Banco de Dados - Firebird

Definindo o hardware para o Firebird

Este tutorial tem o objetivo de relatar o funcionamento do *hardware* em relação ao Firebird, para servir como guia na definição de arquitetura para um servidor de banco de dados.

> “Basicamente o sistema gerenciador de banco de dados Firebird utiliza 3 *hardwares* de um computador, são eles:

1) Processador

É responsável por executar uma tarefa, por exemplo “Buscar os dados do cadastro de clientes”.

Basicamente quanto maior for a velocidade de processamento (GHz) mas rápido uma determinada tarefa será processada, e quanto mais núcleos um processador tiver, mais tarefas simultâneas poderá executar, ou seja, irá atender mais rapidamente às solicitações recebidas.

O Firebird trabalha diferente nas suas arquiteturas, quanto a capacidade de executar tarefas simultâneas, ou seja, usar mais de 1 núcleo, vamos entender melhor isso nas características de cada arquitetura.

2) Memória RAM

É a memória primária, de acesso rápido, que não guarda informações, apenas as mantêm enquanto o computador estiver ligado, ou até que o sistema operacional faça a reciclagem (gerencie e identifique a ociosidade, liberando a informação).

O processador acessa essa memória para responder às solicitações de leitura e escrita. Caso uma informação ainda não esteja na memória RAM, ela será buscada no HD e inserida na RAM, portanto se a informação já está na memória RAM, o processador vai conseguir responder mais rapidamente a tarefa solicitada.

O Firebird mantém um “Cache (dados em memória)” por “página de dados”, dessa forma quando um usuário solicita a informação X que está na página 1, a mesma é carregada do HD para o Cache na memória RAM, dessa forma se outro usuário solicitar a informação Y que também está na página 1, o Firebird irá responder mais rapidamente, pois não precisa buscar os dados da página no disco, carregar para a memória RAM e depois fazer a leitura.

3) HD

É a memória física, responsável por guardar as informações, será acessada sempre que necessário gravar ou ler alguma informação. Também é considerada a memória secundária, pois o acesso tanto de leitura, quanto de escrita, ocorre primeiro na memória RAM e depois no HD (se necessário).

Como vimos no item anterior, uma tarefa solicitada pelo Firebird não acessa diretamente o HD, porém isso não significa que ele tenha menos importância que a memória RAM, e sse hardware também tem muita influência no desempenho, pois quanto maior for sua velocidade de leitura/escrita, mais rápido vai entregar ou gravar a informação. E também quanto mais solicitações de leitura/escrita receber, mais irá demorar para responder.

Conclusão

Levando em consideração os 3 principais *hardwares* citados acima, podemos concluir que quanto maior a capacidade, melhor o Firebird irá funcionar, porém não é bem assim na prática, o que acontece é que precisamos configurar a arquitetura correta para o ambiente, e definir o uso que o Firebird irá fazer do *hardware*, de acordo com a capacidade do mesmo e a necessidade do cliente.

Mas já podemos relacionar alguns fatores importantes na definição do *hardware* para se obter melhor desempenho:

1. Considere usar uma máquina dedicada ao banco de dados, ou seja, não tenha outros softwares instalados, não utilize essa máquina com um terminal para usuário. Esse ponto é extremamente importante e precisamos estar ciente de que os *hardwares* são gerenciados pelo sistema operacional, portanto se tivermos outros softwares instalados, e/ou usuários utilizado a máquina como um terminal, teremos concorrência de *hardware*, e o sistema operacional terá que balancear a carga de trabalho do processador, além de gerar mais operações de leitura/escrita na memória RAM e HD.
2. Utilize um sistema operacional sem ambiente gráfico, Linux por exemplo, que utiliza menos memória, e menos processamento;
3. Utilize HD com arquitetura SSD, que pode ser 10 vezes mais rápido que o HD comum, e/ou ainda RAID (verifique as diferentes formas de operação e considere usar SSD também nesse sistema);

Voltar à página inicial

Definindo a arquitetura do Firebird

Este tutorial tem o objetivo de explicar as diferentes arquiteturas do Firebird, para servir como um guia de escolha.

O Firebird vêm em duas versões, chamadas de arquiteturas: *Classic Server* e *Superserver*. Desde o Firebird 2.5, o Classic Server pode operar em dois modos: "tradicional" *Classic* e *SuperClassic*, totalizando 3 modelos. Qual deles você deve escolher? As diferenças mais importantes estão listadas abaixo. Na grande maioria dos casos, todos os três modelos funcionam igualmente bem e oferecem (quase) as mesmas possibilidades.

1) Super Server

O *Super Server* é modelo padrão de arquitetura, e nele existe apenas um *cache* de páginas (dados em memória RAM) que é compartilhado por todas as conexões, além de um único processo atender a todos os usuários.

Por ser compartilhado, este *cache* é muito eficiente. Quando vários usuários acessam as mesmas áreas do banco de dados ou quando algumas tabelas são muito mais acessadas que outras, todos os usuários se beneficiam de um *cache* grande e bem preenchido.

