

Analizador de performance de programa Progress

Evandro Rafael Ochner

Wilson Pedro Carli (Orientador)

Roteiro da Apresentação

- **Introdução**
 - Objetivos do trabalho
- **Fundamentação teórica**
 - Contexto atual do tema
 - Conceitos básicos
 - Trabalhos correlatos
- **Desenvolvimento do trabalho**
 - Requisitos principais do problema
 - Especificação
 - Técnicas e ferramentas utilizadas
 - Apresentação da especificação

Roteiro da Apresentação

- Implementação
 - Técnicas e ferramentas utilizadas
 - Operacionalidade da implementação
- Resultados e discussão
- Conclusão
- Extensões

Introdução

- As tecnologias são aprimoradas e aperfeiçoadas para que a humanidade tenha recursos que possam facilitar a sua vida no dia-a-dia;
- É necessário que a solução de determinado problema seja apresentada o mais rápido possível;
- O teste de performance faz parte de um processo de otimização dos sistemas e pode ser classificado em três categorias:
 - Velocidade;
 - Escalabilidade;
 - Estabilidade.

Introdução

- Neste trabalho é enfocada a categoria de velocidade, tendo como parâmetros a quantidade de registros lidos pelo banco de dados em relação a registros utilizados pelo programa que for analisado;
- São relacionados dados do programa em análise através do código fonte do mesmo:
 - Nome das tabelas que estão sendo utilizadas;
 - Tipos acessos que estão sendo feitos ao banco de dados com cada uma das tabelas (inclusão, consulta, alteração ou exclusão);
 - Índices informados explicitamente no código fonte para avaliar se foram corretamente escolhidos.

Introdução

- A análise dos resultados finais é a parte mais importante do processo;
- Devem ser decididas quais atitudes serão tomadas em relação ao programa que foi analisado para melhorar sua performance.

Introdução

Objetivos do trabalho

- **Objetivo geral:**
 - Construir uma ferramenta capaz de fazer a análise de um programa desenvolvido na linguagem Progress e voltado ao banco de dados Progress, visando auxiliar na otimização da busca de registros na base de dados.
- **Objetivos específicos:**
 - Relacionar as informações das tabelas com os tipos acessos utilizados;
 - Verificar a utilização correta de índices de busca nas tabelas;
 - Contabilizar os acessos ao banco de dados, assim como utilização de registros;
 - Fazer uma estatística em relação as informações coletadas;
 - Emitir um relatório com as informações reunidas durante todo o processo.

Fundamentação teórica

Contexto atual do tema

- Muitos sistemas computacionais crescem de forma exponencial e mesmo assim devem manter seu desempenho de forma otimizada;
- A primeira razão para que seja utilizada a análise de performance é assegurar-se de que o sistema em análise encontra-se em conformidade com as necessidades projetadas. É estabelecer quanto desempenho pode ser extraído do sistema atualmente;

