



# Segurança de dados

## Aula 09 – Automação de processos

---

Gustavo Bianchi Maia

[gustavo.maia@faculdadeimpacta.com.br](mailto:gustavo.maia@faculdadeimpacta.com.br)

# Sumário

---

- Revisão de Backups e Restore
- Correção Exercícios - Prévia AC3 - parte 1
- Processos automatizáveis
- Exercícios - Adaptação códigos
- Introdução ao SQL Server Agent
  - Jobs
  - Steps
  - Scheduler
  - Operadores / alertas
- Exercícios - Prévia AC3 - parte 2

# Backups ou Cópias de segurança

Os backups ( ou cópias de segurança ), são a cópia de dados de um dispositivo para outro que possam ser restaurados em caso de perda dos dados originais.

Cópias de segurança são geralmente confundidas com arquivos e sistemas tolerantes a falhas. Diferem de arquivos (MDF, LDF) pois enquanto arquivos são cópias primárias dos dados, cópias de segurança são cópias secundárias dos dados. Diferem de sistemas tolerantes a falhas pois cópias de segurança assumem que a falha causará a perda dos dados, enquanto sistemas tolerantes a falhas assumem que a falha não causará.

As cópias de segurança devem obedecer vários parâmetros, tais como: o tempo de execução, a periodicidade, a quantidade de exemplares das cópias armazenadas, o tempo que as cópias devem ser mantidas, a capacidade de armazenamento, o método de rotatividade entre os dispositivos, a compressão e criptografia dos dados

Fonte: <http://pt.wikipedia.org/wiki/Backup>

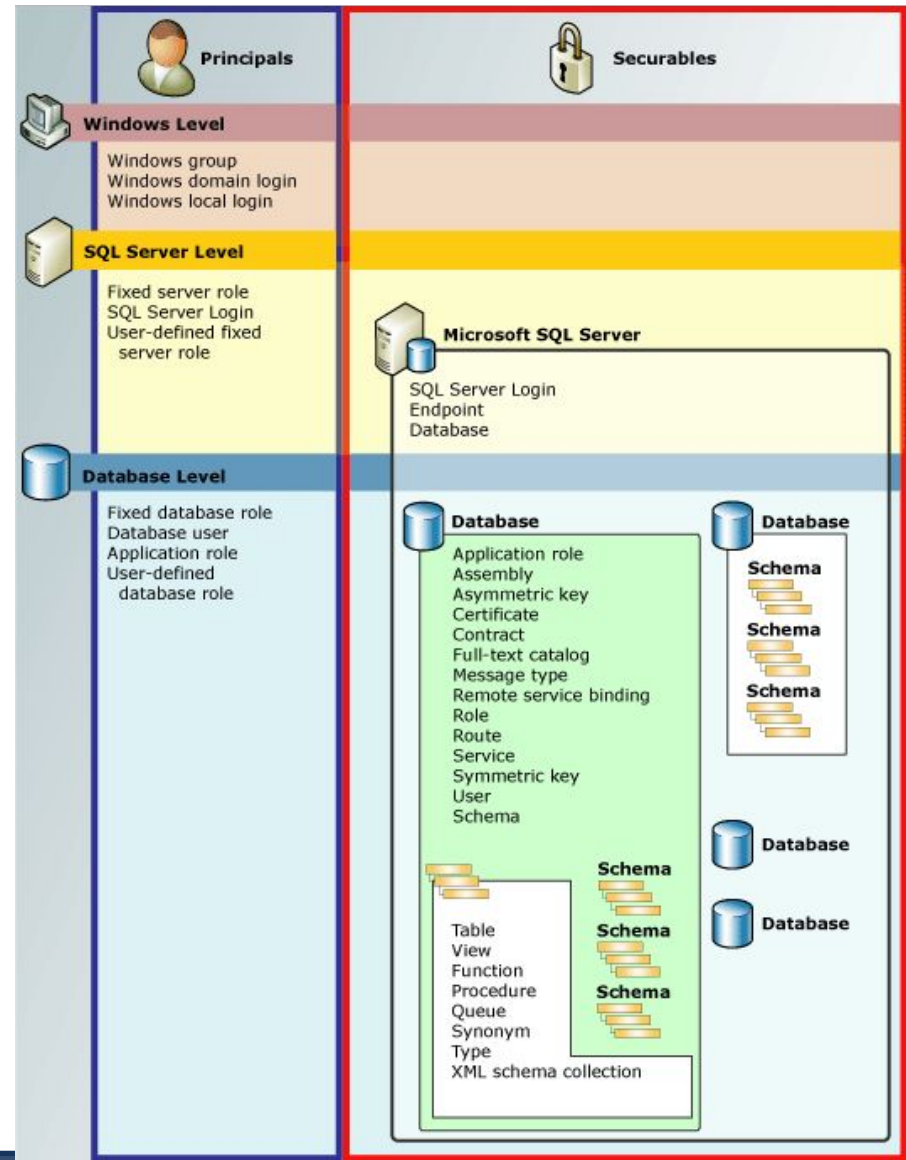
# Database backups vs database scripting