Por exemplo, quando o usuário **A** executa:

```
SELECT NOME FROM CLIENTES WHERE CODIGO = 1;
```

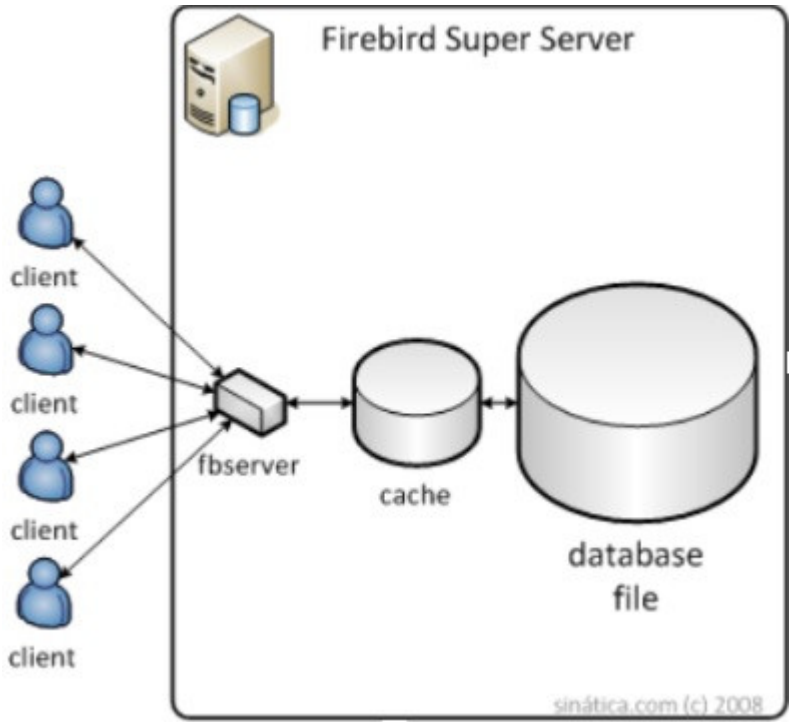
Algumas páginas relacionadas a tabela **CLIENTES** e ao índice da chave primária **CODIGO** são carregados para o *cache*.

Quando o usuário **B** executa:

```
SELECT NOME, ENDERECO, TELEFONE FROM CLIENTES WHERE CODIGO = 2;
```

A solicitação é atendida mais rapidamente, porque as páginas que este comando precisa consultar já estão no *cache*.

Veja no diagrama (considere cada “client” como um usuário do sistema):



Note também que existe apenas um único processo (*fbserver*) onde todos os usuários se conectam, e o importante aqui é que se um processo quebrar todas as conexões serão perdidas.

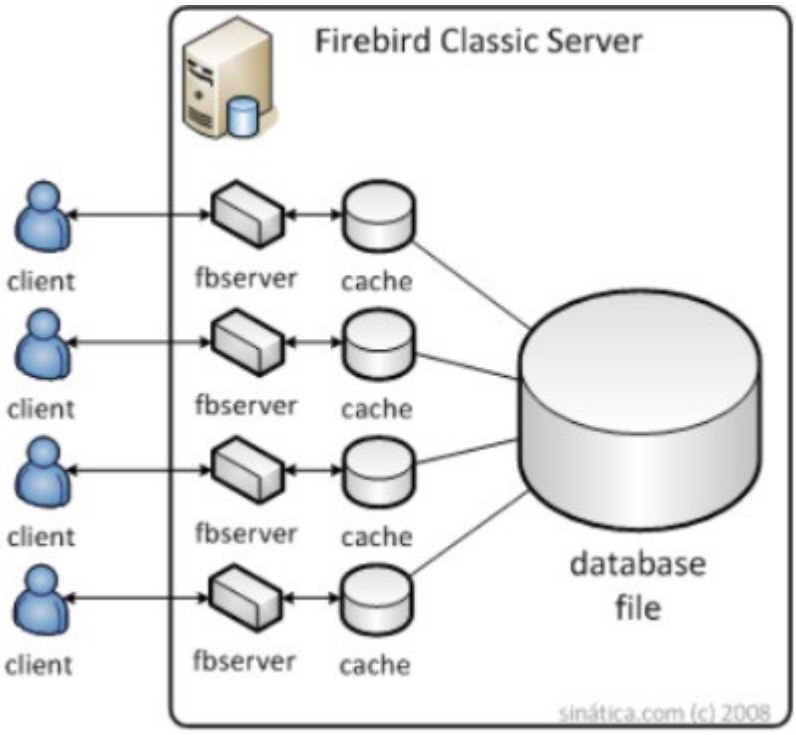
2) Classic Server

No *Classic Server*, cada usuário tem um *cache* próprio e está conectado a um processo dedicado.

O *cache* próprio é muito menos eficiente. Se dois usuário acessam a mesma área do banco de dados, esta área será copiada no *cache* de cada um deles. Usando o exemplo do item anterior, quando o usuário **B** executasse seu comando, ele não teria o benefício de um *cache* já preenchido e o Firebird precisaria acessar o disco novamente para responder.

Além do mais, a sincronização entre os *caches* é feita através do disco. Isto aumenta consideravelmente o custo de leitura/escrita em ambientes de alta concorrência.

Veja no diagrama:



Um grande benefício deste modelo é a resiliência oferecida pelos múltiplos processos. Se um deles tiver problemas, apenas o usuário conectado a ele será desconectado. Todo o restante do banco de dados continua funcionando normalmente.

O outro grande benefício é a escalabilidade. Acredito que esta característica seja a responsável por boa parte das instalações do *Classic Server*. Mesmo em casos onde o cache dedicado produz resultados inferiores ao cache compartilhado do *Super Server*, a escalabilidade compensa. Basta adicionar mais hardware e pronto, seu servidor fica mais rápido.