Fundamentação teórica

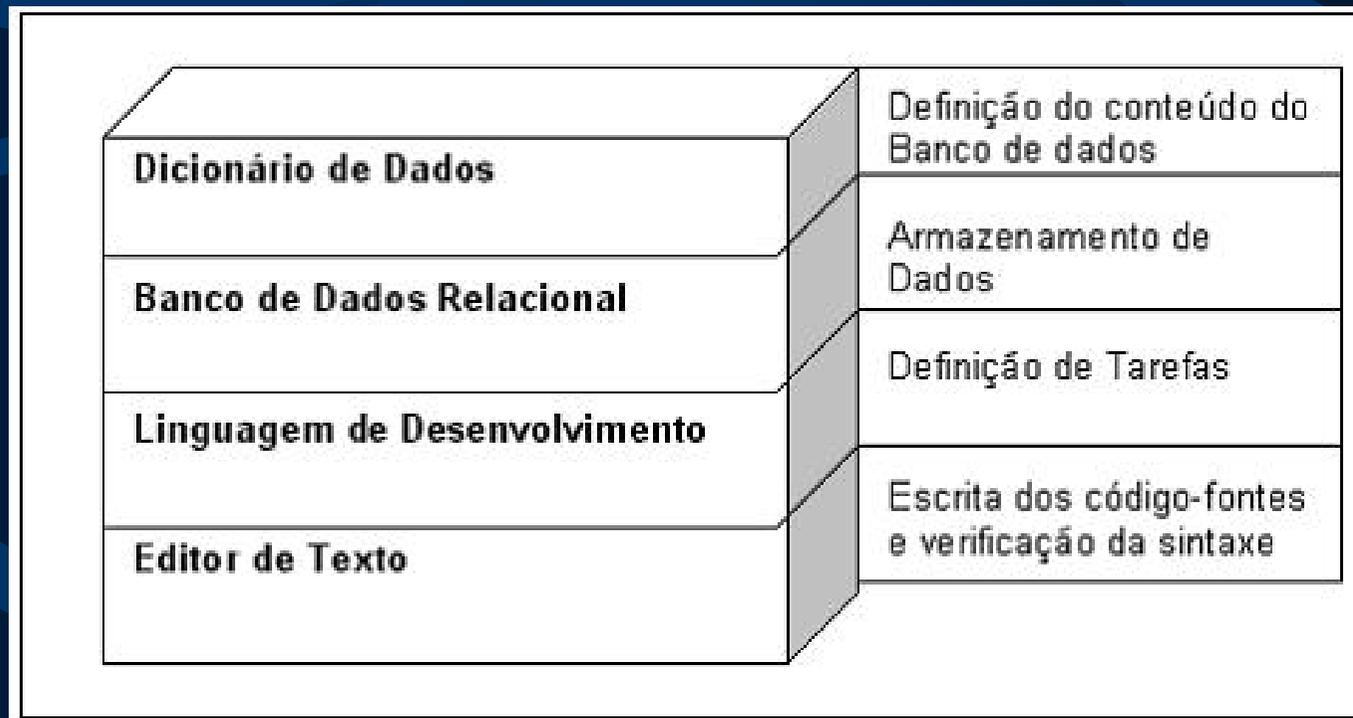
Contexto atual do tema

- A segunda razão, é planejar quando algo deve ser feito, a fim de suportar uma carga maior que a atual. Isso pode incluir reescrever parte da solução atual, reestruturando a mesma, ou a adição de mais funcionalidades ao sistema;
- Existem basicamente duas formas para que a performance de um sistema seja aumentada:
 - Melhoria de hardware da infra-estrutura onde o sistema opera (o que geralmente é considerado um investimento bastante caro);
 - Melhoria de desempenho de software em todo o sistema, aproveitando ao máximo o hardware existente (é necessário utilizar-se de um instrumento para medir a performance).

Fundamentação teórica

Conceitos básicos

- Banco de dados Progress
 - Relacional;
 - Dividido basicamente em 4 ítems:



Fundamentação teórica

Conceitos básicos

- Linguagem Progress

- Linguagem de quarta geração, bastante abstrata e permite o desenvolvimento focado em regra de negócio, com menor preocupação com controles de estruturas de dados;
- Desenvolvida principalmente para o uso com o banco de dados Progress pois possui estruturas de manipulação de registros próprias para esse banco;
- Comentários:

```
/* Listagem de associados */  
FOR EACH associados WHERE associados.cooperativa = 1 NO-LOCK:  
    DISPLAY associados.conta associados.nome.  
END.
```

Fundamentação teórica

Conceitos básicos

- Os tipos de variáveis disponíveis no linguagem são: character, date, decimal, integer, logical, handle, memptr, raw, recid, rowid e widget-handle;
- Includes files são arquivos que podem ser utilizados como extensão de qualquer código fonte:

```
/* Utilização de includes */  
{ c:\includes\var_global.i }
```

Fundamentação teórica

Conceitos básicos

- Argumentos são valores que podem ser passados de forma literal como parâmetro para programas ou includes:

```
/* A.p */  
RUN c:\programas\programa1.p "tbAssociados" "nro_conta".  
/* ou */  
{ c:\includes\include1.i "tbAssociados" "nro_conta" }
```

```
/* B.p */  
FOR EACH {1} WHERE {1}.{2} > 0 NO-LOCK:  
    DISPLAY {1}.nm_associado.  
END.
```

Fundamentação teórica

Conceitos básicos

– Estruturas de controle:

```
/* Exemplo de utilização do comando DO */  
DEFINE VARIABLE aux_contador AS INTEGER NO-UNDO.  
DO aux_contador = 1 TO 10:  
    DISPLAY aux_contador.  
END.
```

```
/* Exemplo de utilização do comando DO WHILE */  
DEFINE VARIABLE aux_contador AS INTEGER NO-UNDO.  
DO WHILE aux_contador <= 10:  
    DISPLAY aux_contador.  
    aux_contador = aux_contador + 1.  
END.
```

```
/* Exemplo de utilização do comando REPEAT */  
REPEAT:  
    FIND NEXT tbAssociados.  
    DISPLAY tbAssociados.nro_conta  
        tbAssociados.nm_associado.  
END.
```

Fundamentação teórica

Conceitos básicos

```
/* Exemplo de utilização do comando FOR EACH */
FOR EACH tbAssociados NO-LOCK
    BREAK BY tbAssociados.cod_agencia
        BY tbAssociados.nm_associado:

    IF FIRST-OF(tbAssociados.cod_agencia) THEN
        DO:
            DISPLAY "Associados da Agencia " AT 12
                tbAssociados.cod_agencia ":"
                WITH FRAME f_agencia.

            CASE tbAssociados.cod_agencia:
                WHEN 1 THEN
                    MESSAGE "Bairro Centro".
                WHEN 2 THEN
                    MESSAGE "Bairro Velha".
                OTHERWISE
                    MESSAGE "Bairro Garcia".

            END CASE.
        END. /* FIM DO IF */

    DISPLAY tbAssociados.nro_conta
        tbAssociados.nm_associado
        WITH FRAME f_associados CENTERED.

END. /* Fim do FOR EACH */
```

Fundamentação teórica

Trabalhos correlatos

- Ferramenta desenvolvida por Maas
 - Desenvolvida em linguagem Progress;
 - Padronização e customização de código fonte.
- Sun Studio Performance Analyser
 - Análise de algoritmos;
 - Análise de estruturas de dados (pilhas, filas).

