

# Possibilitando o desenvolvimento do banco de dados ágil com o Toad – Parte 2

Quest™

Implemente os componentes de automação do Toad Development Suite for Oracle: Team Coding, Code Analysis, SQL Optimizer e Code Tester

Escrito por John Pocknell, gerente sênior de produtos, Quest®



O que é preciso para deixar a implantação do banco de dados tão ágil quanto o desenvolvimento de aplicativos?

Como apontamos em [Possibilitando o desenvolvimento do banco de dados ágil com o Toad – Parte 1](#), o caminho para o desenvolvimento ágil do banco de dados o afasta dos processos tradicionais e manuais e o leva em direção a pipelines completamente automatizados que simplificam o trabalho e minimizam o risco de perda de dados ou tempo de inatividade. Juntos, o Toad Development Suite for Oracle e o Toad Intelligence Central oferecem automação para sincronizar os ciclos de desenvolvimento das suas equipes de software de aplicativos e de bancos de dados.

Agora, na parte 2 deste resumo técnico, o foco está no Team Coding, Code Analysis, SQL Optimizer e Code Tester, componentes do Toad Development Suite for Oracle.

Os componentes abrangem quatro áreas de interesse na automação do desenvolvimento do banco de dados:

1. Acesso ao controle de origem
2. Teste de unidade PL/SQL
3. Análises de código
4. Otimização de SQL e PL/SQL

## CONFIGURAÇÃO DO AMBIENTE

Como ilustrado na Figura 1, a implementação do Toad Development Suite for Oracle inclui a execução do Toad for Oracle nos desktops da equipe de desenvolvimento. Ela também abrange vários repositórios do Toad e a integração com os sistemas de controle de versão.

Para a colaboração, o Team Coding é um utilitário integrado ao seu sistema de controle de versões.

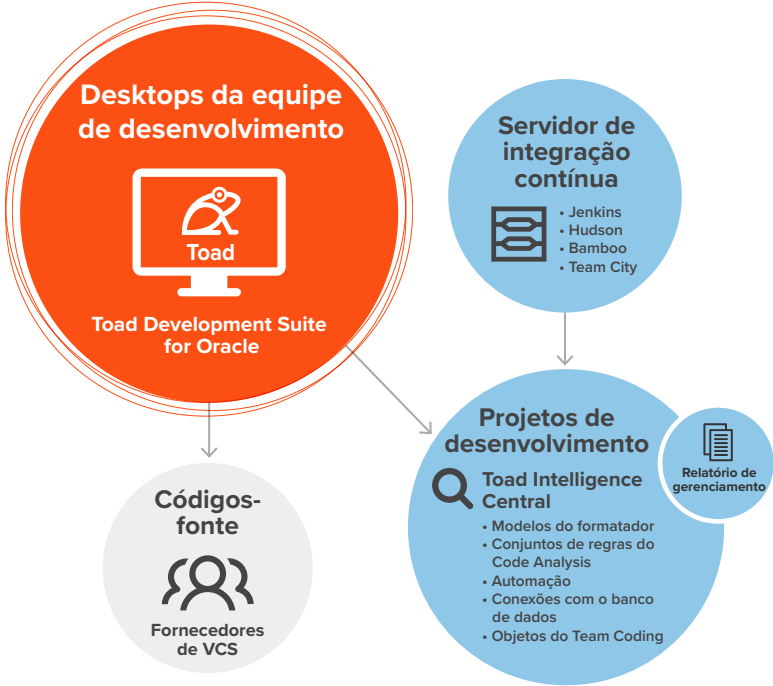


Figura 1: Relacionamentos entre bancos de dados, controle de versão e produtos Toad

O Toad Intelligence Central é executado em um servidor Windows que pode ser acessado por todos os desenvolvedores que utilizam o Toad (esse assunto será abordado na Parte 3 deste resumo técnico).