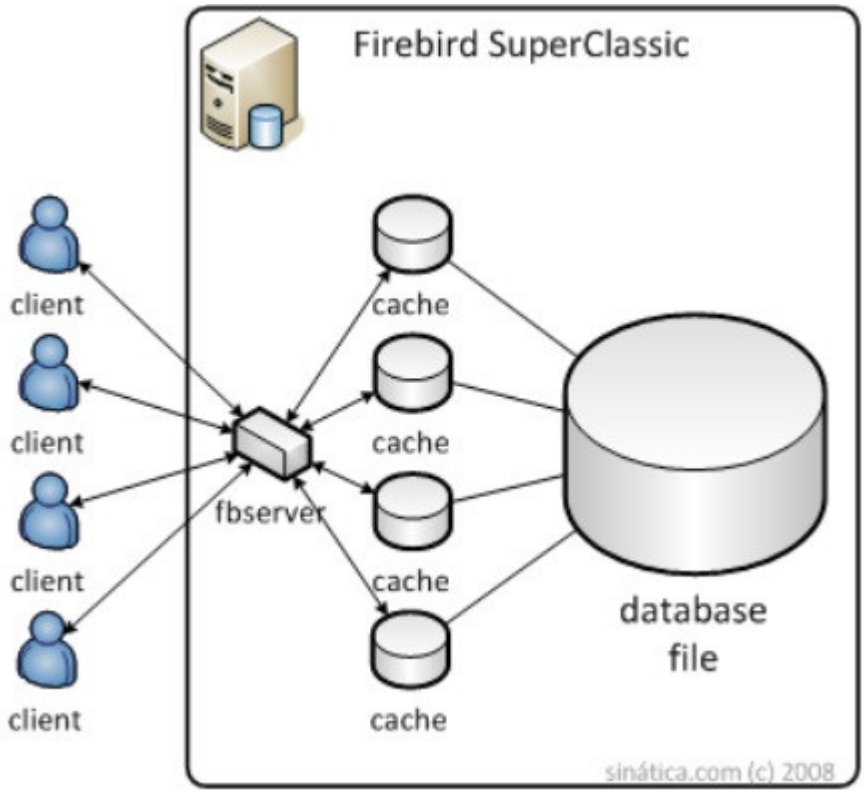
Mas esta escalabilidade não vem de graça. Imagine que você tem 200 usuários simultâneos. São 201 processos, um para cada usuário e mais um para ficar ouvindo novas conexões. Seu sistema operacional deve gerenciar todos estes processos e mantê-los em sincronia. Eles consomem muitos recursos de kernel e isto significa que ele pode ser relativamente lento. Imagine que um processo utilize +-60MB de memória, será necessário 12GB de memória disponível apenas para o Firebird.

Veja neste exemplo: Firebird 2.5 *Classic* com 7 usuários conectados. São 8 processos, 18 *threads*, 1050 identificadores.

Nome da Imagem	Identifi...	Threads
fb_inet_server.exe	133	2
fb_inet_server.exe	133	2
fb_inet_server.exe	134	2
fb_inet_server.exe	116	4
fb_inet_server.exe	135	2
fb_inet_server.exe	133	2
fb_inet_server.exe	133	2
fb_inet_server.exe	133	2

3) Super Classic

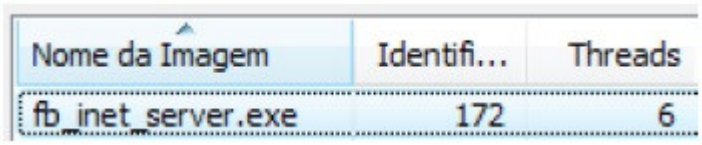
Super Classic é uma evolução e resolve o maior problema do *Classic*: todos aqueles processos pode deixar o Firebird lento e torna a manutenção mais difícil. Dessa forma o *Super Classic* opera em um único processo, assim como no *Super Server*, porém com *cache* próprio para cada usuário.



Olhando desta forma e considerando o nome, pode parecer uma arquitetura híbrida entre o *Classic Server* e *Super Server*, mas não é. O que fizeram foi colocar todos esses processos dentro de *threads*. Agora cada usuário tem uma *thread* dedicada dentro de um único processo.

Criar centenas de *threads* é muito mais barato do que criar centenas de processos e não há perda de escalabilidade. A sincronização de *cache* é feita diretamente na memória, o que reduz o custo de leitura/escrita. Outros controles que costumavam ser entre processos agora são entre *threads,* e muito mais rápidos.

Considerando o mesmo exemplo do item anterior, com 7 usuários conectados, nessa arquitetura são 1 processo, 6 *threads*, e 172 identificadores.



Nome da Imagem	Identifi...	Threads
fb_inet_server.exe	172	6

Conclusão

Esta compilação de casos mais comuns é uma sugestão e serve como guia, um ponto de partida para sua escolha. Sua implantação pode ter detalhes não contabilizados aqui.

