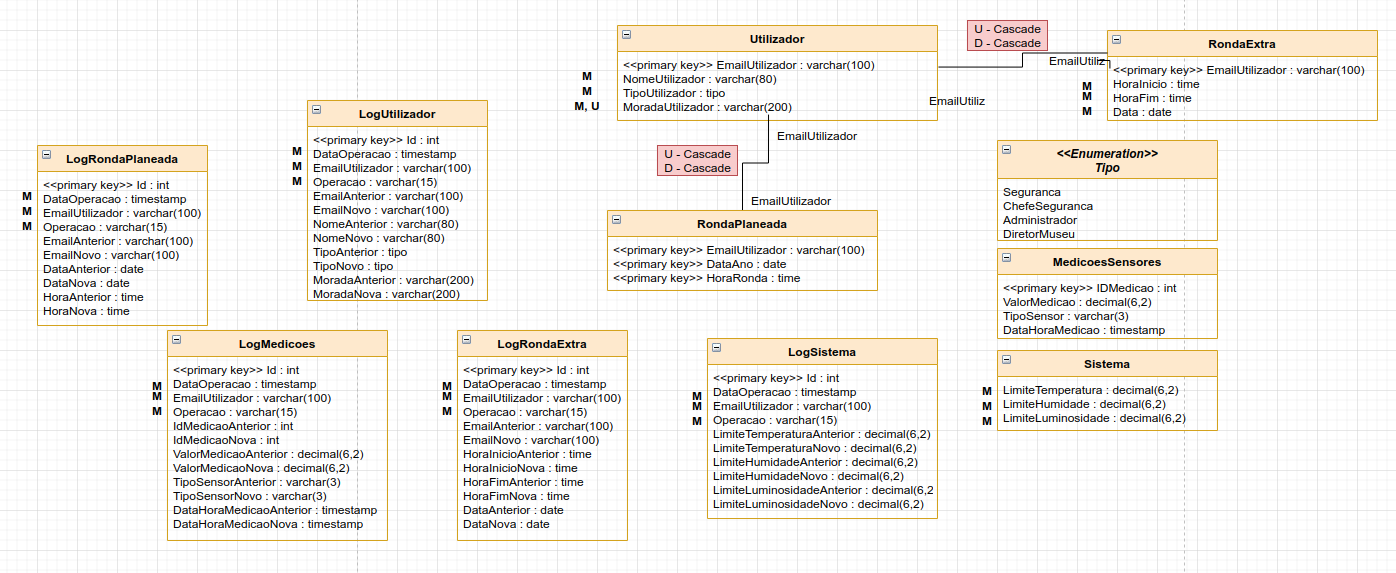
A base de dados deve respeitar o diagrama de classes (deverão ser criadas as tabelas para registo de logs). Excepcionalmente poderão ser adicionados, alterados ou removidos campos, desde que devidamente justificado no relatório. O esquema deverá ser apresentado exclusivamente através de um esquema semelhante ao aqui exemplificado:



Na ligação, junto a cada tabela, coloca-se o(s) atributo(s) que assegura a ligação entre as tabelas (sigla\_hotel em ambas as talelas, no exemplo) e a operação de integridade referencial escolhida para a operação de Delete (D) e Update (C).

À esquerda de cada campo de preenchimento obrigatório devem indicar M (de Mandatory), tal como exemplificado (não é necessário indicar nas chaves primárias). Os campos com valor único devem ser assinalados com a letra U.

Caso seja pertinente poderão ser adicionados comentários a justificar opções pouco óbvias.>



Retirámos o DiaSemana e inserimos os campos DataAno e HoraRonda diretamente em RondaPlaneada. A nossa base para esta decisão é o facto de DiaSemana ser algo demasiado vago (Há muitas segundas-feiras num ano), podendo ser compensado com um dia especifico do ano - date - que permite uma melhor localização temporal em caso de qualquer tipo de problema.

Exemplo: Se uma peça do museu tiver desaparecido e o último inventário onde essa peça apareceu tenha sido há 1 mês, podemos facilmente procurar as rondas dos dias imediamente aseguir ao dia que foi feito o inventário.

Decidimos pelo Cascade no que diz respeito a todas as operações de update, pois queremos que as alterações feitas numa tabela sejam devidamente actualizadas nas tabelas ligadas à mesma.

Relativamente à remoção de um Utilizador, esta operação deverá ter um comportamento Cascade, ou seja, a eliminação de um utilizador deverá ser seguida da eliminação das Rondas Planeadas e das Rondas Extras criadas para este utilizador. Os valores próprios para cada Ronda Planeada ou Ronda Extra terão sido devidamente registados nos respectivos Logs.

Em termos das tabelas de logs, estas deverão ser totalmente independentes pois queremos que todas as operações fiquem imaculadamente guardadas. Consideramos que a DataOperacao, EmailUtilizador e Operacao sejam Mandatory em todas as tabelas de Logs pois estes são dados absolutamente necessários para a posterior auditoria. Decidimos pôr os valores antes e depois da operação efectuada apenas para facilitar a auditoria.

Quanto ao utilizador, estará sempre na tabela Utilizador e ao mesmo tempo no sistema da BD - deve ser feito através de um SP para criar o utilizador e associar um user com uma senha.

Para termos uma base de dados mais segura, a password associada ao utilizador da BD será encriptada pelo sistema da BD e mantida internamente, pelo que nunca iremos ter nenhum campo ou tabela com as passwords de utilizadores. Assim, isolamos este tipo de informação sensível dos utilizadores da BD.

- Discussão sobre Tabela dos Utilizadores: saber se se deveria usar uma Primary Key como número de Empregado, para poder relacionar com outras tabelas e cada utilizador poder efectuar login com essa referência, e para poder saber se ainda era um empregado da Instituição (com um boolean para a propriedade “Empregado”). Na discussão de grupo, não foram mencionadas objecções para o atributo “Email” como Primary Key, pelo que foi mantido como tal.

