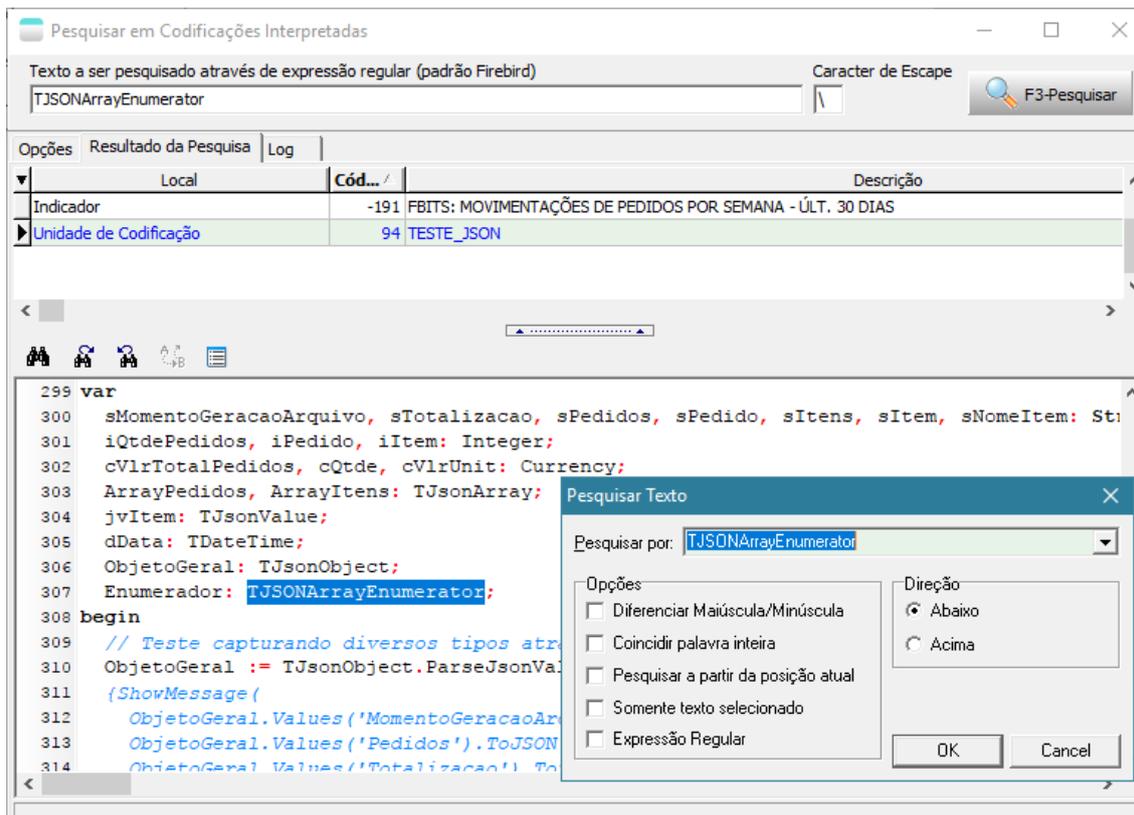
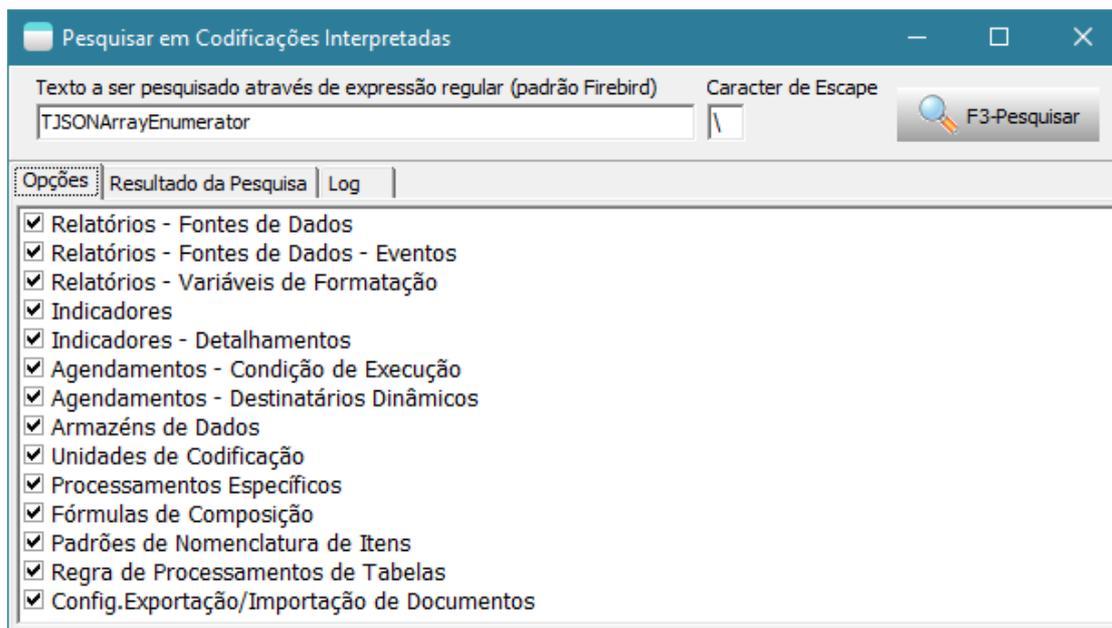