Super Server

- Se você tiver 1 banco de dados principal e, opcionalmente, 2 a 5 bancos de dados menores (e menos carregados) no servidor, escolha a arquitetura do SuperServer e configure cada banco de dados em databases.conf;
- Servidores pequenos;
- Ambientes onde o *cache* compartilhado é mais desejável que a escalabilidade do *Super Classic*;

Classic Server

- Ambientes onde a estabilidade é a principal prioridade;
- Servidores com vários processadores e muita memória (verificar quanto cada processo utiliza em média e multiplicar pelo total de conexões);
- Bancos de dados grandes com centenas de usuários;

Super Classic

- Se você tem muitos bancos de dados no servidor (de 10 a 1000), e eles são mais ou menos iguais em termos de carga, e seu aplicativo é estável (ou seja, você nunca viu “encerramento anormal” no firebird.log), escolha SuperClassic. É melhor do que o Classic em termos de melhor controle de classificação de memória e desempenho;

- Servidores com vários processadores;
- Bancos de dados grandes com centenas de usuários;
- Ambientes onde o *cache* dedicado é mais vantajoso que o compartilhado do *Super Server*;
- Ambientes onde o *Classic Server* não se adapta mais;

Voltar à página inicial

Configurações para melhorar o desempenho

Este tutorial tem o objetivo de apresentar as configurações do Firebird, para obter um melhor desempenho.

Como existem diferentes arquiteturas de servidores, diferentes necessidades, investimentos, etc, o Firebird vem de fábrica configurado com parâmetros para atender e funcionar relativamente bem em todas as situações. Porém quando temos a necessidade de ir além, e obter um desempenho melhor do servidor de banco de dados, podemos configurar manualmente esses parâmetros de acordo com a necessidade e arquitetura.

Os parâmetros citados ficam no arquivo **firebird.conf**, que está no diretório de instalação do Firebird, e podemos abrir com um simples editor de texto e alterar os parâmetros desejados. Nesse mesmo arquivo há a explicação de cada parâmetro, então em caso de dúvidas verifique-o.

> Antes de qualquer alteração considere fazer uma cópia de *backup* do arquivo firebird.conf

Os principais parâmetros para otimização são:

- **DefaultDbCachePages** - Define o tamanho do arquivo de paginação utilizado pelo cache da máquina. Quanto maior o valor, maior a quantidade de dados que trafegam em um mesmo processo simultâneo, resultando em pesquisas mais rápidas.

- **FileSystemCacheThreshold** - Esta opção define o tamanho do arquivo de cache que o Firebird utiliza para salvar os arquivos da paginação da configuração acima.
- **FileSystemCacheSize** - Esta configuração define a porcentagem da memória o Firebird pode utilizar para o cache. Não é recomendado definir valores acima de 70 ou 80 se o servidor não for 100% dedicado ao banco de dados.
- **CpuAffinityMask** - Esta configuração aplica-se apenas a servidores com sistema operacional *Windows* e a arquitetura *Super Server*, ele define quantos "cores" do processador o Firebird pode utilizar para os processos.

> Para aplicar as configurações é necessário retirar o caractere # da frente da linha

Para cada arquitetura do Firebird são recomendados diferentes configurações, veja a seguir:

Super Server

- Configuração para servidor com processador dual-core, e pelo menos 4GB memória:

```
DefaultDbCachePages = 4096
FileSystemCacheThreshold = 67108864
FileSystemCacheSize = 70CpuAffinityMask = 3
```

- Configuração para servidor com processador dual-core, e pelo menos 8GB memória:

```
DefaultDbCachePages = 8192
FileSystemCacheThreshold = 134217728
FileSystemCacheSize = 70CpuAffinityMask = 3
```

- Configuração para servidor com processador quad-core ou superior, e acima de 8GB memória:

```
```

```
DefaultDbCachePages = 16384
FileSystemCacheThreshold = 268435456
FileSystemCacheSize = 80
Para utilizar:
2 processadores/núcleos: CpuAffinityMask = 3
3 processadores/núcleos: CpuAffinityMask = 7
4 processadores/núcleos: CpuAffinityMask = 15
5 processadores/núcleos: CpuAffinityMask = 31
6 processadores/núcleos: CpuAffinityMask = 63
7 processadores/núcleos: CpuAffinityMask = 1278 processadores/núcleos: CpuAffinityMask = 255
```

Para mais processadores, o cálculo é: **2^x-1** (Exemplo: **2^12-1 = 4095**)

Para ambas as configurações acima ainda considere os seguintes parâmetros de acordo com a versão instalada:

- Versão do Firebird 32 bits

```
TempCacheLimit = 367108864LockHashSlots = 11011
```

- Versão do Firebird 64 bits

```
TempCacheLimit = 967108864LockHashSlots = 20011
```

Super Classic

Utilização em ambientes com muitos usuários e processos, configuração para servidor com processador dual-core ou superior, e pelo menos 8GB memória:

- Versão do Firebird 32 bits

```
DefaultDbCachePages = 256
TempBlockSize = 2048576
```

```
TempCacheLimit = 27108864
LockMemSize = 3048576LockHashSlots = 20011
```

- Versão do Firebird 64 bits, servidor com poucas conexões

```
DefaultDbCachePages = 384
TempBlockSize = 2048576
TempCacheLimit = 567108864
LockMemSize = 5048576LockHashSlots = 30011
```

- Versão do Firebird 64 bits, servidor com 8GB de memória, e até 150 conexões

```
DefaultDbCachePages = 768
TempBlockSize = 2048576
TempCacheLimit = 1073741824
LockMemSize = 30971520LockHashSlots = 30011
```

- Versão do Firebird 64 bits, servidor com 16GB de memória ou mais, e até 300 conexões

```
DefaultDbCachePages = 1500
TempBlockSize = 2048576
TempCacheLimit = 2100000000
LockMemSize = 31457280LockHashSlots = 49999
```