Backups de bancos de dados ( ou database backups ), **garantem** não apenas a **integridade dos dados** ( PK → FK ), como todo o esquema de permissões internas àquele banco.

Database Backups em bancos relacionais que seguem as regras do **ACID**, ainda garantem **transparência** para todos os demais usuários enquanto o backup estiver sendo realizado.

**Tudo o que está 'fora' do banco de dados não é levado junto de um database backup**, são exemplos: logins, server roles, server objects, SQL Agent Jobs, linked servers, entre outros.

Database scripting é a técnica que alguns bancos usam para gerar os dados via Script ( ou DUMPs ) sem todas as garantias citadas acima.



# Tipos de Backups

---

Backups que mantêm ou garantem a integridade do banco:

## **Total / Full:**

Todo o banco de dados é copiado, é o modo mais simples e mais utilizado porém também o mais lento e pesado.

## **Diferencial / Differential:**

Apenas os dados modificados desde o último backup completo/diferencial são armazenados. ( algumas documentações traduzem o diferencial como parcial )

## **Log:**

Apenas o log transacional é copiado, ele contém todas as instruções que causaram qualquer tipo de mudança no banco desde o último backup completo/diferencial/log.

Backups que NÃO mantêm ou garantem a integridade do banco:

## **Parcial:**

Levam apenas parte dos dados, permissões, objetos ou estruturas.

(\*) muitas vezes o termo backup parcial nem é classificado como tipo de backup.

# Estratégias de Backup

As estratégias de backup variam dependendo do tamanho do banco de dados, concorrência com a atividade de backup, criticidade de operação e janela de perda.

A seguinte estratégia supõe um banco grande o suficiente para não aceitar Backups Completos todos os dias, substituindo-os por diferenciais durante a semana, mas aceitam uma janela de perda de 15 minutos.

A regra é intercalar Completo > Diferencial(\*) >>> Log

Hora / Dia	Sábado	Domingo	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo	Segunda
0:00	Diferencial	FULL	Diferencial	Diferencial	Diferencial	Diferencial	Diferencial	Diferencial	FULL	Diferencial
0:15	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG
0:30	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG
...										
23:45	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG

(\*) muitas organizações não realizam backups diferenciais, isso é possível se você realizar backups FULL constantemente

# Estratégias de Restauração

O restore deve ser realizado em ordem, das unidades maiores para as menores. Os dados devem sempre ser 'encaixados' conforme LSN ( logical sequential number ) das instruções que geraram o log transacional.

Supondo uma empresa seguindo nossa estratégia de backup proposta, onde um "CRASH" ocorreu na quarta-feira às 00:35, para recuperá-lo, você precisará:

- Backup FULL do Domingo Anterior
- Backup Diferencial de cada dia útil até 'antes' do crash
- Backup de log até o momento(\*) do crash.