**INSTALAÇÃO DO DESKTOP DO DESENVOLVEDOR**

Instale o Toad Development Suite for Oracle nos sistemas de desenvolvimento. Ele inclui os seguintes produtos:

- Toad for Oracle Xpert Edition (inclui o SQL Optimizer for Oracle)
- Code Tester for Oracle
- Benchmark Factory for Oracle
- Toad Data Modeler
- Toad for MySQL Freeware

A implementação também requer um cliente Oracle para Windows. O cliente deve corresponder o máximo possível ao servidor Oracle no número da versão e na quantidade de bits (32 vs. 64). O Toad Development Suite for Oracle é flexível o suficiente para trabalhar com o Oracle Instant Client.

O Toad for Oracle pode ser instalado diretamente no desktop Windows com o

instalador da Web ou com um pacote de rede com instalação silenciosa.

**Implementação do Team Coding**

Para a colaboração, o Team Coding é um utilitário integrado ao seu sistema de controle de versões. Os desenvolvedores acessam o código PL/SQL e outros objetos do banco de dados da Oracle como de costume, mas o Team Coding controla e monitora o check-out/check-in dos arquivos correspondentes no controle de origem. O Team Coding oferece suporte aos seguintes fornecedores de VCS:

- Serena PVCS Version Manager
- Microsoft Visual SourceSafe
- Microsoft Team Foundation Server
- Microsoft Visual Studio Team Services (a partir do Toad 12.9)
- Mercurial (a partir do Toad 12.10)
- IBM Rational ClearCase
- Perforce
- CVS
- Git
- Subversion (CollabNet recomendado)

Normalmente, o Team Coding requer que um conjunto de tabelas do repositório seja instalado em um esquema no banco de dados da Oracle, mas a partir do Toad for Oracle v12.10, você poderá ter os objetos do Team Coding instalados no servidor do Toad Intelligence Central. O resultado é um repositório totalmente centralizado para todos os seus projetos sem a necessidade de instalar o Team Coding em cada instância do banco de dados.

Para obter mais informações sobre a configuração do Team Coding com a integração VCS, [aqui](#).

Para obter mais informações sobre o uso do Team Coding com VCS, [aqui](#).

### Implementação do Code Analysis

O Code Analysis é um recurso baseado em regras para a análise de código. Ele é fornecido com aproximadamente 200 regras de codificação predefinidas em diversos Conjuntos de regras de categoria padrão, como Estrutura do programa, Manutenção e Eficiência. Como mostra a Figura 2, você pode usar

seus próprios padrões de codificação para criar seus próprios Conjuntos de regras e, então, compartilhar as regras com os membros de cada equipe. O Code Analysis também inclui um repositório para armazenar os resultados das suas análises de código.

Para obter mais informações sobre a configuração e o uso do Code Analysis, [aqui](#).

Você pode selecionar uma opção no Team Coding que exigirá uma análise de código sempre que desenvolvedores tentarem verificar seus códigos em um controle de origem. O Toad for Oracle evitará que os códigos que não atenderem aos níveis de qualidade exigidos sejam verificados até que os problemas sejam corrigidos.

Para obter mais informações sobre a configuração das políticas do Team Coding, [aqui](#).

Outra opção é publicar os resultados das análises de código no Toad Intelligence Central, que apresenta tendências históricas e informações baseadas em eventos por meio deste servidor da Web.

Você pode selecionar uma opção no Team Coding que exigirá uma análise de código sempre que desenvolvedores tentarem verificar seus códigos em um controle de origem.