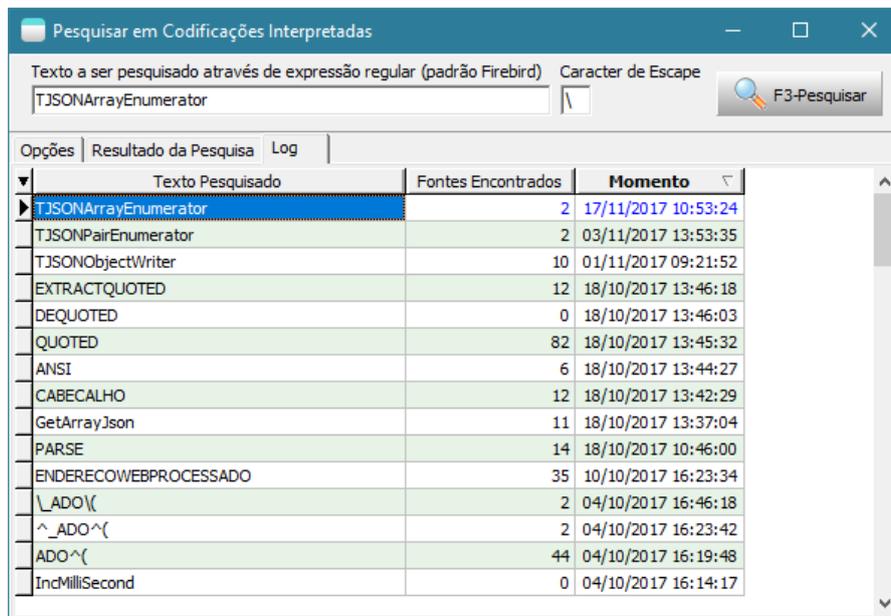