Hora / Dia	Sábado	Domingo	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo	Segunda
0:00	Diferencial	FULL	Diferencial	Diferencial	Diferencial	Diferencial	Diferencial	Diferencial	FULL	Diferencial
0:15	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG
0:30	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG
0:35					CRASH					
23:45	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG	LOG

# Revisão - BACKUP

**Manuais:** [BACKUP \(Transact-SQL\) | Microsoft Docs](#)  
[BACKUP \(Transact-SQL\) - SQL Server | Microsoft Docs](#)

```
BACKUP DATABASE OU LOG consultorio --Criando um backup do banco ou do log
TO DISK = 'C:\BD\Backups\Consultorio.bak'
WITH format --opção de criar um novo cabeçalho
, init --sobrescrevendo-o caso um arquivo como mesmo nome já existir
OU noint permite a adição de vários backups no mesmo arquivo ( APPEND )
, mirror TO DISK = 'C:\BD\Backups\Consultorio_1b.bak' --cópia de espelho
, differential --Backup diferencial
, compression --Backup com compressão de dados
, copy_only --Backup sem ponto de restauração para não interferir na restauração
, norecovery --Backup deixando-o não operacional ( NORECOVERY )
, STATS = 10 --Backup com feedback a cada 10% concluídos
, encryption (algorithm = aes_256, server certificate = ...) -- Criptografia
, expiredate = '20210430' OU retaindays = 7 -- Backups que expiram
```



# Revisão - Conferência antes de um RESTORE

*--Listar os backups dentro de um arquivo*

```
RESTORE filelistonly FROM DISK = 'c:\bd\backups\impacta.bak'
```

*--Ver detalhes dos bancos*

```
EXEC Sp_helpdb
```

*--Ver detalhes de um banco específico*

```
EXEC Sp_helpdb 'impacta'
```

# Exemplos - RESTORE

**Manuais:** [RESTORE \(Transact-SQL\) - SQL Server | Microsoft Docs](#)

[Restore a Database to a New Location \(SQL Server\) - SQL Server | Microsoft Docs](#)

```
RESTORE DATABASE OU LOG impacta --Criando um restore do banco ou do Log
FROM DISK = 'c:\bd\backups\impacta.bak'
WITH recovery --Deixa o banco em modo ONLINE NÃO aceita futuros restores
OU standby = 'c:\bd\impacta.trn' --Modo STANDBY/READONLY aceita futuros restores
OU norecovery --Deixa o banco OFFLINE aceita futuros restores em continuação
, STATS = 10 --Restore com feedback a cada 10% concluídos
, REPLACE --para sobrescrever um banco já existente

--Para evitar que um banco seja restaurado sobre
-- os mesmos arquivos físicos ( MDF, LDF ) um novo caminho deve ser fornecido.
, move 'impacta' TO 'c:\bd\mdf\impacta_Clone.mdf'
, move 'impacta_log' TO 'c:\bd\mdf\impacta_Clone_log.ldf'

, FILE = 3 --Para especificar qual arquivo dentro da mídia utilizar.
, stopat = 'Apr 15, 2020 12:00 AM' --Em restore de Log, indica ponto de parada
```

# Exemplos - RESTORE

--Backup de um banco ONLINE **NÃO aceita** futuros restores em continuação

```
RESTORE DATABASE impacta
FROM DISK = 'c:\bd\backups\impacta.bak'
WITH recovery
```

--Backup de um banco OFFLINE

-- mas que **aceita** futuros restores em continuação

```
RESTORE DATABASE impacta
FROM DISK = 'c:\bd\backups\impacta.bak'
WITH standby = 'c:\bd\impacta.trn'
```

--Backup de um banco STANDBY / READONLY

-- mas que **aceita** futuros restores em continuação

```
RESTORE DATABASE impacta
FROM DISK = 'c:\bd\backups\impacta.bak'
WITH norecovery , REPLACE
```

(\*)opção **REPLACE** é utilizada para SOBRESCREVER UM BANCO JÁ EXISTENTE

# Exemplos - RESTORE

*/\*Restaurando em sequência os backups diferencial depois completo, ambos salvos no mesmo arquivo, por isso a ordem dos arquivos deve ser especificada \*/*

```
RESTORE DATABASE adventureworks2008r2
FROM DISK = 'Z:\SQLServerBackups\AdventureWorks2008R2.bak'
WITH FILE = 1 norecovery;
```

```
RESTORE DATABASE adventureworks2008r2
FROM DISK = 'Z:\SQLServerBackups\AdventureWorks2008R2.bak'
WITH FILE = 2 recovery;
```

*/\*Uma vez ONLINE, o banco não aceita mais instruções\*/*

*--Listar os backups dentro de um arquivo*

```
RESTORE filelistonly FROM DISK = 'c:\bd\backups\impacta.bak'
```