Desenvolvimento do trabalho

Requisitos principais do problema

- Fazer a análise de programas Progress a partir do código fonte dos mesmos;
- Listar as tabelas que são usadas no código fonte analisado;
- Identificar os tipos de acesso feitos ao banco de dados referente às tabelas encontradas (consulta, inclusão, alteração e exclusão);

Desenvolvimento do trabalho

Requisitos principais do problema

- Verificar se os índices informados explicitamente no código fonte são os índices adequados para a busca do banco de dados, utilizando a lógica que o sistema gerenciador de banco de dados Progress utiliza para escolher um índice;
- Contabilizar registros lidos pelo banco de dados e registros utilizados pelo programa durante sua execução;

Desenvolvimento do trabalho

Requisitos principais do problema

- Permitir ao usuário da ferramenta fazer parametrizações que serão consideradas na comparação entre a quantidade de registros lidos pelo banco de dados e registros utilizados no programa em análise;
- Emitir um relatório com possíveis situações onde haja uma quantidade excessiva de registros lidos pelo sistema gerenciador de banco de dados, na execução do programa, comparado com a quantidade de registros utilizados pelo programa em análise.

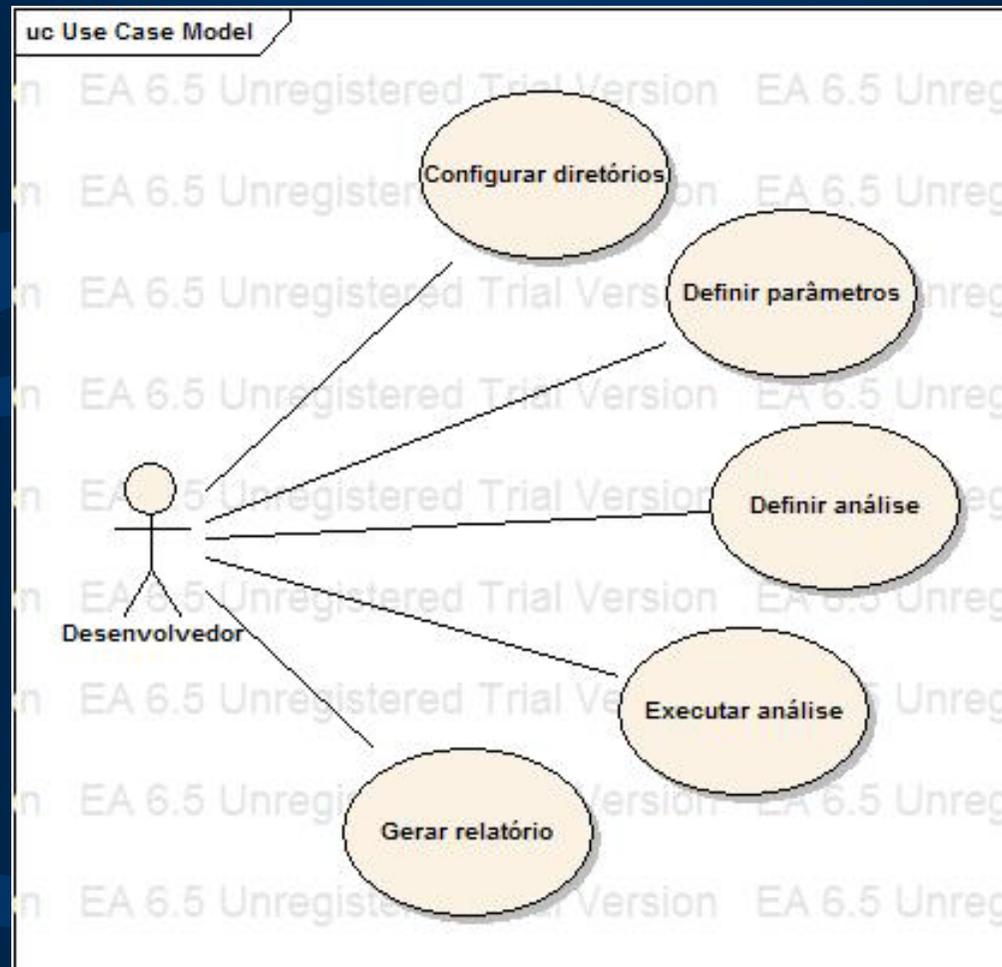
Especificação

Técnicas e ferramentas utilizadas

- Unified Modeling Language (UML);
- Para a modelagem foram utilizados os diagramas de casos de uso, diagramas de seqüência e diagrama de classes;
- Para a especificação da aplicação, utilizou-se a ferramenta Enterprise Architect (na versão 6.5).