- Discussão sobre Logs: principal divergência pelo grupo, por haver duas correntes distintas para a criação dos Logs do sistema: a primeira a ser vista como Logs por Grupos de Utilizadores, onde cada tabela gerada pelos triggers accionados seria actualizada em função do Tipo de Utilizador (foi convencionado que seriam 4 tipos de utilizador + 1 (auditor de dados)); a outra opinião era no modelo de haver Logs para cada uma das Tabelas dado o modelo relacional, assim como um Log que englobasse todos os triggers activados, tendo assim uma dupla certificação dos acontecimentos nos dados.

## Apreciação Crítica e esquema relacional implementado

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): Razoável  Breve Justificação:  Utilizando o email como primary key na tabela de utilizadores possibilita ataques por SQL Injection.  Foram feitas alterações? (Sim/Não): \_\_\_\_\_\_\_  **Novo Esquema (assinale e justifique as alterações)**  <Apenas preencher caso tenham procedido a alterações> |

## Esquema relacional da base de Dados Mysql Destino

A BD de auditoria (destino) apenas terá tabelas independentes com a informação de que operações foram efectuadas na BD origem. Como tal, não haverá regras de integridade relacional (Cascade ou Restrict). Cada tabela terá os valores antes e depois da operação para facilitar o trabalho do auditor e terá como primary key um ID que será incrementado a cada operação nova.

Consideramos que a DataOperacao, EmailUtilizador e Operacao sejam Mandatory em todas as tabelas de Logs pois estes são dados absolutamente necessários para a posterior auditoria.

## Apreciação Crítica e esquema relacional implementado

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): Boa  Breve Justificação: A especificação da base de dados cumpre todos os requisitos do enunciado  Foram feitas alterações? (Sim/Não): Sim  **Novo Esquema (assinale e justifique as alterações)**  Não implementamos o autoincrement na primary key ID das tabelas de log uma vez que como os registos são copiados integralmente das tabelas de origem este já foi criado e a sua cópia dá erro. |

## Migração entre Bases de Dados

### Forma de Migração

- Discussão sobre a forma de fazer a migração entre bases de dados: no tipo de ficheiro que deverá ser criado, pensamos que o melhor tipo de ficheiro será o CSV, pois será mais fácil de exportar a informação necessária para a migração dos dados. Para manter a coerência de dados durante a fase de migração, pensamos que deverá ser seguida uma sequência de eventos, desde a criação de um ficheiro, exportação dos dados da base de dados de origem, seguida da importação de dados na base de dados de destino, e, por fim, a eliminação do ficheiro. Caso este ficheiro não tenha sido criado, na fase de importação de dados será detectada a ausência do ficheiro e deverá voltar ao início do processo, mantendo a integridade dos dados e segurança na migração dos dados.

Informação a exportar:

- Se BD destino vazia: Cópia dos logs, réplica das Tabelas de Logs da BD origem.

- Actualização diária BD destino: importação de informação com base nos novos registos do último dia, utilizando os logs.

Periodicidade:

Todos os dias, à noite.

Quem toma iniciativa:

BD origem toma a iniciativa de exportar a informação para a BD destino.

### Apreciação Crítica à especificação da forma de migração

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): Razoável  **Análise crítica (clareza, completude, rigor):**  A metodologia proposta para a migração não prevê a hipótese de haver um erro e não ser feita a cópia um ou mais dias. Nesse caso apenas é copiada a informação do último dia.  Não é claro também o que quer dizer informação do último dia. Pode ser o último dia existente na base de dados ou o dia actual da cópia.  A Base de dados de destino nunca poderá estar vazia uma vez que a sua implementação de acordo com o modelo proposto no Esquema relacional, faz parte dos requisitos enunciado. As tabelas essas sim podem, no início, estar vazias (sem registos) mas deveram estar criadas. |

## Utilizadores Base de Dados de Origem

<Nesta secção deverá ser explicado de que forma deverá ser feita a manutenção (e.g., a criação e privilégios) de utilizadores Mysql. Nomeadamente deverá ser indicado, para cada tipo de utilizador, que privilégios ele tem sobre que tabelas e Stored procedures (caso sejam referenciados mais adiante).

**Sugestão de tabela**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela** | **Tipo de Utilizador** | | | |
| **Administrador** | **Director de Museu** | **Chefe de Segurança** | **Segurança** |
| **Utilizador** | **RE** | **-** | **-** | **-** |
| **Rondas Planeadas** | **REL** | **L** | **REL** | **L** |
| **Rondas Extra** | **RL** | **L** | **L** | **-** |
| **Sistema** | **REL** | **L** | **-** | **-** |
| **MediçõesSensores** | **L** | **L** | **-** | **-** |
| **Stored Proc.** |  |  |  |  |
| **Criação de Utilizador** | **X** | **-** | **-** | **-** |
| **Remoção de Utilizador** | **X** | **-** | **-** | **-** |
| **Alteração de Password** | **X** | **X** | **X** | **X** |
| **Alteração de Morada** | **X** | **X** | **X** | **X** |
| **Inserção de Ronda Planeada** | **X** | **-** | **X** | **-** |
| **Alteração de Ronda Planeada** | **X** | **-** | **X** | **-** |
| **Eliminação de Ronda Planeada** | **X** | **-** | **X** | **-** |
| **Inserção de Ronda Extra** | **X** | **-** | **X** | **X** |
| **Consultar tabela utilizadores** | **X** | **X** | **X** | **-** |
| **Exportação de Informação para Migração** | **-** | **-** | **-** | **-** |

**Em que R=Remoção E=Escrita, L=Leitura, X=Executar e - = sem permissões**

Caso seja pertinente poderão ser adicionados comentários explicar a estratégia a seguir.>

A tabela Utilizador estará limitada ao máximo em termos de permissões, por forma a obrigar a utilização do Store Procedure (ConsultarUtilizadores) - assim, haverão logs sobre quem consultou esta tabela.

Apenas daremos permissões de leitura ao director do museu, todas as outras operações são irrelevantes para este tipo de utilizador, na nossa opinião.