Novidades presentes no módulo de BI (Inteligência de Negócios e Gerador de Relatórios) da Tek-System

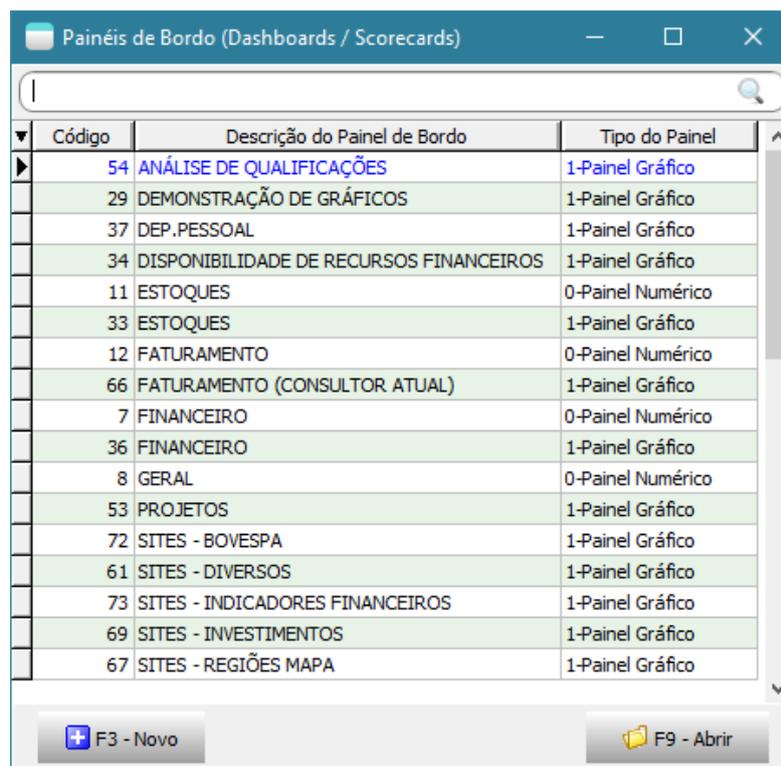
Inovações

- Criada funcionalidade que permite **pesquisar em todas as codificações interpretadas**. Muito útil para poder se lembrar de como é a sintaxe (uso) de um determinado comando ou rastrear uso de recursos compartilhados como variáveis e funções. Está disponível no menu processamentos. E ainda mantém o log das últimas pesquisas. Esta pesquisa utiliza expressões regulares no padrão do Firebird. O usuário poderá ter acesso à documentação oficial pressionando-se F1 quando o cursor estiver no campo do texto a ser pesquisado.