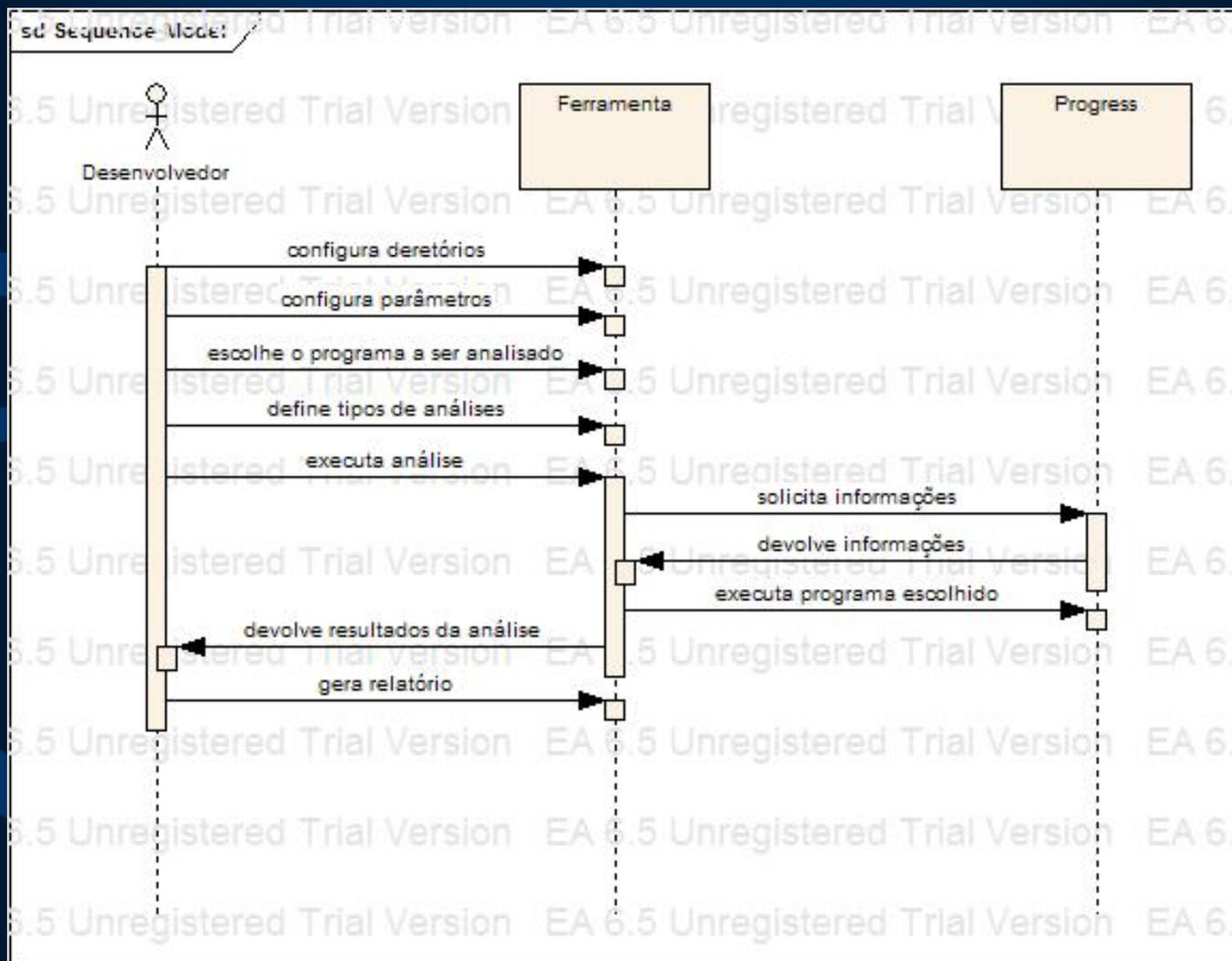
Especificação

Apresentação da especificação



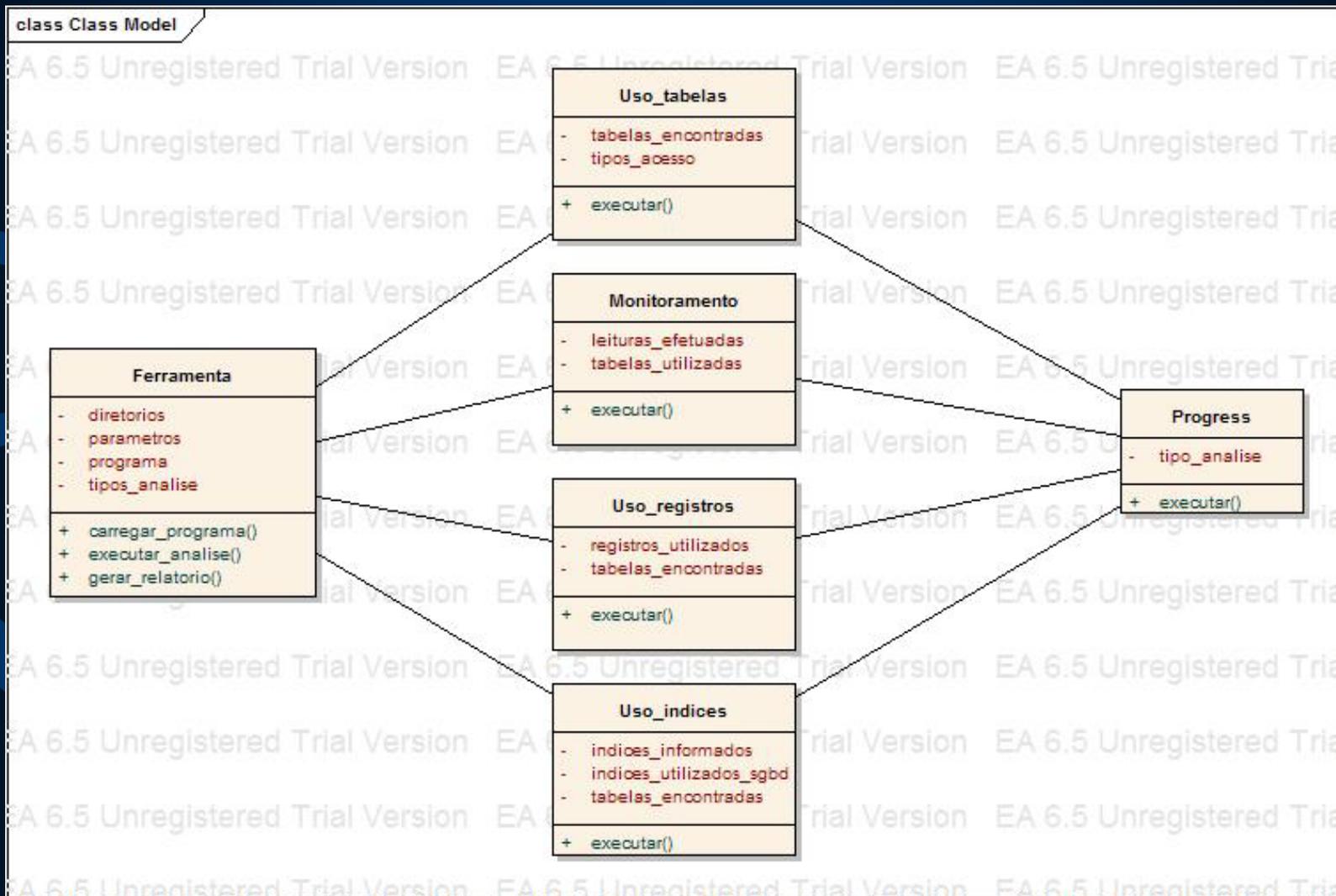
Especificação

Apresentação da especificação



Especificação

Apresentação da especificação



Implementação

Técnicas e ferramentas utilizadas

- Para desenvolvimento da aplicação foram utilizadas duas linguagens de programação:
 - Delphi (versão 6);
 - Progress (versão 9.1D).