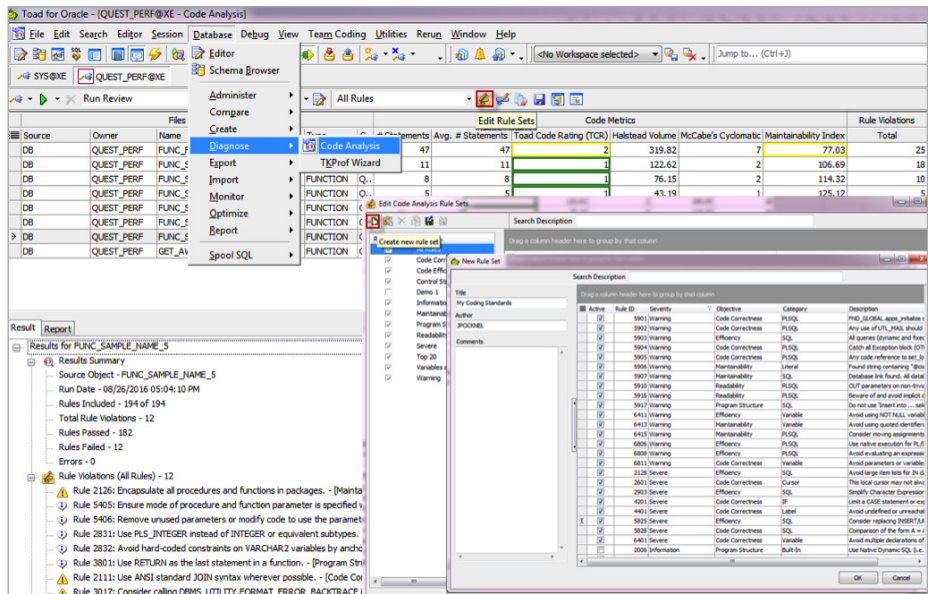


Figura 2: Criação de um novo Conjunto de regras no Code Analysis e seleção de regras da biblioteca

O Auto Optimize SQL gera múltiplas regravações da sua instrução SQL original até encontrar uma que seja executada de forma mais rápida.

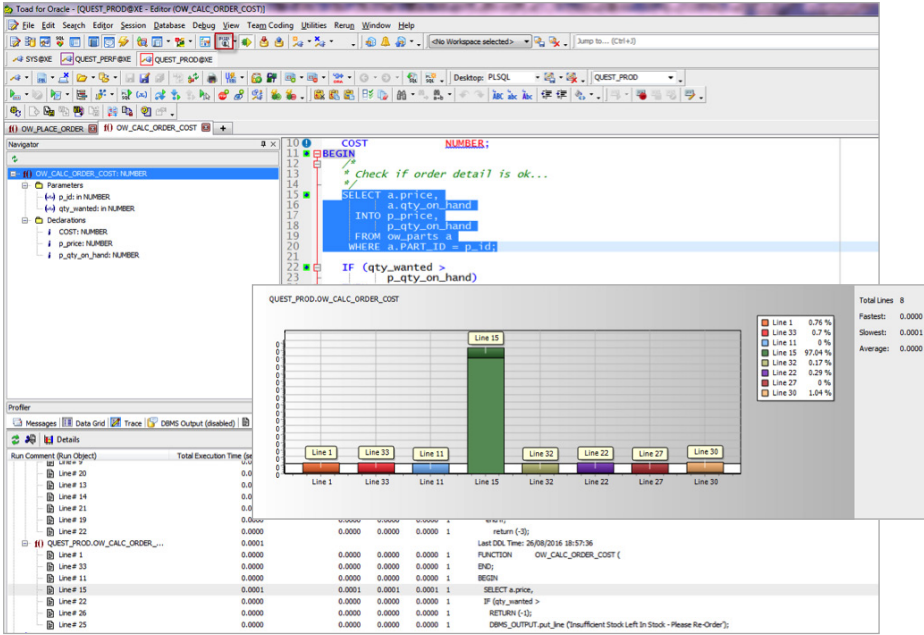


Figura 3: PL/SQL Profiler identifica uma instrução SQL com execução lenta dentro de um programa PL/SQL

**Otimização de SQL e PL/SQL**

Quando um programa PL/SQL é executado de forma muito lenta, tente encontrar o gargalo. Se o problema for uma instrução SQL mal gravada, tente aprimorá-la. O Toad e o SQL Optimizer tratam do processo em duas etapas.

**Etapa 1: Definição de perfil PL/SQL**

O recurso PL/SQL Profiler no Toad funciona em conjunto com o Editor para

cronometrar a execução de cada linha do código PL/SQL conforme o programa é executado. O PL/SQL Profiler armazena os dados em um repositório e os representa graficamente como mostra a Figura 3. Qualquer gargalo no desempenho se tornará imediatamente aparente.

Para obter mais informações sobre a configuração e o uso do PL/SQL Profiler, assista a este vídeo.