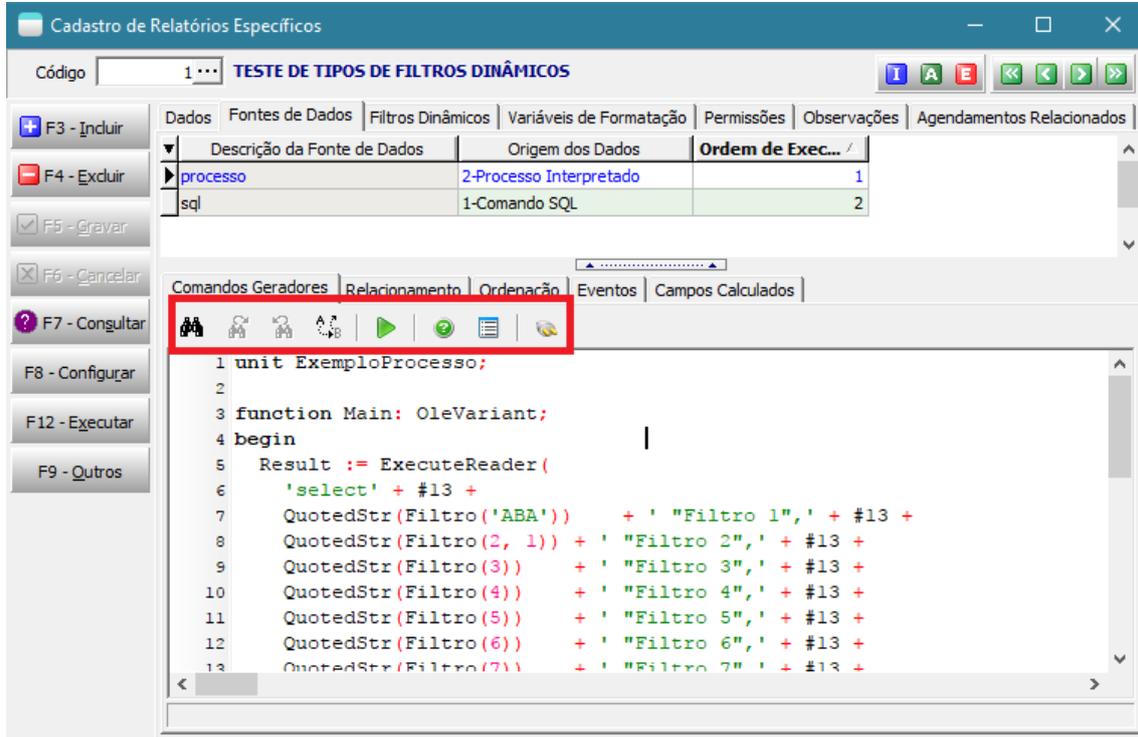




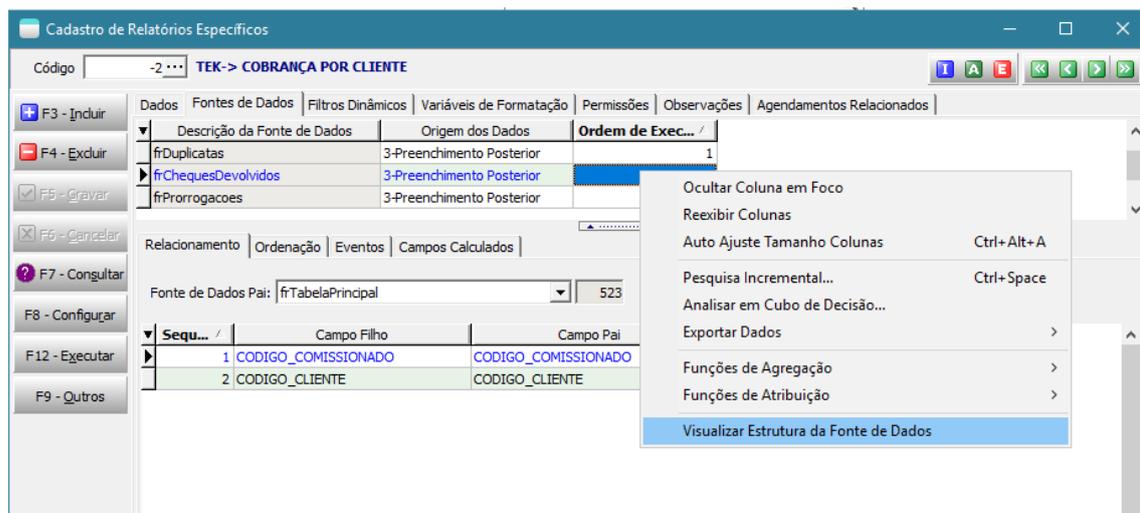
- Criada nova tela de abertura para selecionar os painéis de bordo. Padronizando com as telas que chamam processamentos e relatórios específicos e também armazéns de dados.