- O sistema gerenciador de banco de dados utilizado foi Progress (versão 9.1D);
- O relatório gerado pela ferramenta é feito em formato HTML;

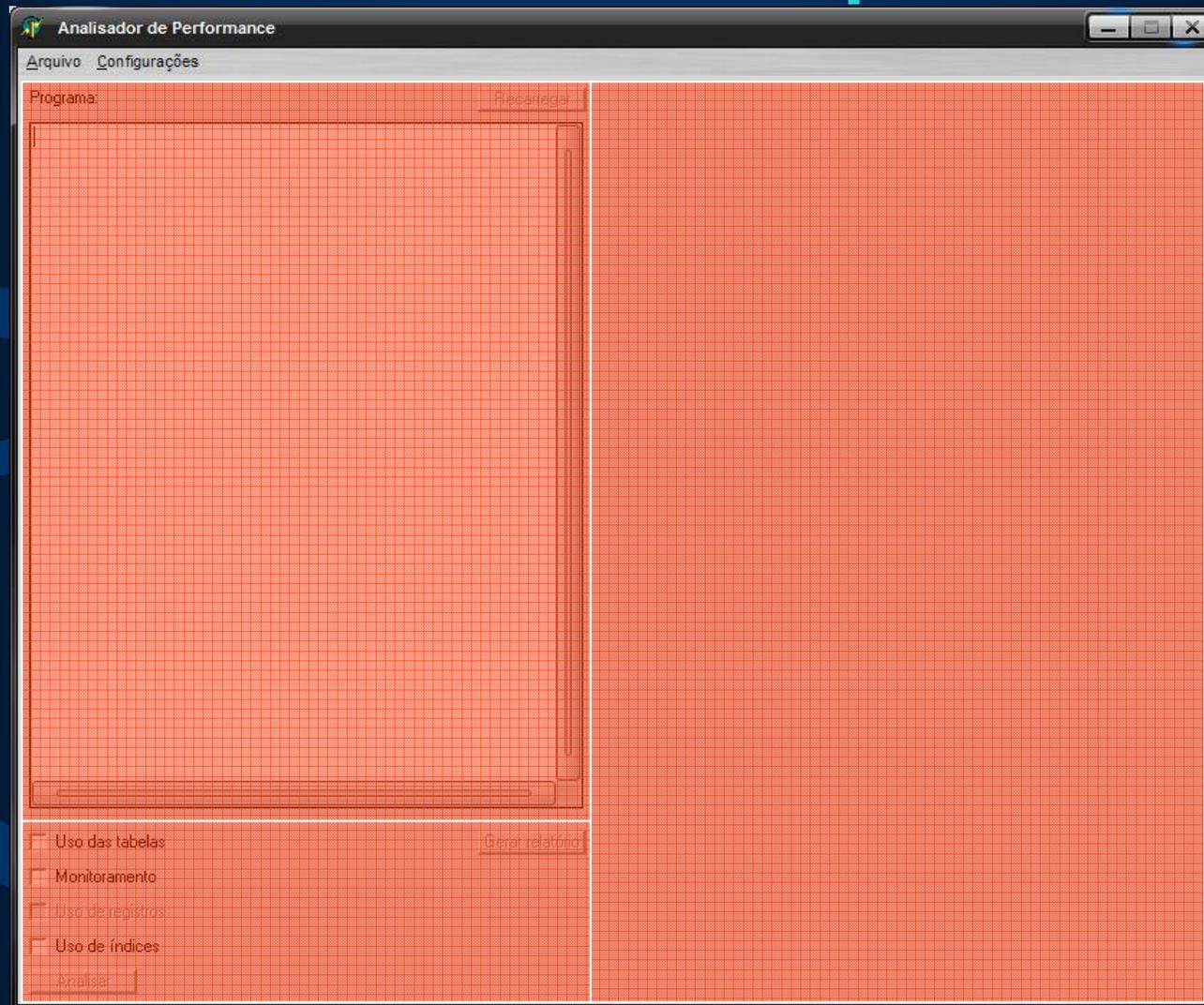
Implementação

Técnicas e ferramentas utilizadas

- A comunicação entre as linguagens citadas é feita através de arquivos texto;
- Para a criação da Backus-Naur Form (BNF) com os comandos da linguagem Progress foi utilizada a ferramenta GALS (versão 2003.10.03).