O Administrador tem permissões de escrita, remoção e leitura em todas as tabelas excepto a Utilizadores (tem que utilizar o SP) e só de leitura na tabela de medições dos sensores e nos logs, pois queremos que estes dados sejam o mais correctos possível.

O chefe de segurança poderá monitorizar, alterar ou remover a tabela das rondas planeadas, bem como o poder de consultar e registar as rondas extraordinárias.

O segurança apenas terá permissão de leitura para consultar as rondas planeadas. Para registar uma ronda extra, o segurança terá usar o SP - Inserção de Ronda Extra.

O SP “Exportação de Informação para Migração” só deve ser executado pelo Sistema Operativo.

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

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): Boa  **Análise crítica (clareza, completude, rigor):**  Falta acrescentar a escrita ao “Administrador” na ronda extra como comprova a frase: “O Administrador tem permissões de escrita, remoção e leitura em todas as tabelas excepto a Utilizadores” O Chefe de segurança devia também ter o poder de só monitorizar as medições dos sensores.  **Solução Implementada:**  Implementado tal como foi sugerido na tabela |

## Utilizadores Base de Dados de Destino

|  |  |
| --- | --- |
|  | **Tipo de Utilizador** |
| **Tabela** | **Auditor** |
| LogUtilizador | L |
| LogRondaExtra | L |
| LogRondaPlaneada | L |
| LogMedicoes | L |
| LogSistema | L |
| **Stored Proc.** |  |
| ConsultaUtilizador | X |
| ConsultaRondaExtra | X |
| ConsultaRondaPlaneada | X |
| ConsultaMedicoes | X |
| ConsultaSistema | X |

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

Para a BD de auditoria, apenas será necessário dar permissões de leitura sobre todas as tabelas, mais do que essa permissão é desnecessário e até consideramos que seria perigoso para a integridade dos logs.

Criámos alguns stored procedure, que não são mais do que comandos select sobre cada uma das tabelas. Apenas como forma de facilitar o trabalho do auditor.

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

|  |
| --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): Muito Boa  **Análise crítica (clareza, completude, rigor):**  A gestão de utilizadores faz sentido e cumpre com os requisitos.  **Solução Implementada:**  Implementado tal como foi sugerido na tabela |

## Triggers de suporte à criação de logs e migração

<Caso seja pertinente poderão ser adicionados comentários explicar a estratégia a seguir.>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome Trigger** | **Base de Dados** | **Tabela** | **Tipo de Operação (I,U,D)** | **Evento**  **(A, B)** | **Notas (apenas indicar aquilo que não seja óbvio)** |
| Inserção Utilizador | Origem | Utilizador | I | Inserção |  |
| Alteração Utilizador | Origem | Utilizador | U | Alteração |  |
| Remoção Utilizador | Origem | Utilizador | D | Remoção |  |
| Inserção RondaPlaneada | Origem | Ronda Planeada | I | Inserção |  |
| Alteração RondaPlaneada | Origem | Ronda Planeada | U | Alteração |  |
| Remoção RondaPlaneada | Origem | Ronda Planeada | D | Remoção |  |
| Inserção RondaExtra | Origem | Ronda Extra | I | Inserção |  |
| Remoção RondaExtra | Origem | Ronda Extra | D | Remoção | Só existe no caso de um utilizador ser eliminado, vão ser removidas as rondas e registadas essas remoções no log |

Todos os triggers de inserção têm que ter todas as colunas que se referem a campos anteriores a NULL.

Exemplo:

Inserção de um Utilizador na Tabela de Log Utilizador:

(id); “dia”; “Email@administrador.com”; I; NULL; “EmailNovo”; NULL; “NomeNovo”; NULL; “TipoUtilizadorNovo”; NULL; “MoradaNova”.

Todos os triggers de remoção têm que ter todas as colunas que se referem a campos novos a NULL.

Exemplo:

Remoção de um Utilizador na Tabela de Log Utilizador:

(id); “dia”; “Email@administrador.com”; D; “EmailAnterior”; NULL; “NomeAnterior”; NULL; “TipoUtilizadorAnterior”; NULL; “MoradaAnterior”; NULL.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): Fraca  Breve Justificação:  Não foram especificados os triggers para as tabelas, Medicoes e Sistema. Falta também um trigger para o update na tabela de RondaExtra. No caso do trigger Remoção RondaExtra a garantia que em caso de um utilizador ser apagado todas as suas rondas extra também o são é dada pela Foreign Key Option, On Delete Cascade.  **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) | | Inserção Utilizador | X |  |  |  | | Alteração Utilizador | X |  |  |  | | Remoção Utilizador | X |  |  |  | | Inserção RondaPlaneada | X |  |  |  | | Alteração RondaPlaneada | X |  |  |  | | Remoção RondaPlaneada | X |  |  |  | | Inserção RondaExtra | X |  |  |  | | Remoção RondaExtra |  | X |  |  | |