*--Ver detalhes dos bancos*

```
EXEC Sp_helpdb
```

*--Ver detalhes de um banco específico*

```
EXEC Sp_helpdb 'impacta'
```

# Exercícios - Prévia AC3 - parte 1

- Crie um Banco de dados chamado IMPACTA, em modo de recuperação cheio ( FULL ), com arquivo físico ( MDF ) localizado em C:\BD\MDF e arquivo de logs ( LDF ) em C:\BD\LDF.

```
USE master
go
CREATE DATABASE [IMPACTA]
ON PRIMARY ( NAME = N'IMPACTA'
, filename = N'C:\BD\MDF\IMPACTA.MDF'
, size = 8192kb
, filegrowth = 65536kb )
log ON ( NAME = N'IMPACTA_LOG'
, filename = N'C:\BD\LDF\IMPACTA_LOG.LDF'
, size = 8192kb
, filegrowth = 65536kb )
```

go

*--forçando o modo de recuperação cheio ( FULL ) - OPCIONAL pois FULL é o padrão*

```
ALTER DATABASE [IMPACTA] SET recovery FULL
```

go

# Exercícios - Prévia AC3 - parte 1

- Popule este banco com alguma tabela e alguns dados ( apenas para que ele não fique em branco ).

```
USE impacta;
```

```
go
```

```
CREATE TABLE alunos
```

```
(
```

```
    id    INT PRIMARY KEY IDENTITY,
```

```
    nome  VARCHAR(45)
```

```
)
```

```
INSERT INTO alunos
```

```
VALUES      ('FULANO'),
```

```
            ('CICLANO'),
```

```
            ('BELTRANO')
```

```
go
```

```
SELECT * FROM    impacta.dbo.alunos
```

```
go
```

# Exercícios - Prévia AC3 - parte 1

- Realize o BACKUP FULL deste banco e armazene-o em C:\BD\Backups ( de o nome que achar melhor).

```
BACKUP DATABASE [IMPACTA]
    TO DISK = 'C:\BD\Backups\BKP_IMPACTA.BAK'
WITH format
    , init;
go
```

# Exercícios - Prévia AC3 - parte 1

- Realize o RESTORE deste banco no mesmo servidor em modo NORECOVERY, com nome IMPACTA\_HA ( lembre-se de alterar o destino dos arquivos físicos ).

```
USE master
```

```
go
```

```
RESTORE DATABASE impacta_ha
```

```
FROM DISK = 'C:\BD\Backups\BKP_IMPACTA.BAK'
```

```
WITH norecovery
```

```
    , move 'IMPACTA' TO 'C:\BD\MDF\IMPACTA_HA.MDF'
```

```
    , move 'IMPACTA_LOG' TO 'C:\BD\LDF\IMPACTA_HA_LOG.LDF'
```

```
go
```



# Exercícios - Prévia AC3 - parte 1

- Faça alguma alteração no banco de dados IMPACTA ( como inserir novas linhas na tabela de testes ).

```
INSERT INTO impacta.dbo.alunos
VALUES      ('FULANA'),
            ('CICLANA'),
            ('BELTRANA')
```

go

*--Neste momento os bancos estão diferentes.*

```
SELECT * FROM    impacta.dbo.alunos
SELECT * FROM    impacta_ha.dbo.alunos
```

go

# Exercícios - Prévia AC3 - parte 1

---

- Faça um **BACKUP de LOG** do banco IMPACTA e também armazene-o em C:\BD\Backups.

```
USE master
```

```
go
```

```
BACKUP log [IMPACTA]
```

```
TO DISK = 'C:\BD\Backups\IMPACTA_LOG.LOG'
```

```
WITH init
```

```
go
```

# Exercícios - Prévia AC3 - parte 1

- Realize o **RESTORE do LOG** em **modo STANDBY** ( aponte o arquivo de standby para C:\BD\TRN ), do backup do LOG DO BANCO IMPACTA no banco IMPACTA\_HA.

```
USE master
```

```
go
```

```
RESTORE log [IMPACTA_HA]
```

```
FROM DISK = 'C:\BD\Backups\IMPACTA_LOG.LOG'
```

```
WITH standby = 'C:\BD\TRN\IMPACTA.TRN'
```

```
go
```

# Exercícios - Prévia AC3 - parte 1

- **Confira**, executando um SELECT no banco IMPACTA\_HA, se as informações adicionais criadas no banco IMPACTA estão presentes no banco IMPACTA\_HA.