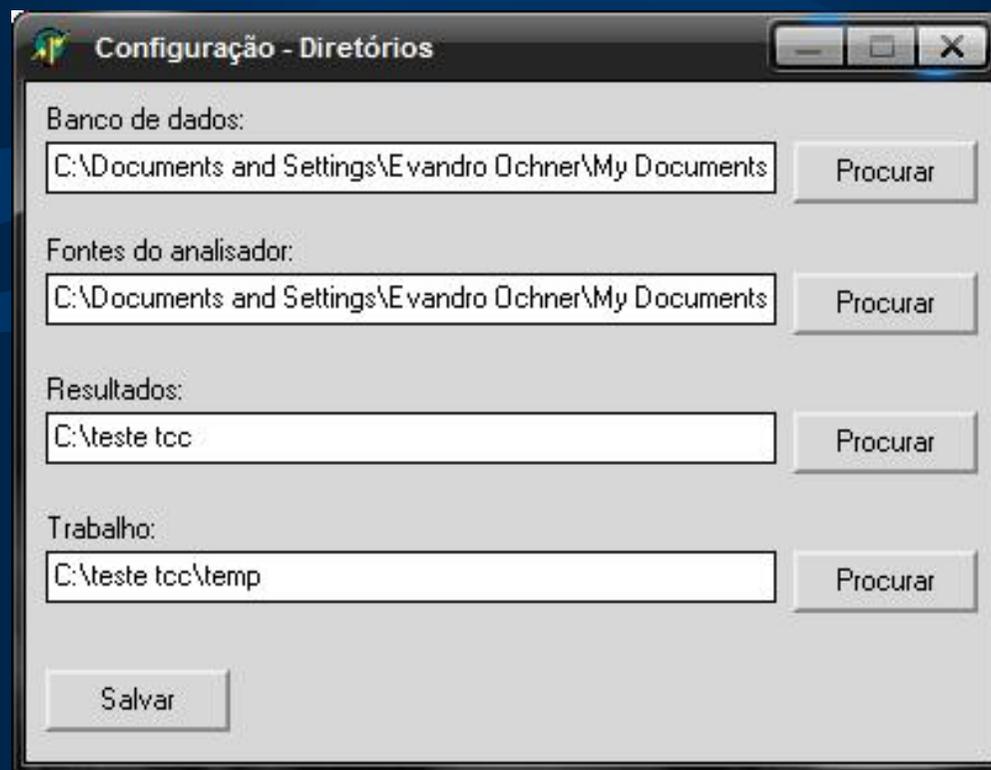
Implementação

Operacionalidade da implementação



Implementação

Operacionalidade da implementação



Implementação

Operacionalidade da implementação

Configuração - Parâmetros

Tipo de Contabilização

Percentual Quantidade

Valor padrão:

Padronizar por tabela:

Tabela	Valor
ASSOCIADOS	20

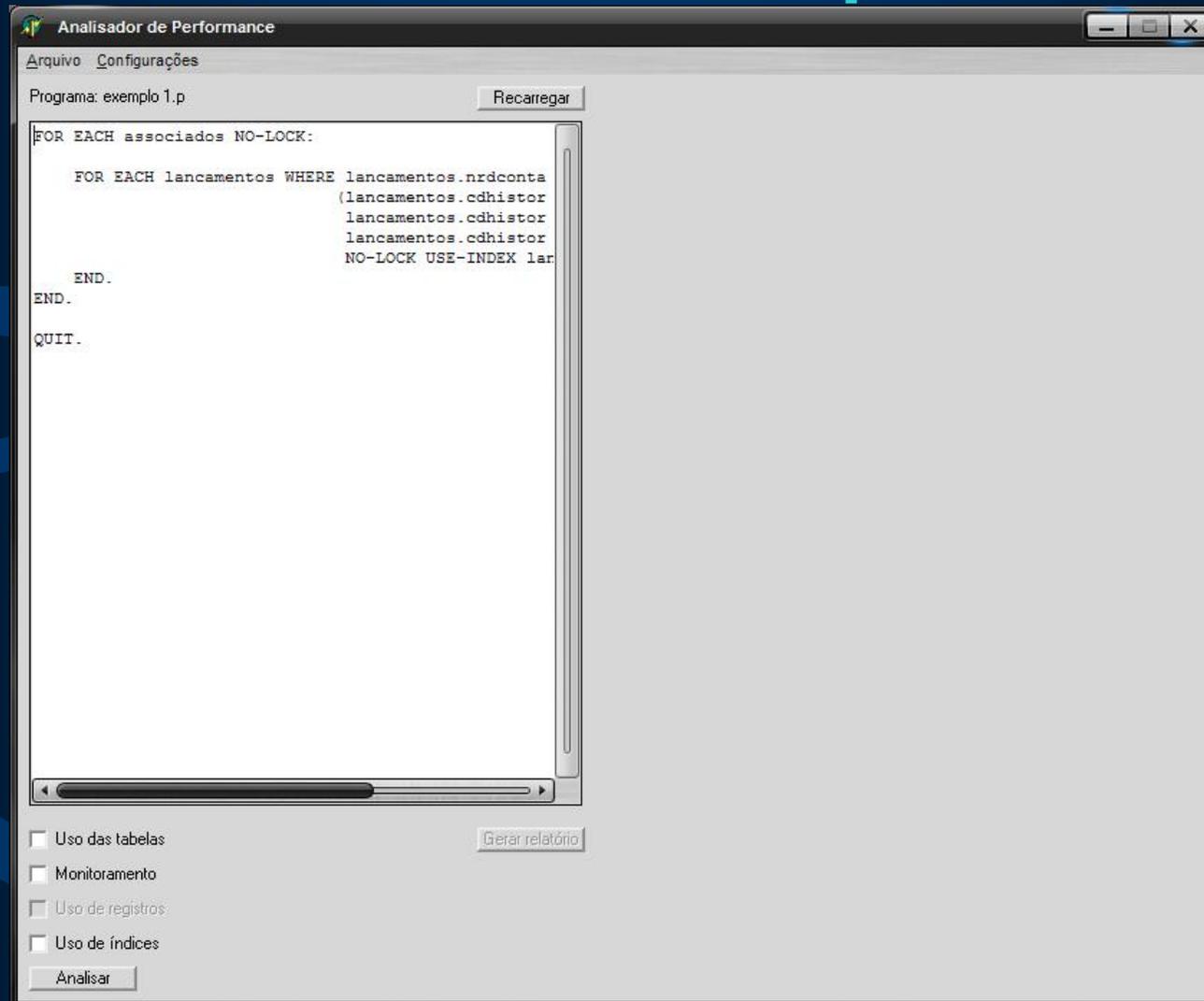
Adicionar tabela

Excluir tabela

Salvar

Implementação

Operacionalidade da implementação



Implementação

Operacionalidade da implementação

Arquivo Configurações

Programa: exemplo 1.p Recarregar

```
FOR EACH associados NO-LOCK:
  FOR EACH lancamentos WHERE lancamentos.nrdconta
    (lancamentos.cdhistor
    lancamentos.cdhistor
    lancamentos.cdhistor
    NO-LOCK USE-INDEX lar
  END.
END.
QUIT.
```

Uso das tabelas:

Tabela	Leitura	Atualização	Criação	Exclusão
ASSOCIADOS	SIM	NÃO	NÃO	NÃO
LANCAMENTOS	SIM	NÃO	NÃO	NÃO

Uso das tabelas Gerar relatório

Monitoramento

Uso de registros

Uso de índices

Analisar

Implementação

Operacionalidade da implementação

```
...
/* gera o XREF */
COMPILE VALUE (SESSION:PARAMETER) XREF VALUE(trabalho + "\uso_tabelas.xref").

/* importa o XREF e separa as tabelas e os tipos de acesso */
INPUT STREAM str_1 FROM VALUE(trabalho + "\uso_tabelas.xref") NO-ECHO.

DO WHILE TRUE ON ERROR UNDO, LEAVE
    ON ENDKEY UNDO, LEAVE:

    IMPORT STREAM str_1 UNFORMATTED aux_dsdlinha.

    ...
    IF (aux_dsdlinha MATCHES "*SEARCH*") OR
       (aux_dsdlinha MATCHES "*UPDATE*") OR
       (aux_dsdlinha MATCHES "*CREATE*") OR
       (aux_dsdlinha MATCHES "*DELETE*") THEN
        DO:
            ...
        END.
    END.
END.
...
```

Implementação

Operacionalidade da implementação

Arquivo Configurações

Programa: exemplo 1.p Recarregar

```
FOR EACH associados NO-LOCK:  
  
    FOR EACH lancamentos WHERE lancamentos.nrdconta  
                                (lancamentos.cdhistor  
                                lancamentos.cdhistor  
                                lancamentos.cdhistor  
                                NO-LOCK USE-INDEX lar  
  
    END.  
END.  
QUIT.
```

Monitoramento de registros:

Tabela	Lidos	Atualizados	Criados	Excluídos
ASSOCIADOS	2957	0	0	0
LANCAMENTOS	1478400	0	0	0

Usado das tabelas Gerar relatório

Monitoramento

Usado de registros

Usado de índices

Analisar

Implementação

Operacionalidade da implementação

Analizador de Performance

Arquivo Configurações

Programa: exemplo 1.p Recarregar

```
FOR EACH associados NO-LOCK:
    FOR EACH lancamentos WHERE lancamentos.nrdconta
        (lancamentos.cdhistor
        lancamentos.cdhistor
        lancamentos.cdhistor
        NO-LOCK USE-INDEX lar
    END.
END.
QUIT.
```

Monitoramento de registros:

Tabela	Lidos	Atualizados	Criados	Excluídos
ASSOCIADOS	2957	0	0	0
LANCAMENTOS	1478400	0	0	0

Uso de registros:

Tabela	Registros	Limite	Excesso	
ASSOCIADOS	2957	35%	0,00%	
LANCAMENTOS	4485	50%	32.863,21%	*