#### Triggers Implementados para gestão de logs e migração

|  |
| --- |
| 1. Nome Trigger:utilizador\_AFTER\_INSERT   -- Insert no utilizador  CREATE DEFINER=`root`@`localhost` TRIGGER `new\_schema2`.`utilizador\_AFTER\_INSERT` AFTER INSERT ON `utilizador`  FOR EACH ROW  BEGIN  insert into new\_schema2.logutilizador  (DataOperacao, EmailUtilizador, Operacao, EmailNovo, NomeNovo, TipoNovo, MoradaNova)  Values  (getdate(), user(), 'I', NEW.EmailUtilizador, NEW.NomeUtilizador, NEW.TipoUtilizador, NEW.MoradaUtilizador);  END  2. Nome Trigger: utilizador\_AFTER\_UPDATE  -- Update no utilizador  CREATE DEFINER=`root`@`localhost` TRIGGER `new\_schema2`.`utilizador\_AFTER\_UPDATE` AFTER UPDATE ON `utilizador`  FOR EACH ROW  BEGIN  insert into new\_schema2.logutilizador  (DataOperacao, EmailUtilizador, Operacao, EmailNovo, EmailAnterior, NomeNovo, NomeAnterior, TipoNovo, TipoAnterior, MoradaNova, MoradaAnterior)  Values  (getdate(), user(), 'U', NEW.EmailUtilizador, OLD.EmailUtilizador, NEW.NomeUtilizador, OLD.NomeUtilizador, NEW.TipoUtilizador, OLD.TipoUtilizador, NEW.MoradaUtilizador, OLD.MoradaUtilizador);  END  3. Nome Trigger: utilizador\_AFTER\_DELETE  -- Delete no utilizador  *CREATE DEFINER=`root`@`localhost` TRIGGER `new\_schema2`.`utilizador\_AFTER\_DELETE` AFTER DELETE ON `utilizador`*  *FOR EACH ROW*  *BEGIN*  *insert into new\_schema2.logutilizador*  *(DataOperacao, EmailUtilizador, Operacao, EmailAnterior, NomeAnterior, TipoAnterior, MoradaAnterior)*  *Values*  *(getdate(), user(), 'D',OLD.EmailUtilizador, OLD.NomeUtilizador,OLD.TipoUtilizador,OLD.MoradaUtilizador);*  *END*  4. Nome Trigger: rondaplaneada\_AFTER\_INSERT  -- Insert no rondaplaneada  CREATE DEFINER=`root`@`localhost` TRIGGER `new\_schema2`.`rondaplaneada\_AFTER\_INSERT` AFTER INSERT ON `rondaplaneada`  FOR EACH ROW  BEGIN  insert into new\_schema2.logrondaplaneada  (DataOperacao, EmailUtilizador, Operacao, EmailNovo, DataNova, HoraNova)  Values  (getdate(), user(), 'I', NEW.EmailUtilizador, NEW.DataAno, NEW.HoraRonda);  END  5. Nome Trigger: rondaplaneada\_AFTER\_UPDATE  -- Update no rondaplaneada  CREATE DEFINER=`root`@`localhost` TRIGGER `new\_schema2`.`rondaplaneada\_AFTER\_UPDATE` AFTER UPDATE ON `rondaplaneada`  FOR EACH ROW  BEGIN  insert into new\_schema2.logrondaplaneada  (DataOperacao, EmailUtilizador, Operacao, EmailNovo, EmailAnterior, DataNova, DataAnterior, HoraNova, HoraAnterior)  Values  (getdate(), user(), 'U', NEW.EmailUtilizador, OLD.EmailUtilizador, NEW.DataAno, OLD.DataAno, NEW.HoraRonda, OLD.HoraRonda);  END  6. Nome Trigger: rondaplaneada\_AFTER\_DELETE  -- Delete no rondaplaneada  CREATE DEFINER=`root`@`localhost` TRIGGER `new\_schema2`.`rondaplaneada\_AFTER\_DELETE` AFTER DELETE ON `rondaplaneada`  FOR EACH ROW  BEGIN  insert into new\_schema2.logrondaplaneada  (DataOperacao, EmailUtilizador, Operacao, EmailAnterior, DataAnterior, HoraAnterior)  Values  (getdate(), user(), 'D',OLD.EmailUtilizador, OLD.DataAno, OLD.HoraRonda);  END  7. Nome Trigger: rondaextra\_AFTER\_INSERT  -- Insert no rondaextra  CREATE DEFINER=`root`@`localhost` TRIGGER `new\_schema2`.`rondaextra\_AFTER\_INSERT` AFTER INSERT ON `rondaextra`  FOR EACH ROW  BEGIN  insert into new\_schema2.logrondaextra  (DataOperacao, EmailUtilizador, Operacao, EmailNovo, HoraInicioNova, HoraFimNova, DataNova)  Values  (getdate(), user(), 'I', NEW.EmailUtilizador, NEW.HoraInicio, NEW.HoraFim, NEW.Data);  END  8. Nome Trigger: rondaextra\_AFTER\_DELETE  -- Delete no rondaextra  CREATE DEFINER=`root`@`localhost` TRIGGER `new\_schema2`.`rondaextra\_AFTER\_DELETE` AFTER DELETE ON `rondaextra`  FOR EACH ROW  BEGIN  insert into new\_schema2.logrondaextra  (DataOperacao, EmailUtilizador, Operacao, EmailAnterior, HoraInicioAnterior, HoraFimAnterior, DataAnterior)  Values  (getdate(), user(), 'D', OLD.EmailUtilizador, OLD.HoraInicio, OLD.HoraFim, OLD.Data);  END |

## Stored Procedures de suporte à criação de logs e migração

<Caso seja pertinente poderão ser adicionados comentários explicar a estratégia a seguir.>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Base de Dados | Nome Procedimento | Parâmetros Entrada | Parâmetros Saída | Muito breve descrição |
| Origem | **Criação de Utilizador** | Email, Nome, Tipo, Morada, Senha |  | Insere utilizador na tabela e no sistema da BD |
| Origem | **Remoção de Utilizador** | Email |  | Elimina utilizador do sistema BD e tabela Utilizador |
| Origem | **Alteração de Password** | NovaPassword |  | Altera a password na BD |
| Origem | **Alteração de Morada** | NovaMorada |  | Altera a morada na tabela Utilizador |
| Origem | **Inserção de Ronda Planeada** | Email(de quem fará a ronda), data, hora |  | Insere uma ronda na tabela RondaPlaneada |
| Origem | **Alteração de Ronda Planeada** | Email (de quem fará a ronda), data, hora |  | Altera a ronda na tabela RondaPlaneada |
| Origem | **Eliminação de Ronda Planeada** | Email, Data, Hora |  | Remove uma ronda previamente planeada |
| Origem | **Inserção de Ronda Extra** | Email, HoraInicio, HoraFim, Data |  | Regista uma ronda extraordinária |
| Origem | **Consultar tabela utilizadores** |  | Tabela Utilizador | Comando select sobre a tabela Utilizador |
| Destino | **Consulta Log Utilizador** |  | Tabela Log Utilizador | Consulta tabela LogUtilizador |
| Destino | **Consulta Log Ronda Extra** |  | Tabela Log Ronda Extra | Consulta tabela LogRondaExtra |
| Destino | **Consulta Log Ronda Planeada** |  | Tabela Log Ronda Planeada | Consulta tabela LogRondaPlaneada |
| Destino | **Consulta Log Medicoes** |  | Tabela Log Medições | Consulta tabela LogMedicoes |
| Destino | **Consulta Log Sistema** |  | Tabela Log Sistema | Consulta tabela LogSistema |
| Origem | **Exportação de Informação para Migração** |  | Dados de Logs | Copia informação dos logs do dia anterior para ficheiro |