Classic Server

- Até 8GB de memória

```
DefaultDbCachePages = 384TempBlockSize = 2048576
```

- Até 8GB de memória, versão do Firebird 32 bits

```
```

```
TempCacheLimit = 367108864
LockHashSlots = 11011LockMemSize = 5048576
```

- Até 8GB de memória, versão do Firebird 64 bits

```
TempCacheLimit = 77108864
LockHashSlots = 30011LockMemSize = 7048576
```

- 16 ou mais GB de memória, versão do Firebird 64 bits

```
DefaultDbCachePages = 1500
TempBlockSize = 2048576
TempCacheLimit = 67108864
LockMemSize = 31457280LockHashSlots = 49999
```

Após feito as alterações no arquivo, `gstat -h databasename` e verifique se os buffers de página são 0. Se estiver definido com outro valor, defina-o como 0 com o comando `gfix -buff 0.`

> **Importante!**

- Sempre renomeie e mantenha uma cópia do firebird.conf original e, para o Firebird 3.0, databases.conf;
- Para Firebird 3.0, verifique sempre o databases.conf! Deve haver os valores que você definiu explicitamente. Certifique-se de entender por que você os definiu.
- Lembre-se: as configurações do firebird.conf são aplicadas a TODOS os bancos de dados ativos no servidor como valores padrão, databases.conf (no 3.0) pode substituir o firebird.conf.
- Não se esqueça do ajuste geral do sistema operacional: consulte essa breve lista de verificação de desempenho.

Mais informações aqui.

Voltar à página inicial

Instalando múltiplos Firebird na mesma máquina

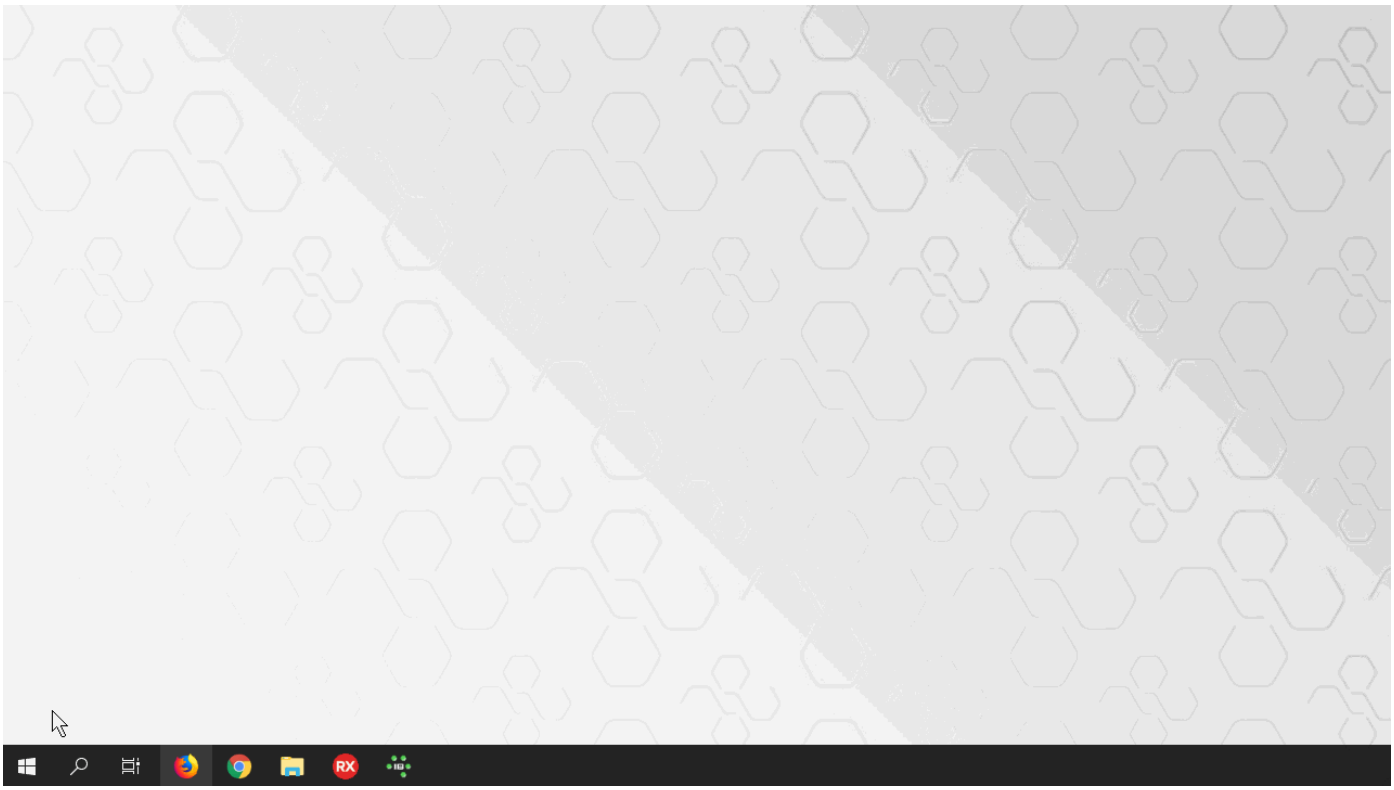
Este tutorial tem o objetivo de orientar a instalação de múltiplas instâncias do Firebird na mesma máquina/servidor.