*--Neste momento os bancos voltaram a ficar iguais.*

```
SELECT * FROM impacta.dbo.alunos
```

```
SELECT * FROM impacta_ha.dbo.alunos
```

# Exercícios - Prévia AC3 - parte 1

Extra, pensem em como resolver o seguinte problema:

Agora, você tem que armazenar N backups de LOG ( gerados de 5 em 5 mins ) e em algum momento futuro, você pode/deve restaurá-los, tais backups não podem se sobrescrever nem tampouco posso perder um deles.

O processo deve ser todo automático, sem intervenção humana.

*--Declaro variável*

```
DECLARE @NOME_BACKUP VARCHAR(255)
```

*--Gero um nome aleatório baseado na data*

```
SELECT @NOME_BACKUP = 'C:\BD\BACKUPS\IMPACTA_'
    + CONVERT(VARCHAR(max), Getdate(), 112)
    + LEFT(Replace(Replace(CONVERT(VARCHAR(max), CONVERT(TIME,
        Getdate()))), ':', ''), '.', ''), 6)
    + '.LOG'
```

*--imprimo o nome só para conferência*

```
SELECT @NOME_BACKUP -- C:\BD\BACKUPS\IMPACTA_20210426144143.LOG
```

*--Realizo o backup cujo destino será o caminho definido na variável.*

```
BACKUP LOG impacta TO DISK = @NOME_BACKUP WITH init;
```

---

# **Automatização de processos**

# Automatização de processos

---

- Muitos processos, em especial os associados à manutenção ou administração do servidor, devem ser realizados em momentos de baixa atividade do servidor ou em janelas de manutenção específicas ( madrugadas, finais de semana, feriados ).
- Não é sempre que um DBA precisa, ou deve, ficar acordado para a realização destas atividades, então, utiliza-se de agendadores ( Schedulers ) para executar tais scripts nos horários específicos.
- Existem serviços que realizam estes agendamentos, como:
  - Cronstab - [Crontab – Wikipédia, a enciclopédia livre \(wikipedia.org\)](https://pt.wikipedia.org/wiki/Crontab)
  - Windows Task Scheduler - [Windows Task Scheduler - Wikipedia](https://pt.wikipedia.org/wiki/Windows_Task_Scheduler)
  - SQL Server Agent - [SQL Server Agent - SQL Server Agent | Microsoft Docs](https://docs.microsoft.com/pt-br/sql/ssas/sql-server-agent-overview)

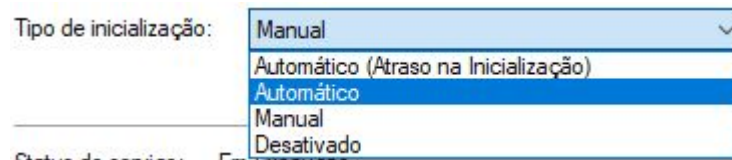
Nós utilizaremos o SQL Server Agent, porém, aqueles que utilizam apenas o SQL Server na edição EXPRESS só tem à sua disposição o Windows Task Scheduler.

# SQL Server Agent

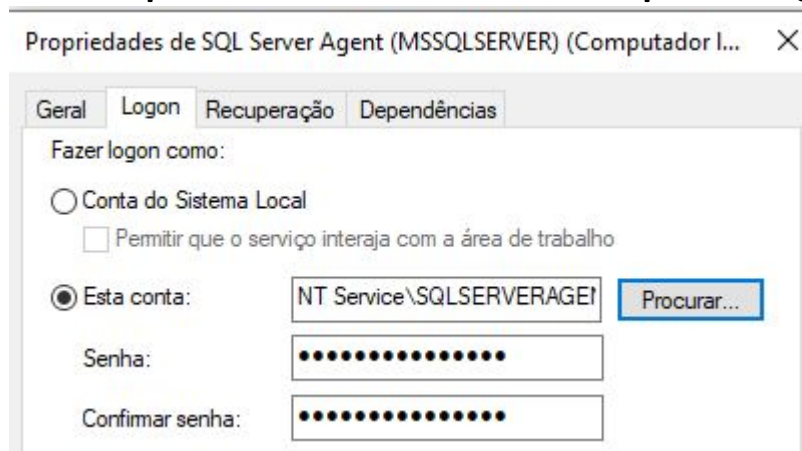
O SQL Server Agent é um serviço e portanto, deve estar iniciado.