### Apreciação Crítica de Stored Procedures

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): Razoável  Breve Justificação:  Na nossa opinião foram especificados um número exagerado de Stored Procedures. Não é necessário fazer tudo por Stored Procedures.  Não implementamos o SP *Alteração de Ronda Planeada* uma vez que todos os parâmetros de entrada são as primary keys da tabela RondaPlaneada, não há dados para alterar.  **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) | | Criação de Utilizador | X |  |  |  | | Remoção de Utilizador | X |  |  |  | | Alteração de Password | X |  |  |  | | Alteração de Morada | X |  |  |  | | Inserção de Ronda Planeada | X |  |  |  | | Alteração de Ronda Planeada |  |  | X |  | | Eliminação de Ronda Planeada | X |  |  |  | | Inserção de Ronda Extra | X |  |  |  | | Consultar tabela utilizadores | X |  |  |  | | Consulta Log Utilizador | X |  |  |  | | Consulta Log Ronda Extra | X |  |  |  | | Consulta Log Ronda Planeada | X |  |  |  | | Consulta Log Medicoes | X |  |  |  | | Consulta Log Sistema | X |  |  |  | | Exportação de Informação para Migração | X |  |  |  | |

### Stored Procedures Implementados

|  |
| --- |
| 1. Nome SP: CriarUtilizador   -- Cria Utilizador na tabela utilizadores e no SQL  CREATE DEFINER=`root`@`localhost` PROCEDURE `CriarUtilizador`(iEmailUtilizador varchar(100),iNomeUtilizador varchar(80) ,iTipoUtilizador enum('Seguranca','Chefe Seguranca','Administrador','DiretorMuseu') ,iMoradaUtilizador varchar(200) ,iSenhaUtilizador varchar(30))  BEGIN  SET @criarUtilizadorCMD = concat('CREATE USER ''', iEmailUtilizador, '''@''', 'localhost', ''' IDENTIFIED BY ''', iSenhaUtilizador, ''';');  PREPARE criarUtilizadorStatement FROM @criarUtilizadorCMD;  EXECUTE criarUtilizadorStatement;  DEALLOCATE PREPARE criarUtilizadorStatement;    SET @garantirUtilizadorCMD = concat('GRANT ''', iTipoUtilizador ,''' TO ''', iEmailUtilizador, '''@''', 'localhost', ''';');  PREPARE garantirUtilizadorStatement FROM @garantirUtilizadorCMD;  EXECUTE garantirUtilizadorStatement;  DEALLOCATE PREPARE garantirUtilizadorStatement;    INSERT into utilizador  Values(iEmailUtilizador,iNomeUtilizador,iTipoUtilizador,iMoradaUtilizador);  END   1. Nome SP: RemoverUtilizador   –- Apaga Utilizador tanto na base de dados como o utilizador SQL  CREATE DEFINER=`root`@`localhost` PROCEDURE `RemoverUtilizador`(iEmailUtilizador varchar(100))  BEGIN  SET @apagarUtilizadorCMD = concat('DROP USER ''', iEmailUtilizador, '''@''', 'localhost', ''';');  PREPARE apagarUtilizadorStatement FROM @apagarUtilizadorCMD;  EXECUTE apagarUtilizadorStatement;  DEALLOCATE PREPARE apagarUtilizadorStatement;  DELETE FROM utilizador WHERE EmailUtilizador = iEmailUtilizador;  END   1. Nome SP: AlterarPasswordUtilizador   –- Altera Password do utilizador SQL  CREATE DEFINER=`root`@`localhost` PROCEDURE `RemoverUtilizador`(iEmailUtilizador varchar(100))  BEGIN  SET @apagarUtilizadorCMD = concat('DROP USER ''', iEmailUtilizador, '''@''', 'localhost', ''';');  PREPARE apagarUtilizadorStatement FROM @apagarUtilizadorCMD;  EXECUTE apagarUtilizadorStatement;  DEALLOCATE PREPARE apagarUtilizadorStatement;  DELETE FROM utilizador WHERE EmailUtilizador = iEmailUtilizador;  END   1. Nome SP: AlterarMoradaUtilizador   -- Altera a morada do utilizador na tabela utilizador  CREATE DEFINER=`root`@`localhost` PROCEDURE `AlterarMoradaUtilizador`(iEmailUtilizador varchar(100), iMoradaUtilizador varchar(200))  BEGIN  UPDATE utilizador SET MoradaUtilizador = iMoradaUtilizador WHERE EmailUtilizador= iEmailUtilizador;  END   1. Nome SP: CriarRondaPlaneada   -- Cria Ronda Planeada  CREATE DEFINER=`root`@`localhost` PROCEDURE `CriarRondaPlaneada`(iEmailUtilizador varchar(100), iDataAno date, iHoraRonda time)  BEGIN  INSERT into rondaplaneada  Values(iEmailUtilizador, iDataAno, iHoraRonda);  END   1. Nome SP: RemoverRondaPlaneada   -- Remove Ronda Planeada  CREATE DEFINER=`root`@`localhost` PROCEDURE `RemoverRondaPlaneada`(iEmailUtilizador varchar(100), iDataAno date, iHoraRonda time)  BEGIN  DELETE FROM rondaplaneada WHERE EmailUtilizador = iEmailUtilizador AND DataAno = iDataAno AND HoraRonda = iHoraRonda;  END   1. Nome SP: CriarRondaExtra   -- Cria Ronda Extra  CREATE DEFINER=`root`@`localhost` PROCEDURE `CriarRondaExtra`(iEmailUtilizador varchar(100), iHoraInicio time, iHoraFim time, iData date)  BEGIN  INSERT into rondaextra  Values(iEmailUtilizador, iHoraInicio, iHoraFim, iData);  END   1. Nome SP: ConsultaUtilizador   -- Consulta Utilizadores e regista que foi acedido na tabela logutilizadores  CREATE DEFINER=`root`@`localhost` PROCEDURE `ConsultaUtilizador`()  BEGIN  INSERT INTO logutilizador(DataOperacao,EmailUtilizador,Operacao) VALUES(GETDATE(), CURRENT\_USER(),'S')  SELECT \* FROM utilizador  END   1. Nome SP: ConsultarLogUtilizador   -- Consulta tabela logutilizador  CREATE DEFINER=`root`@`localhost` PROCEDURE `ConsultaLogUtilizador`()  BEGIN  SELECT \* FROM logutilizador  END   1. Nome SP: ConsultaLogRondaExtra   -- Consulta tabela logrondaextra  CREATE DEFINER=`root`@`localhost` PROCEDURE `ConsultaLogRondaExtra`()  BEGIN  SELECT \* FROM logrondaextra  END   1. Nome SP: ConsultaLogRondaPlaneada   -- Consulta tabela logrondaplaneada  CREATE DEFINER=`root`@`localhost` PROCEDURE `ConsultaLogRondaPlaneada`()  BEGIN  SELECT \* FROM logrondaplaneada  END   1. Nome SP: ConsultaLogSistema   -- Consulta tabela logsistema  CREATE DEFINER=`root`@`localhost` PROCEDURE `ConsultaLogSistema`()  BEGIN  SELECT \* FROM logsistema  END   1. Nome SP: ConsultaLogMedicoes   -- Consulta tabela logmedicoes  CREATE DEFINER=`root`@`localhost` PROCEDURE `ConsultaLogMedicoes`()  BEGIN  SELECT \* FROM logmedicoes;  END   1. Nome SP: ExportarLogs   -- Exporta logs: um ficheiro para cada tabela de logs. A diretoria pode e deve ser configurada dependendo do ficheiro batch que depois importará tal ficheiro.  CREATE DEFINER=`root`@`localhost` PROCEDURE `ExportarLogs`()  BEGIN  SELECT 'id','DataOperacao','Emailutilizador','Operacao','EmailAnterior','EmailNovo','NomeAnterior','NomeNovo','TipoAnterior','TipoNovo','MoradaAnterior','MoradaNova' UNION  SELECT \* FROM logutilizador  WHERE logutilizador.DataOperacao BETWEEN date\_sub(now(), interval 1 day) AND now()  INTO OUTFILE 'C:\\xampp\\mysql\\bin\\logutilizador.csv'  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n';  SELECT 'id','DataOperacao','Emailutilizador','Operacao','LimiteTemperaturaAnterior','LimiteTemperaturaNovo','LimiteHumidadeAnterior','LimiteHumidadeNovo','LimiteLuminosidadeAnterior','LimiteLuminosidadeNovo' UNION  SELECT \* FROM logsistema  WHERE logsistema.DataOperacao BETWEEN date\_sub(now(), interval 1 day) AND now()  INTO OUTFILE 'C:\\xampp\\mysql\\bin\\logsistema.csv'  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n';  SELECT 'id','DataOperacao','Emailutilizador','Operacao','EmailAnterior','EmailNovo','HoraInicioAnterior','HoraInicioNova','HoraFimAnterior','HoraFimNova','DataAnterior','DataNova' UNION  SELECT \* FROM logrondaextra  WHERE logrondaextra.DataOperacao BETWEEN date\_sub(now(), interval 1 day) AND now()  INTO OUTFILE 'C:\\xampp\\mysql\\bin\\logrondaextra.csv'  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n';  SELECT 'id','DataOperacao','Emailutilizador','Operacao','EmailAnterior','EmailNovo','DataAnterior','DataNova','HoraAnterior','HoraNova' UNION  SELECT \* FROM logrondaplaneada  INTO OUTFILE 'C:\\Program Files\\XAMPP\\mysql\\bin\\logrondaplaneada.csv'  INTO OUTFILE 'C:\\xampp\\mysql\\bin\\logrondaplaneada.csv'  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n';  SELECT 'id','DataOperacao','Emailutilizador','Operacao','idMedicaoAnterior','idMedicaoNova','ValorMedicaoAnterior','ValorMedicaoNova','TipoSensorAnterior','TipoSensorNovo','DataHoraMedicaoAnterior','DataHoraMedicaoNova' UNION  SELECT \* FROM logmedicoes  WHERE logmedicoes.DataOperacao BETWEEN date\_sub(now(), interval 1 day) AND now()  INTO OUTFILE 'C:\\xampp\\mysql\\bin\\logmedicoes.csv'  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n';  END |

## Eventos de suporte à migração de dados

<Nesta secção deverá ser indicados os eventos relevantes para o processo de migração.>

|  |  |  |
| --- | --- | --- |
| Nome Evento | Local Execução  (Origem ou Destino, ou Sistema Operativo) | Muito breve descrição |
| Criação de Ficheiro de Migração | Sistema Operativo | Imperativamente deverá ser criado diariamente um ficheiro (sugerimos um ficheiro CSV), que conterá a informação do dia a reportar, num directório próprio. Sugerimos a criação de um ficheiro batch que correrá este evento, assim como o evento “Exportação de Informação”. |
| Exportação de Informação | BD Origem | Após a criação do ficheiro, deverão ser exportados todos os registos criados nas Tabelas de Logs do dia anterior até àquela hora.  Sugerimos que cada linha de registo indique de qual tabela retirou a informação.  Aqui também deverá indicar o número de registos que foram inseridos.  Depois de terminar a escrita dos registos, solicita o evento ”Importação de Informação”. |
| Importação de Informação | BD Destino | A Base de Dados de Destino deverá verificar a existência de um ficheiro num determinado caminho, de onde deverá importar os registos para cada uma das tabelas de Logs.  Se o ficheiro não existir, deverá solicitar o evento “Criação de Ficheiro”. Depois de terminar a escrita dos registos nas tabelas, solicita o evento ”Eliminação de Ficheiro”. |
| Eliminação de Ficheiro | Sistema Operativo | Deverá ser executado um programa que irá eliminar o ficheiro criado. Após este passo, poderá verificar se os registos estão iguais nas diferentes bases de dados. |