> Geralmente essa configuração é feita quando necessita-se de versões distintas do Firebird, mas se houver a necessidade de instalar múltiplas instâncias de uma mesma versão, o procedimento será o mesmo.

Vamos considerar que já tenha instalado uma instância do Firebird na máquina, mas se ainda não tiver, faça a instalação normalmente da primeira instância, depois siga o passo-a-passo abaixo.

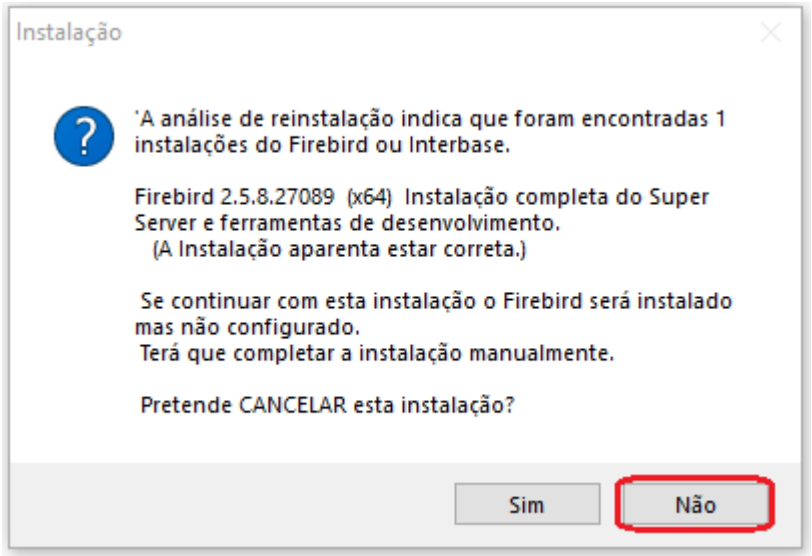
1) Parando o serviço da instância atual

Vá no menu "Iniciar" do "Windows" e digite "serviços", localize o Firebird e pare a execução, como mostra a seguir:



2) Executando o programa de instalação do Firebird

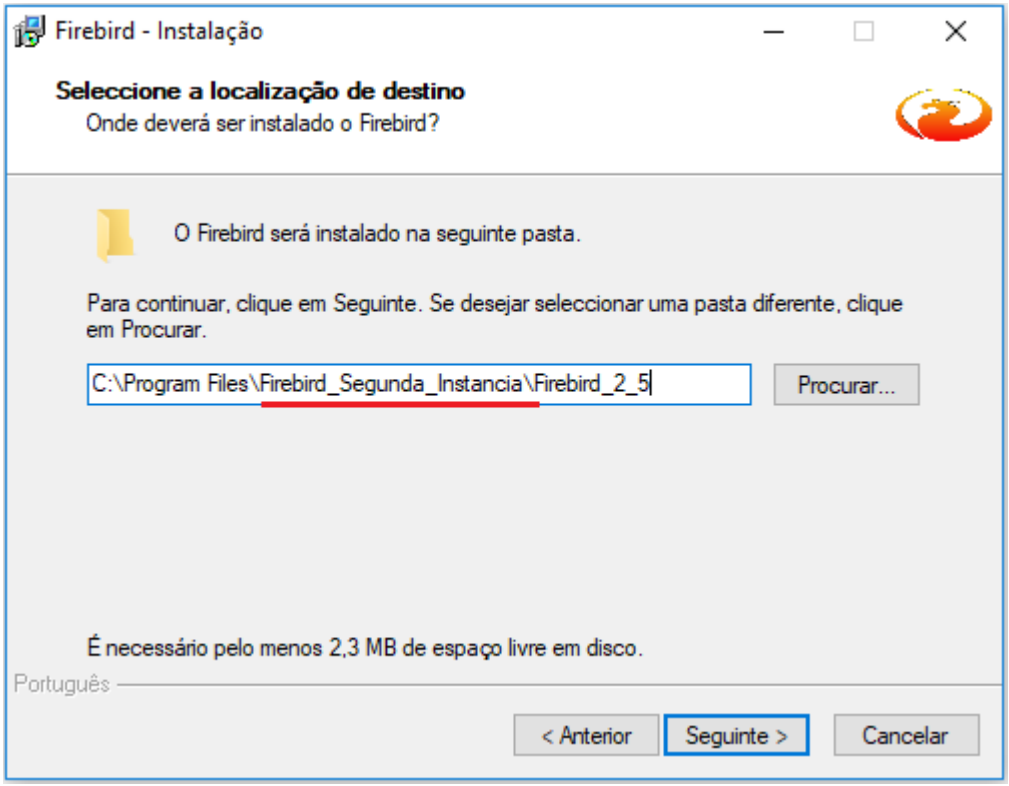
O programa irá identificar que já existe uma instalação, e exibirá a seguinte mensagem:



A mensagem pergunta se você quer cancelar a instalação, pois já existe outra configurada nessa máquina, e ainda explica que ser optar por continuar o Firebird será instalado mas não configurado, e teremos que completar a configuração manualmente. É exatamente isso que precisamos fazer, então clique em "Não".

3) Alterando o diretório de instalação

Prossiga com a instalação até a tela de configuração do diretório, onde deverá ser indicado uma pasta diferente do caminho da instalação já existente, que geralmente é "C:\Program Files\Firebird\Firebird_2_5", troque como desejar. Veja um exemplo na imagem a seguir:



Vá para as etapas seguintes até concluir a instalação.