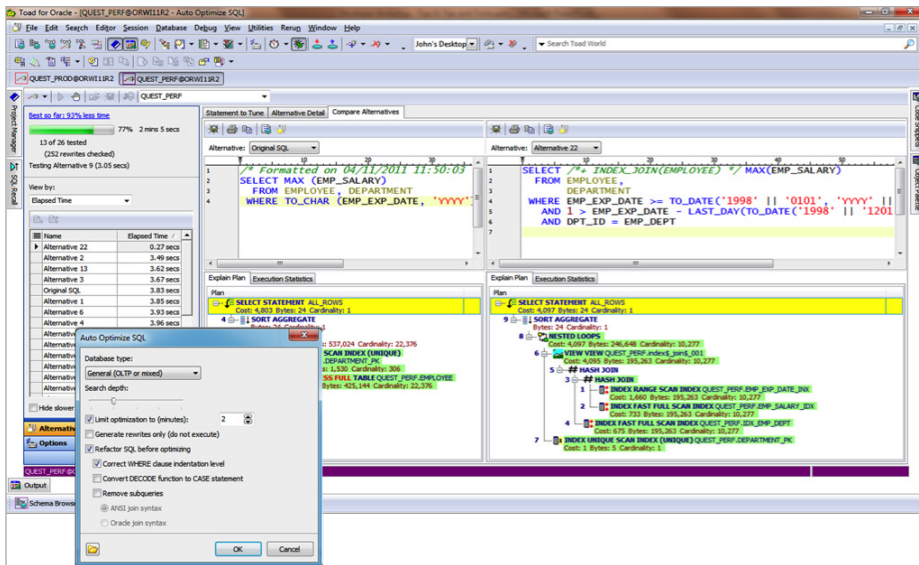


Figura 4: Auto Optimize SQL encontra automaticamente um SQL alternativo mais rápido

## Etapa 2: Otimização do SQL

Depois, use o SQL Optimizer no Toad para resolver os gargalos. O SQL Optimizer reforma automaticamente a instrução para remover quaisquer elementos específicos a PL/SQL, como instruções INTO, e converte variáveis declaradas localmente em variáveis vinculadas.

O recurso Auto Optimize SQL (veja a Figura 4) começará então a gerar

múltiplas regravações da sua instrução SQL original até encontrar uma que seja executada de forma mais rápida. Ele apresenta planos de execução e estatísticas para ajudá-lo a selecionar a melhor instrução alternativa.

Para obter mais informações sobre o uso do Auto Optimize SQL, assista a este vídeo.

Com o Code Tester, ao descrever seus testes com base nos casos de uso, a ferramenta gerará um código de teste como um pacote PL/SQL que implementará a sua definição do teste.

Com o Code Tester, você não precisa escrever o código de teste. Descreva seus testes para que a ferramenta gere um código que implementará a sua definição do teste.