### Apreciação Crítica de Eventos

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Qualidade (Fraca, Razoável, Boa ou Muito Boa): Fraca  Breve Justificação:  A informação é contraditória, não se entende quem é que é responsável pela iniciação da migração, o sistema operativo ou a base de dados de destino ou a base de dados de origem.  Como não encontramos maneira de importar um ficheiro com várias tabelas para várias tabelas no SQL, optamos por atribuir um ficheiro .csv por tabela  **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) | | Criação de Ficheiro de Migração |  | X |  |  | | Exportação de Informação |  | X |  |  | | Importação de Informação |  | X |  |  | | Eliminação de Ficheiro | X |  |  |  | |

### Eventos Implementados

|  |
| --- |
| 1. Nome Evento: Criação do Ficheiro, exportação e delete do mesmo  // Exporta os ficheiros e pede a base de dados do auditor para importar os ficheiros, caso os ficheiros não existirem, exportará outra vez e voltara a tentar importar. (Ficheiro .bat)  @echo off  :start  goto :export  :export  mysql -u root -p -P 29999 "bdorigem" -e "CALL ExportarLogs();"  goto :fileassurance  :fileassurance  if exist logutilizador.csv (  if exist logrondaextra.csv (  if exist logrondaplaneada.csv (  if exist logsistema.csv (  if exist logmedicoes.csv (  goto :import  )else goto :export  )else goto :export  )else goto :export  )else goto :export  )else goto :export  :import  mysql -uroot -p -P 29999 --local-infile "bddestino" < importaux.sql  goto :delete  :delete  del \*.csv  goto :end  :end  echo Migration Complete  2. Nome Evento: Importação do ficheiro na base de dados do auditor  *-- comando SQL para ser corrido na base de dados do auditor para importar. As diretorias dos ficheiros são para serem configuradas.*  LOAD DATA INFILE 'C:\\xampp\\mysql\\bin\\logutilizador.csv' INTO  TABLE logutilizador  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n'  IGNORE 1 ROWS;  LOAD DATA INFILE 'C:\\xampp\\mysql\\bin\\logsistema.csv' INTO  TABLE logsistema  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n'  IGNORE 1 ROWS;  LOAD DATA INFILE 'C:\\xampp\\mysql\\bin\\logrondaextra.csv' INTO  TABLE logrondaextra  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n'  IGNORE 1 ROWS;  LOAD DATA INFILE 'C:\\xampp\\mysql\\bin\\logrondaplaneada.csv' INTO  TABLE logrondaplaneada  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n'  IGNORE 1 ROWS;  LOAD DATA INFILE 'C:\\xampp\\mysql\\bin\\logmedicoes.csv' INTO  TABLE logmedicoes  FIELDS TERMINATED BY ','  ENCLOSED BY '"'  LINES TERMINATED BY '\n'  IGNORE 1 ROWS; |

## 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

As especificações, no geral, estavam pouco rigorosas e robustas: Há uma falta muito grande de triggers essenciais para os logs funcionarem; Há um excesso de SPs sem razão nenhuma; O modelo relacional da BD utiliza o email do utilizador como primary key, facilitando ataques; Há também problemas na migração caso a base de dados do auditor seja apagada, a migração é sempre feita diariamente. Para além disso, também houve uma falta de clareza na especificação em algumas partes: Eventos da migração.

Por outro lado, a especificação teve um modelo relacional bastante completo e também uma definição de roles e management de utilizadores adequado.

**Avaliação Global da Qualidade das Especificações recebidas**

|  |
| --- |
| Avaliação (A,B,C,D,E) : C  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**

|  |
| --- |
| O processo de migração é confuso e pouco resistente a falhas. |
| O excesso de Stored Procedures especificados. |
| Não é garantida a criação de logs para todas as tabelas. |