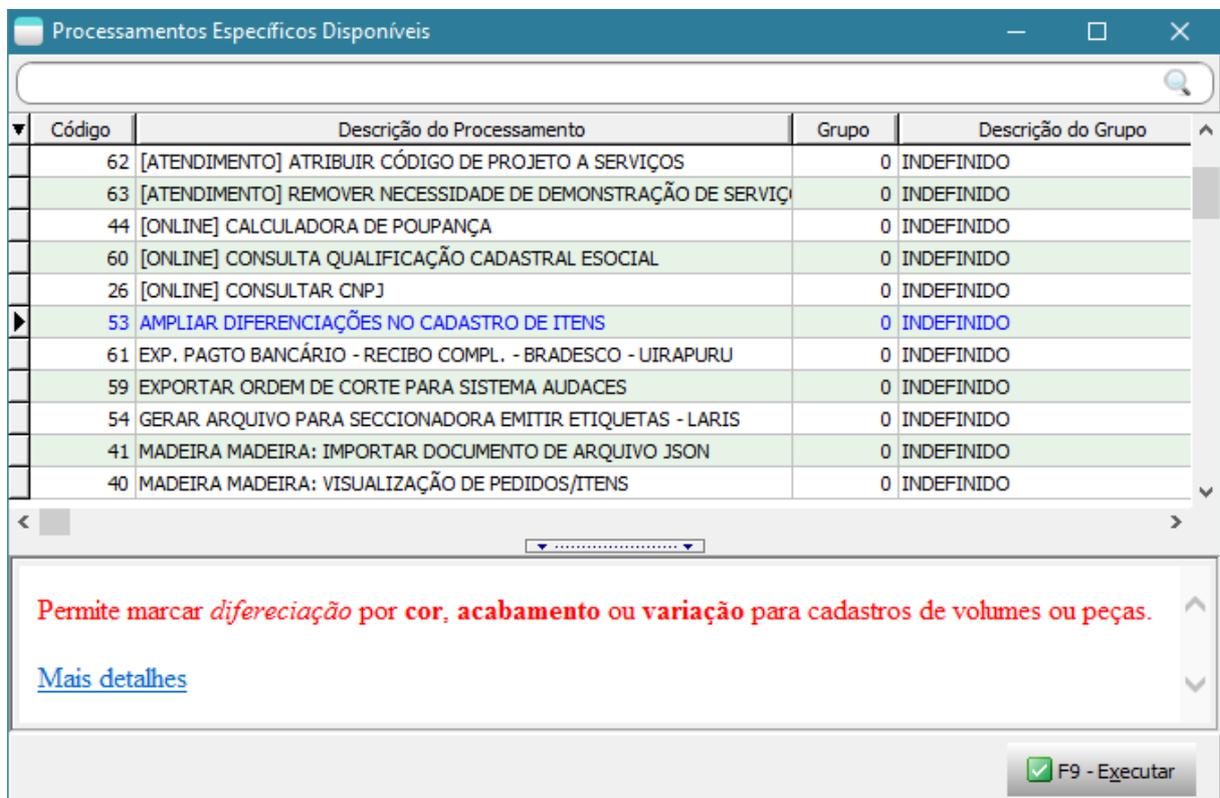
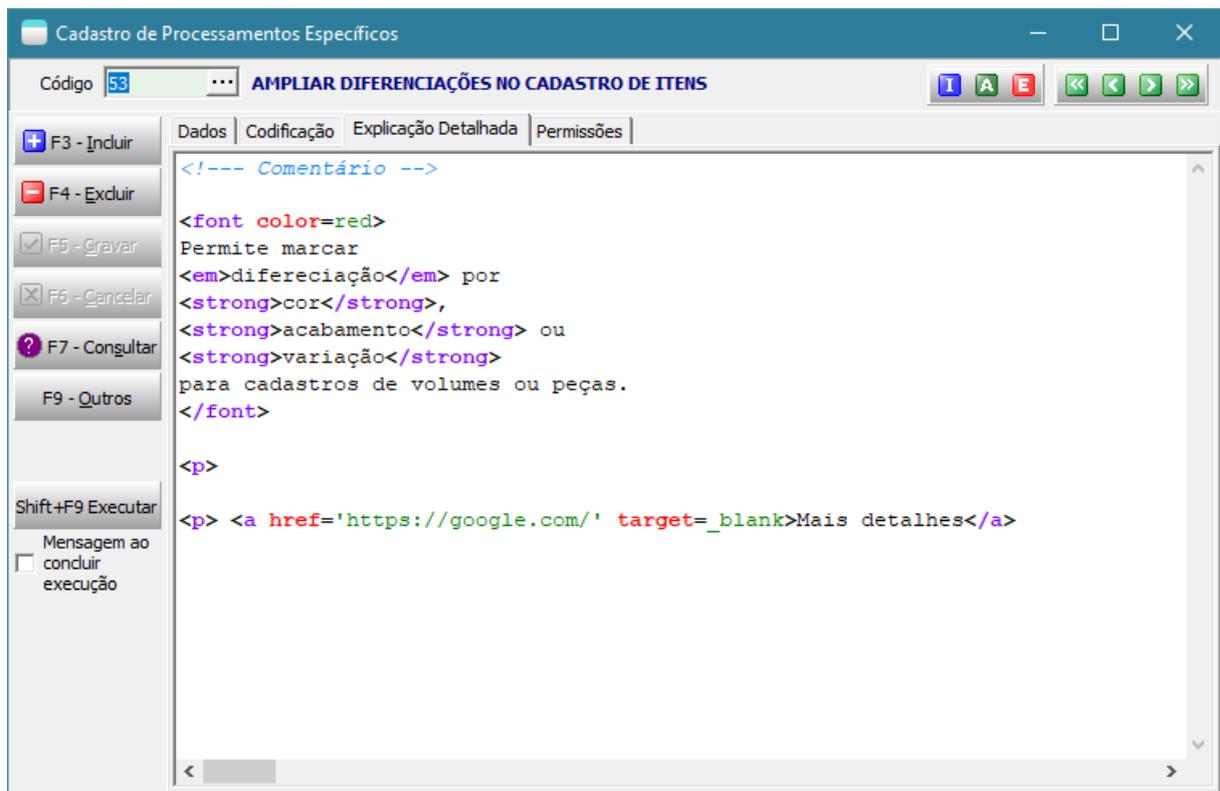
- Criado frame próprio para edição das codificações interpretadas, onde será possível, por enquanto: localizar texto, localizar próxima ocorrência, localizar ocorrência anterior, substituir textos, testar/executar, Ajuda para Compôr, Lista de Funções/Procedimentos, Fontes de Dados ODBC. Alguns botões podem estar visíveis ou habilitados de acordo com condições específicas do local onde se encontra a codificação.