Nome	Descrição	Status
SQL Server (MSSQLSERVER)	Fornecer armazenamento, processamento...	Em Execução
SQL Server Agent (MSSQLSERVER)	Executa trabalhos, monitora o SQL Server,...	Em Execução
SQL Server Browser	Fornecer informações a computadores cli...	

É recomendado deixá-lo em modo automático, para que ele sempre seja iniciado junto com o SQL.



Deve ser iniciado por uma conta com privilégios suficientes

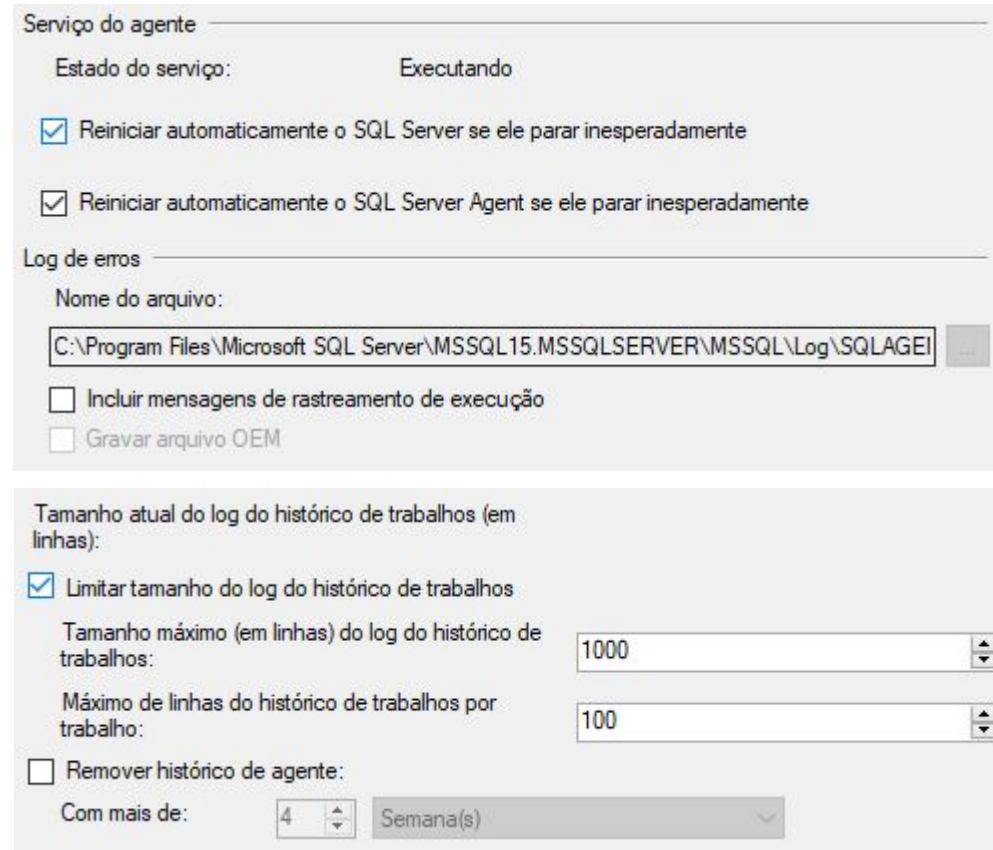
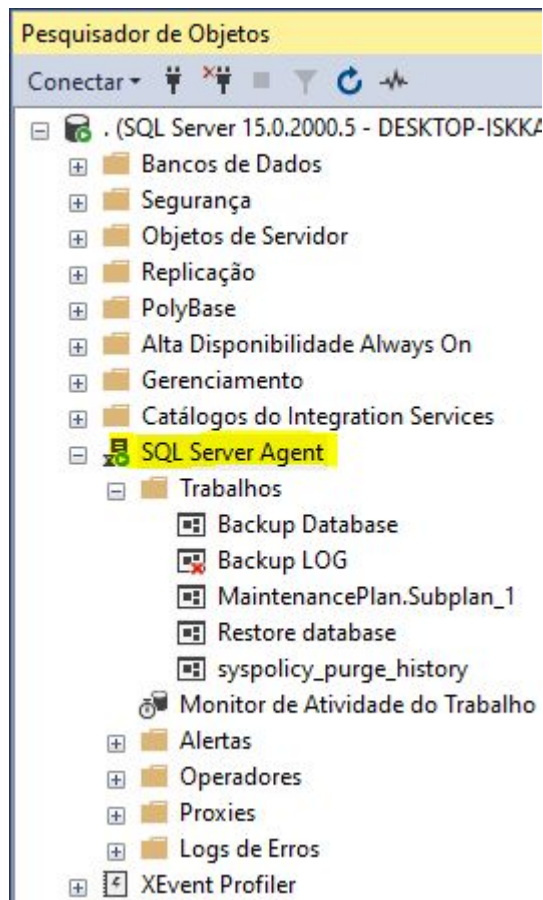