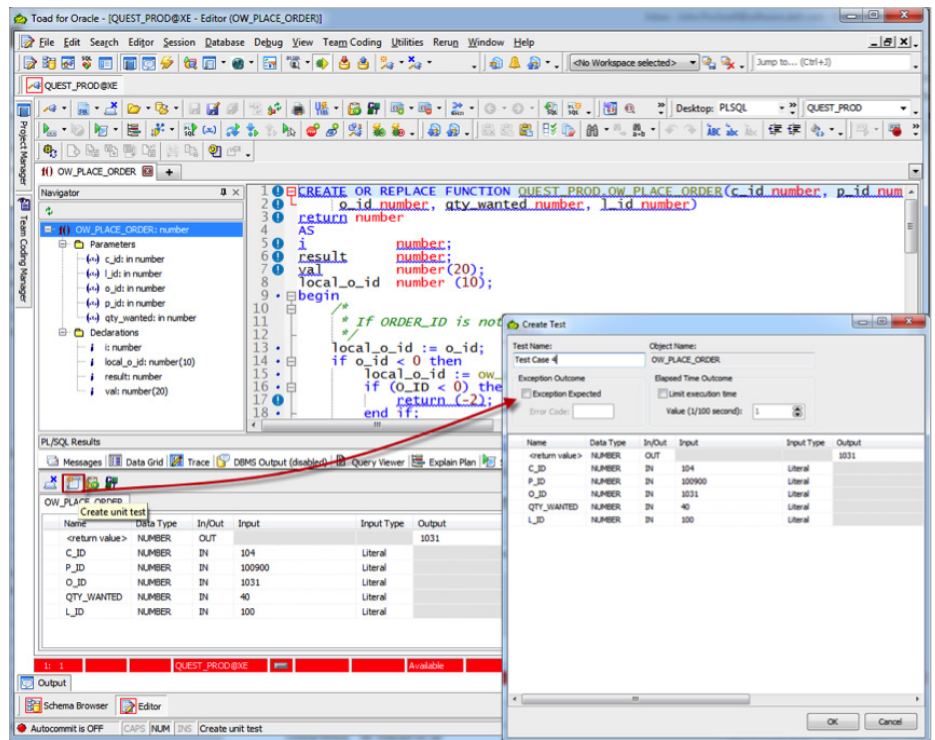


Figura 5: Criação de um teste de unidade PL/SQL no Toad

## IMPLEMENTAÇÃO DO CODE TESTER FOR ORACLE

Testar o código é como uma espada de dois gumes. Os desenvolvedores sabem que o teste é importante, mas ele geralmente envolve escrever um grande volume de código de teste, para mantê-lo sincronizado com as alterações no código do aplicativo, para verificar os resultados do teste e para tomar outras medidas que mantêm os desenvolvedores de forma eficaz em suas funções mais importantes.

O Code Tester for Oracle facilita a definição dos testes, a geração do código de teste e a execução dos testes, tudo dentro de uma interface gráfica fácil de usar. O melhor de tudo é que, com o Code Tester, você não precisa escrever o código de teste. Ao descrever seus testes com base nos casos de uso, a ferramenta gerará um código de teste como um pacote PL/SQL que implementará a sua definição do teste.

O Code Tester armazena os dados de execução dos testes de unidade no repositório sempre que você executar os

testes diretamente do Toad Editor ou do próprio Code Tester.

O repositório do Code Tester precisa ser instalado no mesmo banco de dados em que os desenvolvedores definirão seus testes. Você pode ativar o acesso público ou privado ao repositório dependendo dos seus requisitos.

## Criação dos testes de unidade no Toad Editor

Como mostra a Figura 5, você pode criar testes de unidade diretamente no Toad for Oracle e armazená-los no repositório do Code Tester. O processo cria testes reutilizáveis que serão armazenados para o teste de regressão contínuo durante a vida útil do código.

No Toad Editor, ao executar o código normalmente, o Code Tester usa qualquer valor de entrada e de saída para criar a definição do teste de unidade, que é então armazenada em seu repositório.

Para obter mais informações sobre a criação de testes de unidade PL/SQL no Toad, assista a este vídeo.

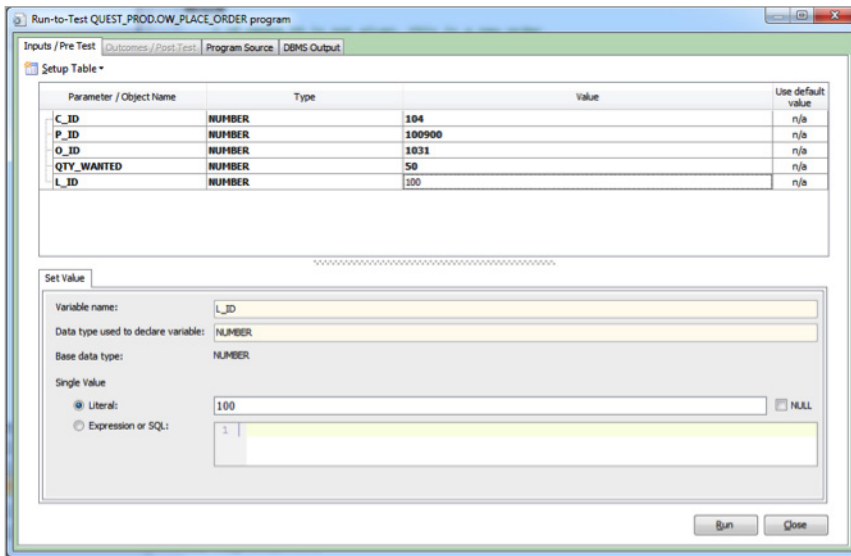


Figura 6: Criação de um teste de unidade PL/SQL no Code Tester com o Run-to-Test

### Criação dos testes de unidade no Code Tester

No Code Tester, a maneira mais simples de criar testes de unidade é usar o recurso Run-to-Test. Ele fornece opções mais abrangentes para definir os valores de entrada e os resultados esperados.