- Visando melhorar a interface, foi ocultada a aba de comandos geradores quando a fonte de dados for do tipo "3-Preenchimento Posterior". Devido a esta mudança, foi necessário criar a opção "Visualizar Estrutura da Fonte de Dados" no menu de contexto da grade de fontes de dados.

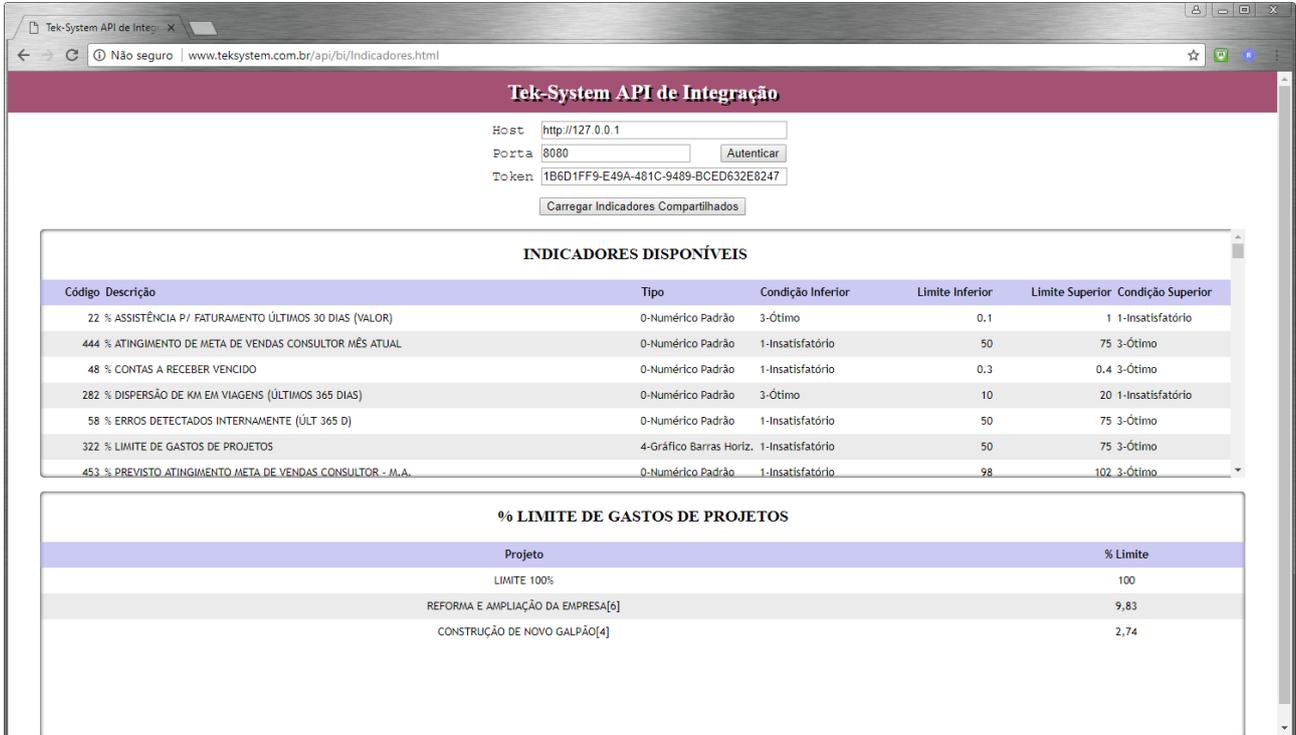


- Criada a possibilidade de digitação de tags HTML na descrição dos processamentos específicos. Desta forma pode-se dar mais destaques ou detalhes quanto às informações.



- Criada API que permitirá o compartilhamento de informações entre empresas parceiras através de dados gerados através de indicadores. Estes dados serão compartilhados através do WebserviceMC em formato JSON que poderão ser consumidos por qualquer linguagem: Delphi, Java, C++, JavaScript, PHP...

Foi desenvolvido um pequeno exemplo que já está on-line e funcional e se encontra em <http://www.teksystem.com.br/api/bi>



A empresa é quem decidirá quais dados deseja compartilhar e com quem. Pois para isto precisará informar para o parceiro (fornecedor, lojista, e-commerce, escritório, prestador de serviços...) o **host** onde se hospeda o WebserviceMC, a **porta** de conexão e também o **token** de usuário.