# SQL Server Agent

Está localizado no pesquisador de objetos ( Object Explorer )

E possui diversas configurações ( botão direito > Propriedades )



# SQL Server Agent

Antes de criar seus agendamentos, alguns conceitos devem ser entendidos:

- Job / Tarefa / Trabalho → nome dado à atividade
- Step / Passo / Etapa → 1 Trabalho conterá N etapas
  - Type / Tipo → toda etapa será um tipo de script
  - Run as / Executar como → e usará estas permissões
  - Database / Banco de dados → se conectando neste banco
  - On success / Ação ao obter êxito → continuação se OK
  - On failure / Ação ao falhar → continuação se falha
- Scheduler / Agendamento / Agendas → horários de execução
  - Schedule Type / Tipo → tipo de agendamento
  - Frequency → frequência ( diário, semanal, ...)
  - Daily Frequency → frequência no dia da execução
  - Duration → fim das execuções

Acompanhe a demonstração do professor no SSMS.

# SQL Server Agent

Os Jobs podem ser executados, parados, habilitados ou desabilitados pela interface ( botão direito sobre o nome do job );

The screenshot displays the SQL Server Agent interface. On the left, the 'Trabalhos' (Jobs) folder is expanded, showing a list of jobs including 'Backup Database', 'Backup LOG', 'Maintenanc...', 'Restore data', and 'syspolicy\_p...'. A right-click context menu is open over the 'Backup LOG' job, showing options such as 'Novo Trabalho...', 'Iniciar Trabalho na Etapa...', 'Parar Trabalho', 'Script de Trabalho como...', 'Exibir Histórico', 'Habilitar', 'Desabilitar', 'Iniciar PowerShell', 'Relatórios', 'Renomear', 'Excluir', 'Atualizar', and 'Propriedades'. The 'Exibir Histórico' option is highlighted.

On the right, the 'Visualizador do Arquivo de Log' (Log File Viewer) window is open, displaying the log history for the selected job. The 'Secionar logs' (Select logs) section shows the 'Histórico do Trabalho' (Job History) selected. The 'Resumo do arquivo de log' (Log file summary) section shows a table with columns: Data, ID da Etapa, Servidor, Nome do Trabalho, and N. The table contains three rows of log entries, all with a status of 'Sucesso' (Success).

Data	ID da Etapa	Servidor	Nome do Trabalho	N
23/09/2020 22:29:00	1	DESKTOP-ISKKAB8	Backup LOG	
23/09/2020 22:28:00	1	DESKTOP-ISKKAB8	Backup LOG	
23/09/2020 22:28:00	1	DESKTOP-ISKKAB8	Backup LOG	

The 'Detalhes da linha selecionada' (Selected line details) section shows the following information:

- Data: 23/09/2020 22:29:00
- Log: Histórico do Trabalho (Backup LOG)
- ID da Etapa: 1
- Servidor: DESKTOP-ISKKAB8
- Nome do Trabalho: Backup LOG
- Nome da Etapa: Backup LOG
- Duração: 00:00:00
- Severidade do Sql: 0
- ID da Mensagem Sql: 3014
- Operador Notificado por Email: [SQLSTATE 01000] (Mensagem 4035) BACKUP LOG processou com êxito 1 páginas em 0.002 segundos (1.220 MB/s).
- Operador Notificado por Envio de Rede: [SQLSTATE 01000] (Mensagem 3014). A etapa foi bem-sucedida.
- Operador Notificado por Pager: 0
- Repetições Tentadas: 0

The 'Status' section shows the 'Última Atualização' (Last update) as 26/04/2021 16:21:18.