Como observado acima, todos os testes de unidade, criados no Toad ou no Code Tester, serão armazenados no repositório do Code Tester. Lá, você poderá visualizar todos os seus testes, pacotes de teste e código PL/SQL (consulte a Figura 7).

Para obter mais informações sobre a criação de testes de unidade PL/SQL abrangentes no Code Tester, assista a este vídeo.

### CONCLUSÃO

A automação é necessária para tornar o desenvolvimento do banco de dados tão ágil quanto o desenvolvimento de aplicativos. A combinação do Toad Development Suite for Oracle e do Toad Intelligence Central oferece aos desenvolvedores de bancos de dados excelentes vantagens em alcançar o ágil desenvolvimento de aplicativos por meio

O recurso Run-to-Test fornece opções mais abrangentes para definir os valores de entrada e os resultados esperados.

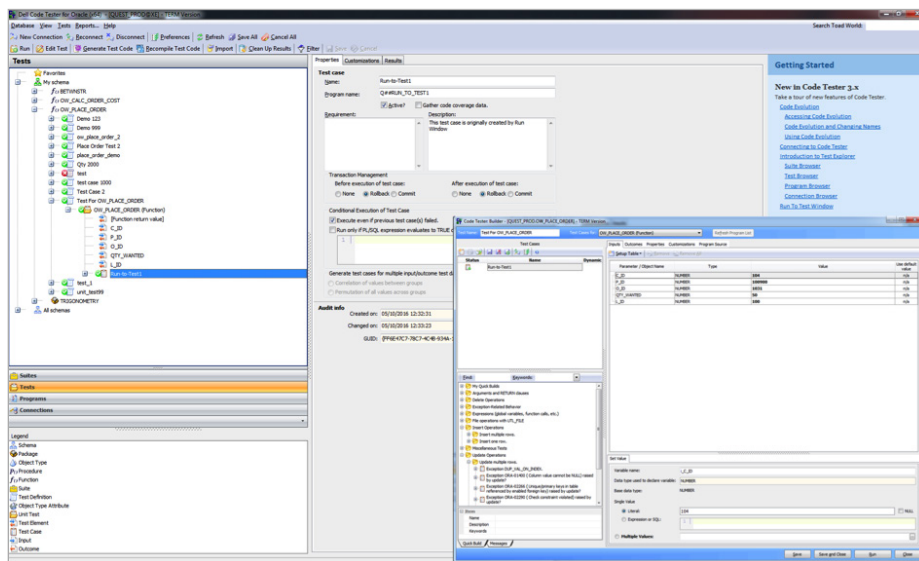


Figura 7: Janela principal do Code Tester com o Test Builder (inserção)

de recursos automatizados, como Team Coding, Code Analysis, PL/SQL Profiler, SQL Optimizer e Code Tester. Agora, os desenvolvedores de bancos de dados têm a oportunidade de reduzir os ciclos de desenvolvimento ao mesmo tempo que minimizam os riscos.

### **SOBRE O AUTOR**

John Pocknell é gerente sênior de gerenciamento de produtos na Quest. Ele trabalha na sede europeia no Reino

Unido e é responsável pelo portfólio de produtos Toad. Ele está na Quest desde 2000 e trabalha no design de banco de dados e nas áreas de desenvolvimento e implantação de produtos, além de conduzir diversos cursos de treinamento do Toad para clientes. John passou os últimos 15 anos divulgando o sucesso do Toad para clientes em diversos eventos em toda a Europa, EUA e a região Ásia-Pacífico. John escreve em vários blogs e artigos

publicados na comunidade de usuários do Toad, o Toad World.

John trabalha com TI há quase 30 anos, sendo a maior parte desse tempo dedicada ao design e desenvolvimento de aplicativos Oracle. Ele é um engenheiro aeronáutico qualificado com mais de 10 anos de experiência no provisionamento de serviços de consultoria de TI e na implementação de sistemas de garantia de qualidade ISO 9001.