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

# Etapa C (Especificação e Implementação do Próprio Grupo)

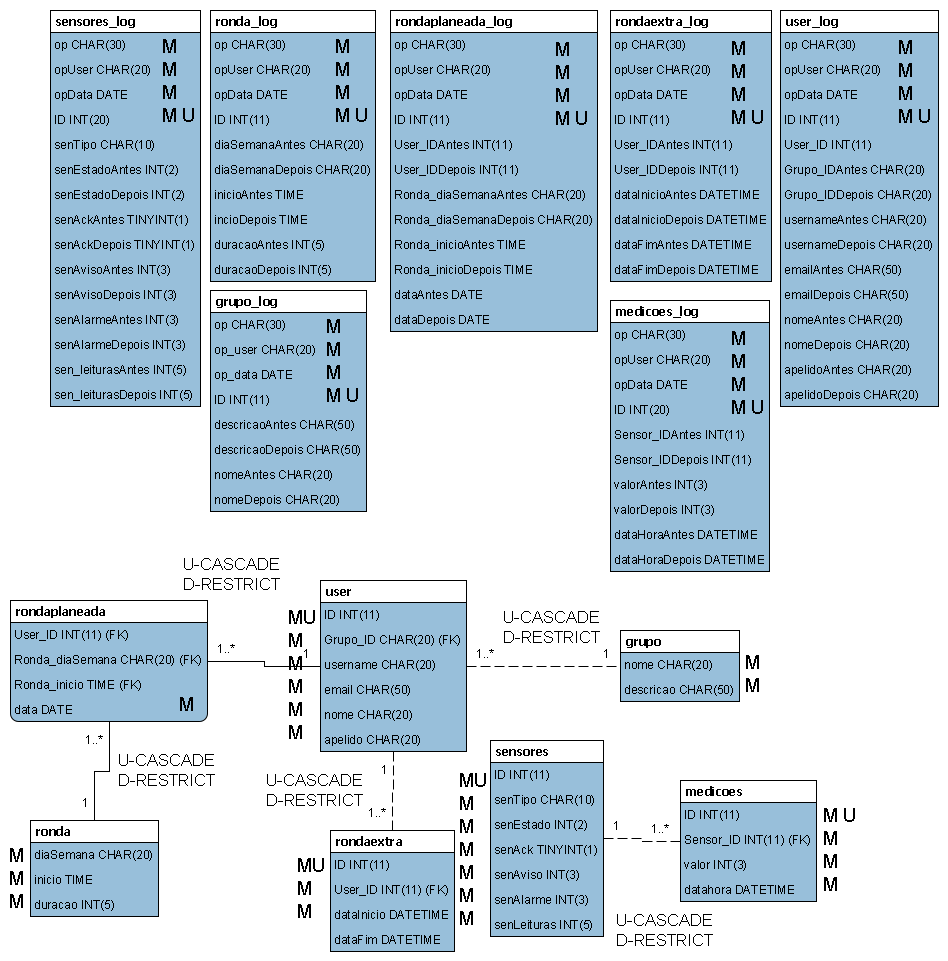
## Especificação do Próprio Grupo (versão compactada)

<Não é necessário colocar literalmente toda a especificação entregue ao outro grupo. Mas tem de estar a informação necessária para se perceber as autocríticas (o que foi mudado e porquê) e as componentes técnicas que suportam a implementação.>.

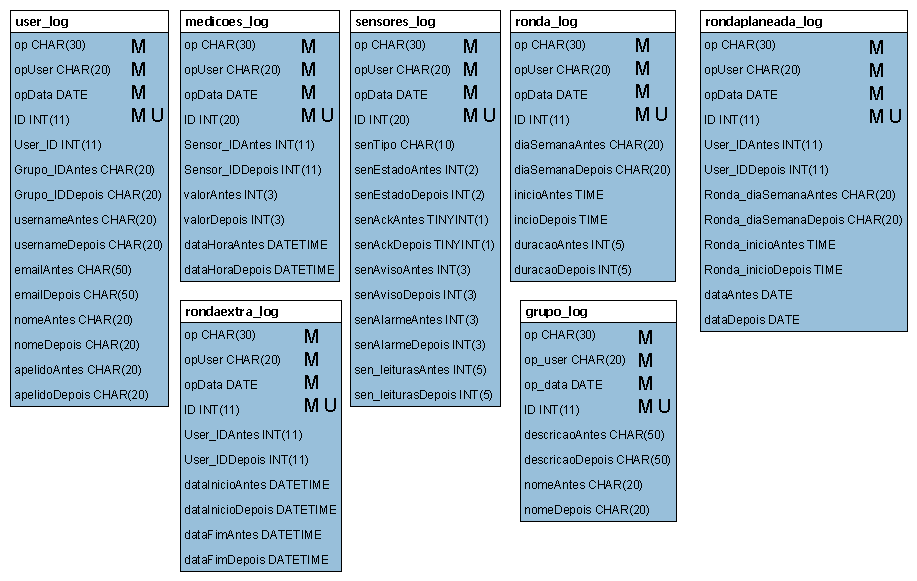
### Especificação do Esquema relacional da base de Dados Origem

- Na tabela grupos foi removido o campo ID e a chave primária passou a ser o nome do grupo. Tal facilita na atribuição dos utilizadores aos grupos.

- Na tabela users foi int



### Especificação do Esquema relacional da base de Dados Destino



### Forma de Migração Especificada

### Especificação de Utilizadores

### Triggers de suporte à gestão de logs e migração

### Stored Procedures de suporte à gestão de logs e migração

### Eventos de suporte à migração de dados especificados

### PHP de suporte à migração de dados especificado

## Avaliação Global da Qualidade das Especificações do próprio grupo

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

**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 |  |  |  |  | |

## Implementação do Próprio Grupo

### Utilizadores implementados Base de Dados Origem

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

### Utilizadores implementados Base de Dados Destino

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **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)**     |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | Base de Dados (O/D) | 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: \_\_\_\_\_ (Base de Dados: )  //*Breve Descrição*  *Código*  2. Nome Trigger: \_\_\_\_\_ (Base de Dados: )  //*Breve Descrição*  *Código*  3. Nome Trigger: \_\_\_\_\_ (Base de Dados: )  //*Breve Descrição*  *Código* |

### Lista de Stored Procedures

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Lista de SP (para cada SP assinalar com x em célula correspondente)**     |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | Base de Dados (O/D) | 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: \_\_\_\_\_ (Base de Dados: )  //*Breve Descrição*  *Código*  2. Nome SP: \_\_\_\_\_ (Base de Dados: )  //*Breve Descrição*  *Código*  3. Nome SP: \_\_\_\_\_ (Base de Dados: )  //*Breve Descrição*  *Código* |

### Lista Eventos

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Lista de Eventos (para cada evento assinalar com x em célula correspondente)**     |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | Base de Dados (O/D) | 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: \_\_\_\_\_ (Base de Dados: )  //*Breve Descrição*  *Código*  2. Nome Evento: \_\_\_\_\_ (Base de Dados: )  //*Breve Descrição*  *Código*  3. Nome Evento: \_\_\_\_\_ (Base de Dados: )  //*Breve Descrição*  *Código* |

### PHP Implementado

|  |
| --- |
| *Código* |

# 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

A segurança na migração por ficheiro e pior uma vez que exporta ficheiros das tabelas, abrindo mais possibilidades para ataques. Para além disso, também pode ser possível ter credenciais no ficheiro .bat (que depois executará a migração) que poderão ser uma falha de segurança.