No menu Exibir Histórico ( Show History ), é possível conferir o histórico de execuções, os sucessos e falhas; ao expandir e clicar sobre a etapa, pode-se inclusive ver em detalhes toda a mensagem de retorno.

# SQL Server Agent - Meu primeiro JOB

Agora siga as orientações do professor para criar um JOB, que deverá ser executado a cada 5 mins, para realizar o BACKUP do LOG do banco IMPACTA, segundo as instruções do exercício extra passado. Não esqueçam de conferir ( na pasta C:\BD\BACKUPS )

Nome da etapa:

Backup LOG

Tipo:

Script Transact-SQL (T-SQL)

Executar como:

Banco de Dados: master

Comando:

```
--Declaro variável
DECLARE @NOME_BACKUP VARCHAR(255)
--Gero um nome aleatório baseado na data
SELECT @NOME_BACKUP = 'C:\BD\BACKUPS\IMPACTA_'
+ CONVERT(VARCHAR(max), Getdate(), 112)
+ LEFT(Replace(Replace(CONVERT(VARCHAR(max), CC
Getdate()), '.', ''), ':', ''), 6)
+ '.LOG'
--imprimo o nome só para conferência
SELECT @NOME_BACKUP -- C:\BD\BACKUPS\IMPACTA_20210426
--Realizo o backup cujo destino será o caminho definido na variável.
BACKUP LOG impacta TO DISK = @NOME_BACKUP WITH init;
```

Abrir...

Selecionar Tudo

Copiar

Colar

Analisar

Propriedades da Agenda de Trabalho - a cada 5 mins

Nome: a cada 5 mins

Tipo de agenda: Recorrente ☒ H

Ocorrência única

Data: 26/04/2021 Hora: 15:31:15

Frequência

Ocorre: Diariamente

Repete-se a cada: 1 dia(s)

Frequência diária

☐ Ocorre uma vez em: 00:00:00

☒ Ocorre a cada: 5 minuto(s)

Iniciando em: 00:00:00

Finalizando em: 23:59:59

---

# **Continuação - Prévia AC3 - parte 2**

# Continuação - Prévia AC3 - parte 2

Agora imaginem que, todo o processo de **BACKUP** deve ser realizado em um **servidor** chamado **PRINCIPAL** ( Banco = Principal ), e todo o processo de **RESTORE** em um **servidor** **SECUNDARIO**



**Principal**



**Principal**



**Secundario**



**Secundario**

Realize o **BACKUP FULL** do banco **PRINCIPAL** e armazene-o em C:\BD\Backups.

Realize o **RESTORE** deste banco em modo **NORECOVERY**, com nome **SECUNDARIO**

Faça um **BACKUP de LOG** do banco **PRINCIPAL** e também armazene-o em C:\BD\Backups.

Realize o **RESTORE do LOG** em **modo STANDBY** ( aponte o arquivo de standby para C:\BD\TRN ), do backup do LOG DO BANCO **PRINCIPAL** no banco **SECUNDARIO**.

**Problema 1:** Qual o planejamento da estratégia de backups /restores ( horários de execução ) ?

**Problema 2:** Quando ( e onde ) o servidor **SECUNDARIO** deve ir 'buscar' seus arquivos ou seja, como fazer para os arquivos irem de um servidor para o outro ?

**Agora é com vocês: Crie os JOBS, agende-os se necessário, e garanta que o processo está rodando.**

# Obrigado

---



## Segurança de dados

### Aula 09 – Automatização de processos

---

**Gustavo Bianchi Maia**  
[gustavo.maia@faculdadeimpacta.com.br](mailto:gustavo.maia@faculdadeimpacta.com.br)