## **SOBRE A QUEST**

A Quest ajuda os clientes a reduzir as tarefas administrativas enfadonhas para que eles possam se dedicar à inovação necessária para ampliar os negócios. As soluções Quest® são escaláveis, acessíveis e simples de usar, proporcionando eficiência e produtividade sem comparação. Juntamente com o convite da Quest à comunidade global para fazer parte de sua inovação, assim como nosso firme compromisso em garantir a satisfação dos clientes, a Quest continuará a acelerar o fornecimento das soluções mais abrangentes para gerenciamento de cloud do Azure, SaaS, segurança, mobilidade da força de trabalho e insights conduzidos por dados.

© 2017 Quest Software Inc. TODOS OS DIREITOS RESERVADOS.

Este guia contém informações confidenciais protegidas por direitos autorais. O software descrito nesse guia é oferecido sob uma licença de software ou um contrato de confidencialidade. Ele pode ser usado ou copiado apenas de acordo com os termos do acordo aplicável. Nenhuma parte deste guia pode ser reproduzida ou transmitida em qualquer forma ou por qualquer meio, eletrônico ou mecânico, inclusive fotocópia e gravação para qualquer propósito, sem a permissão por escrito da Quest Software Inc.

As informações deste documento são fornecidas em relação aos produtos da Quest Software. Este documento, isoladamente ou em conjunto com a venda de produtos da Quest Software, não concede nenhuma licença, expressa ou implícita, por preclusão ou de qualquer outra forma, a qualquer direito de propriedade intelectual. SALVO CONFORME DEFINIDO NOS TERMOS E CONDIÇÕES DA QUEST SOFTWARE ESPECIFICADOS NOS CONTRATOS DE LICENÇA PARA ESTE PRODUTO, A QUEST SOFTWARE NÃO ASSUME QUALQUER RESPONSABILIDADE E RENUNCIA A QUALQUER GARANTIA, EXPRESSA, IMPLÍCITA OU ESTATUTÁRIA, RELACIONADA A SEUS PRODUTOS, INCLUINDO, ENTRE OUTROS, A GARANTIA IMPLÍCITA DE COMERCIALIZAÇÃO, ADEQUAÇÃO A DETERMINADO PROPÓSITO OU NÃO VIOLAÇÃO. EM HIPÓTESE ALGUMA, A QUEST SOFTWARE SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, CONSEQUENCIAL, PUNITIVO, ESPECIAL OU INCIDENTAL (INCLUINDO, SEM LIMITAÇÕES, DANOS POR LUCROS CESSANTES, INTERRUPTÃO DOS NEGÓCIOS OU PERDA DE INFORMAÇÕES) DECORRENTES DO USO OU DA INCAPACIDADE DE USO DESTES DOCUMENTOS, MESMO QUE A QUEST SOFTWARE TENHA SIDO ALERTADA SOBRE A POSSIBILIDADE DE TAIS DANOS. A Quest Software não se responsabiliza por qualquer garantia ou declaração referente à exatidão ou à integridade deste documento e reserva-se o direito de fazer alterações em especificações e descrições de produtos a qualquer momento, sem aviso prévio. A Quest Software não se compromete em atualizar as informações contidas neste documento.

### **Patentes**

A Quest Software tem orgulho de nossa tecnologia avançada. Este produto pode ter a aplicação de patentes e de patentes pendentes. Para ver as informações mais recentes sobre as patentes aplicáveis a esse produto, visite nosso site em [www.quest.com/legal](http://www.quest.com/legal).

### **Marcas comerciais**

Quest, Toad, Benchmark Factory e o logotipo Quest são marcas comerciais e marcas registradas da Quest Software Inc. Para conferir a lista completa de marcas da Quest, acesse [www.quest.com/legal/trademark-information.aspx](http://www.quest.com/legal/trademark-information.aspx). Todas as outras marcas comerciais pertencem a seus respectivos proprietários.

Se você tiver dúvidas sobre o possível uso deste material, entre em contato:

#### **Quest Software Inc.**

A/C: LEGAL Dept  
4 Polaris Way  
Aliso Viejo, CA 92656

Acesse nosso site ([www.quest.com](http://www.quest.com)) para obter informações sobre escritórios regionais ou internacionais.