Uso das tabelas Gerar relatório

Monitoramento

Uso de registros

Uso de índices

Analisar

Implementação

Operacionalidade da implementação

```
function TSemantico.each : string;
begin
    correcao := correcao + 279;

    // Verificação para os FOR EACH'S
    result := 'FIND temp_analisador WHERE temp_analisador.nmtabela = "' +
        nmtabela + '" EXCLUSIVE-LOCK NO-ERROR. ' +

        'IF NOT AVAILABLE temp_analisador THEN ' +
        'DO: ' +
        'CREATE temp_analisador. ' +
        'ASSIGN temp_analisador.nmtabela = "' + nmtabela +
        '". ' +
        'END. ' +

        'ASSIGN temp_analisador.qttotreg = temp_analisador.qttotreg +
        1. ';

end;
```

Implementação

Operacionalidade da implementação

Arquivo Configurações

Programa: exemplo 1.p Recarregar

```
FOR EACH associados NO-LOCK:
  FOR EACH lancamentos WHERE lancamentos.nrdconta
    (lancamentos.cdhistor
    lancamentos.cdhistor
    lancamentos.cdhistor
    NO-LOCK USE-INDEX lar
  END.
END.
QUIT.
```

Uso das tabelas
 Monitoramento
 Uso de registros
 Uso de índices

Analisar

Gerar relatório

Uso de índices:

Linha	Explícito	Escolha SGBD
3	lancamentos2	lancamentos1

Implementação

Operacionalidade da implementação

Analizador de Performance

Arquivo Configurações

Programa: exemplo 1.p Recarregar

```
FOR EACH associados NO-LOCK:
  FOR EACH lancamentos WHERE lancamentos.nrdconta
    (lancamentos.cdhistor
    lancamentos.cdhistor
    lancamentos.cdhistor
    NO-LOCK USE-INDEX lar
  END.
END.
QUIT.
```

Gerar relatório

Uso das tabelas
 Monitoramento
 Uso de registros
 Uso de índices

Analisar

Uso das tabelas:

Tabela	Leitura	Atualização	Criação	Exclusão
ASSOCIADOS	SIM	NÃO	NÃO	NÃO
LANCAMENTOS	SIM	NÃO	NÃO	NÃO

Monitoramento de registros:

Tabela	Lidos	Atualizados	Criados	Excluídos
ASSOCIADOS	2957	0	0	0
LANCAMENTOS	1478400	0	0	0

Uso de registros:

Tabela	Registros	Limite	Excesso	
ASSOCIADOS	2957	35%	0,00%	
LANCAMENTOS	4485	50%	32.863,21%	*

Uso de índices:

Linha	Explícito	Escolha SGBD
3	lancamentos2	lancamentos1

Implementação

Operacionalidade da implementação

Analizador de Performance - Mozilla Firefox

file:///C:/teste%20tcc/

Flops Terra Mail Gmail Google Google - Idiomas Orkut

Analizador de Performance

Programa: exemplo 1.p

Uso de Tabelas

Tabela	Leitura	Atualização	Criação	Exclusão
ASSOCIADOS	SIM	NÃO	NÃO	NÃO
LANCAMENTOS	SIM	NÃO	NÃO	NÃO

Monitoramento de Registros

Tabela	Lidos	Atualizados	Criados	Excluídos
ASSOCIADOS	2957	0	0	0
LANCAMENTOS	1478400	0	0	0

Uso de Registros

Tabela	Registros	Limite	Excesso
ASSOCIADOS	2957	35%	0,00%
LANCAMENTOS	4485	50%	32.863,21%

Limite excedido

Uso de Índices

Linha	Explicito	Escolha SGBD
3	lancamentos2	lancamentos1

Done

Resultados e discussão

- Com a realização deste trabalho foi possível tornar a análise de performance de programas mais simples e de fácil acesso, uma vez que os procedimentos foram automatizados;
- É possível analisar comandos de busca a fim de que se consiga aproveitar ao máximo a capacidade do sistema gerenciador de banco de dados Progress;
- Em relação aos trabalhos correlatos, a ferramenta desenvolvida mescla algumas atividades dos mesmos:
 - Análise de código fonte;
 - Métricas de performance.

Conclusão

- **Pôde-se observar que a melhoria de performance de um sistema é um ponto muito importante nos tempos atuais, onde a informação deve ser precisa e obtida o mais rápido possível;**
- **A realização deste trabalho possibilitou um estudo mais profundo tanto do sistema gerenciador de banco de dados Progress, quanto da linguagem de programação Progress;**

Conclusão

- O aplicativo desenvolvido atende os objetivos aos quais se propõe e facilita a análise de performance de programas;
- As ferramentas utilizadas no desenvolvimento deste trabalho foram adequadas, facilitando e apoiando a construção do aplicativo.

Extensões

- A forma de troca de informações entre as linguagens de programação Delphi e Progress, que pode ser aprimorada utilizando arquivos eXtensible Markup Language (XML);
- A implementação de análise em mais de um nível, ou seja, programas chamados pelo programa em análise seriam analisados também;
- A conexão com vários bancos de dados (atualmente suporta somente um banco de dados);

Extensões

- A indicação ao usuário de qual dos comandos está executando leituras excessivas no banco de dados;
- A alteração automática do código fonte, conforme a necessidade, para a melhora de performance.