4) Configurando o serviço do Firebird

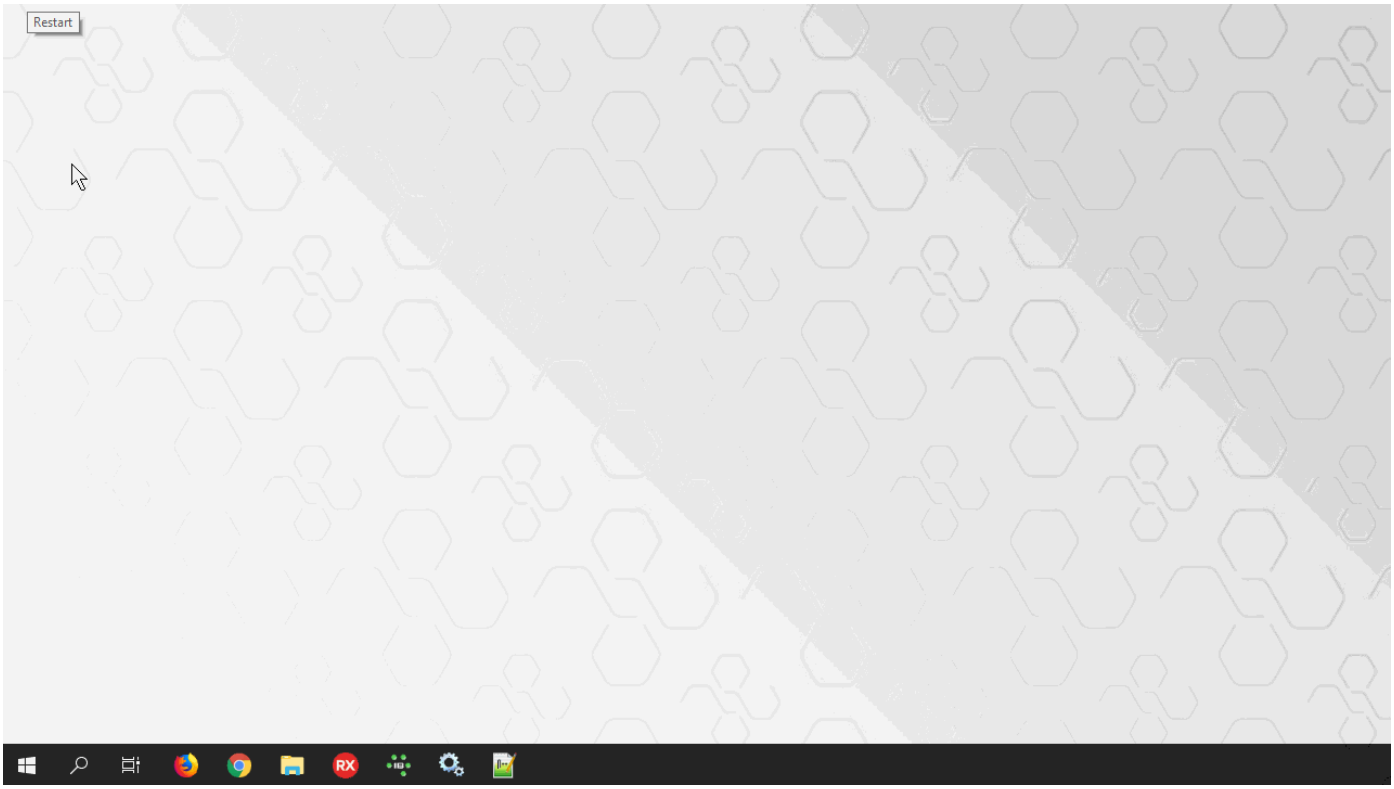
Vá até o diretório de instalação que foi indicado na etapa anterior, e localize o arquivo "firebird.conf", abra-o e procure pelo parâmetro "RemoteServicePort", remova o caractere

"#" e informe uma porta de conexão diferente da padrão 3050, pois essa já está em uso na instalação atual. Como sugestão utilize a porta 3051 ou a sequencia disponível.

```
# ----------------------------
# TCP Protocol Settings
#
# The TCP Service name/Port number to be used for client database
# connections.
#
# It is only necessary to change one of the entries, not both.  The
# order of precendence is the 'RemoteServiceName' (if an entry is
# found in the 'services.' file) then the 'RemoteServicePort'.
#
# Type: string, integer
#
#RemoteServiceName = gds_db
RemoteServicePort = 3051
```

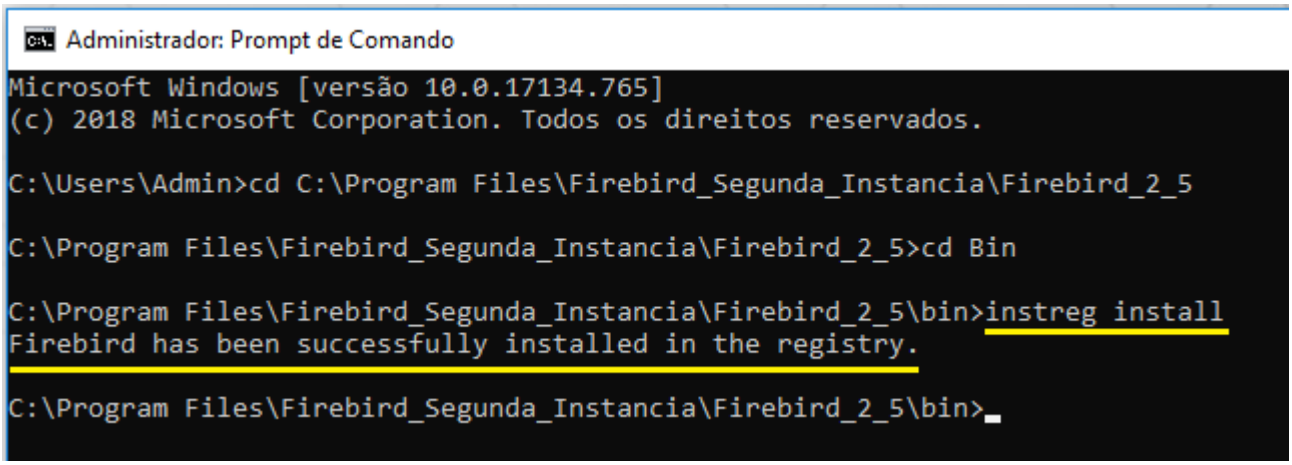
5) Instalando o serviço do Firebird

Abra o "Prompt de Comando" do Windows como administrador e vá até o diretório da nova instância do Firebird



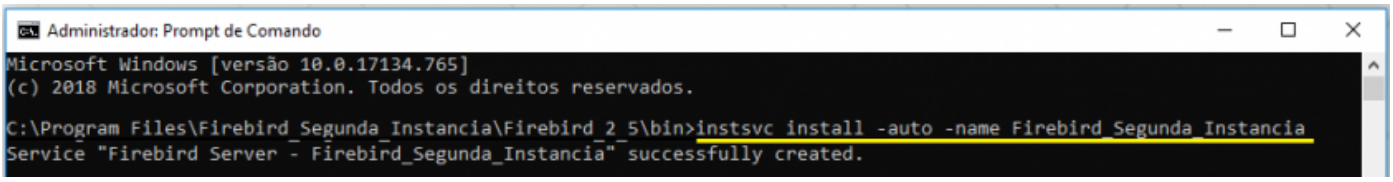
> Se a versão que você está instalando for anterior à 3.0, acesse a pasta "bin" com o comando **cd bin**

Agora execute o comando **instreg install** para registrar o Firebird no Windows.

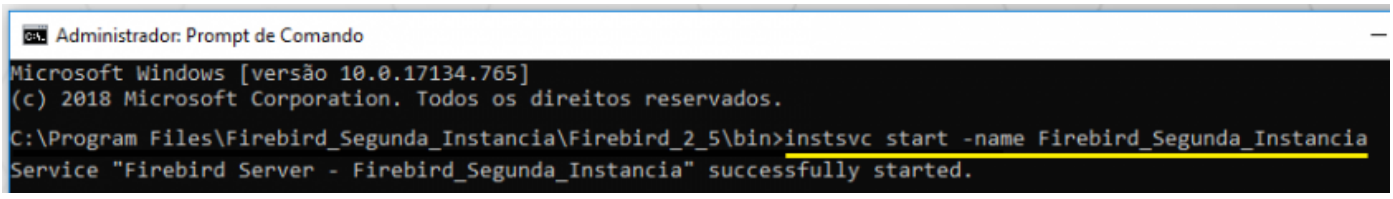


Instale o serviço atribuindo um nome para diferenciar da instalação já existente, com o seguinte comando

instsvc install -auto -name nome_do_servico

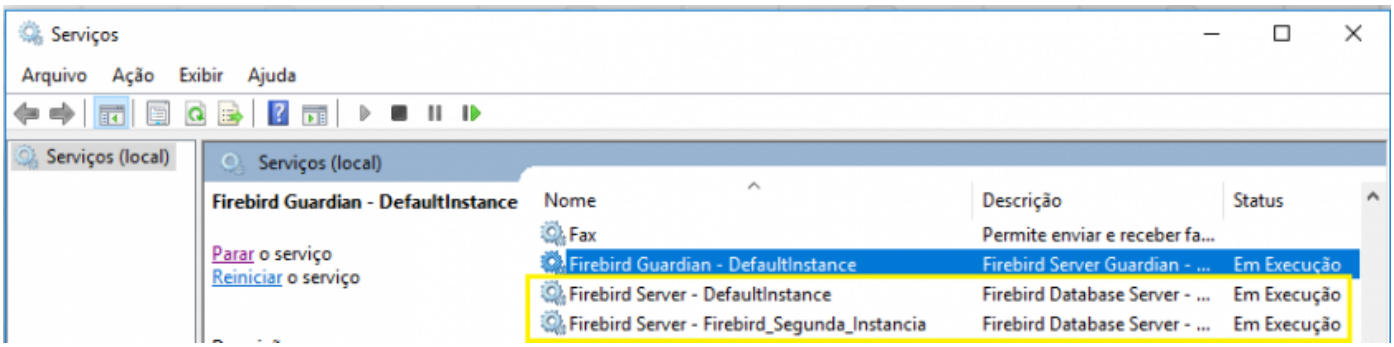


Para concluir precisamos apenas iniciar o novo serviço, com o comando **instsvc start -name nome_do_servico**



> Agora você pode iniciar o serviço da instalação que já existia, e configurar as suas conexões direcionando para os serviços através da porta configurada.

Confira como ficou os serviços instalados:



Voltar à página inicial