Além é claro de ter que dar permissão de acesso deste usuário ao indicador.

Para maiores detalhes técnicos da integração a Tek-system deverá ser consultada. Se as empresas envolvidas forem clientes Tek-System, então a integração poderá ser feita de forma mais facilitada ainda. Exibindo em uma empresa um indicador de outra (ver codificação de exemplo ao final deste documento).

- Criada possibilidade de na interpretação de codificação de detalhamento de indicadores realizar referência à codificação principal através do uses **UnitCodificacaoPrincipalDestelIndicador**. Isto evitará que para compartilhar uma codificação entre os detalhamentos dos indicadores seja necessário criar uma unit separada. Facilitando, por exemplo, o transporte de um indicador de um banco de dados para outro.

Detalhes:

- 1) É necessário salvar a codificação principal (cadastro do indicador) primeiro, para depois fazer uso dela. Pois o sistema buscará as informações já gravadas no banco de dados ao tentar testar os detalhamentos.
- 2) Apesar de na seção uses ser declarada como UnitCodificacaoPrincipalDestelIndicador, ao fazer uso das funções ou constantes compartilhadas, não será necessário indicar este nome na frente. Se, contudo, o usuário quiser fazer referência, deverá usar o nome da unit declarada na codificação principal.

Exemplo de codificação principal do indicador:

```
unit AcompanhamentoProdutividadeDesenvolvedor;

const CodDesenvolvedor = 13;

function Main: OleVariant;
begin
    Result := ExecuteReaderODBC('ATENDIMENTO',
        ' select' + #13 +
        '   extract(year from SERVICOS.DATAFIM_SVR) || '-' || lpad(extract(month from
SERVICOS.DATAFIM_SVR), 2, '0') "Ano-Mes",' + #13 +
        '   sum(SERVICOS.PESO_SVR) "Produtividade"' + #13 +
        ' from SERVICOS' + #13 +
        ' where SERVICOS.ATEND_FIM_SVR = ' + IntToStr(CodDesenvolvedor) + #13 +
        '   and SERVICOS.DATAFIM_SVR between '02/01/2017' and current_date + 1' + #13 +
        ' group by 1');
end;

function DetalharPeriodo(InicioPeriodo, FimPeriodo: TDateTime): OleVariant;
begin
    Result := ExecuteReaderODBC('ATENDIMENTO',
        ' select' + #13 +
        '   SERVICOS.DATAFIM_SVR           "Conclusão",' + #13 +
        '   PESO_SERVICO.DESCRICAO_PES     "Classificação",' + #13 +
        '   SERVICOS.PESO_SVR              "Produtividade",' + #13 +
        '   SERVICOS.AUTOINC_SVR           "Serviço",' + #13 +
        '   SERVICOS.CLIENTE_SVR           "Cliente",' + #13 +
        '   CLIENTES.RAZAO_CLI              "Nome do cliente",' + #13 +
        '   SERVICOS.SISTEMA_SVR           "Sistema",' + #13 +
        '   SISTEMAS.NOME_SIS               "Descrição do Sistema",' + #13 +
        '   MODULOS.DESCRICAO_MOD          "Módulo",' + #13 +
        '   SERVICOS.AREA_SVR              "Área",' + #13 +
        '   SERVICOS.DESCRICAO_SVR         "Descrição do Serviço",' + #13 +
        '   SERVICOS.JUSTIFICATIVA_SVR     "Justificativa",' + #13 +
        '   SERVICOS.OBSERVACAO_SVR        "Realização"' + #13 +
        ' from SERVICOS' + #13 +
        ' left join SISTEMAS      on (SISTEMAS.CODIGO_SIS      = SERVICOS.SISTEMA_SVR)' + #13 +
        ' left join MODULOS       on (MODULOS.CODIGO_MOD       = SERVICOS.MODULO_SVR)' + #13 +
        ' left join CLIENTES      on (SERVICOS.CLIENTE_SVR     = CLIENTES.CODIGO_CLI)' + #13 +
        ' left join PESO_SERVICO  on (PESO_SERVICO.CODIGO_PES = CODPESO_SVR)' + #13 +
        ' where SERVICOS.ATEND_FIM_SVR = ' + IntToStr(CodDesenvolvedor) + #13 +
        '   and SERVICOS.DATAFIM_SVR between ' + DataSQL(InicioPeriodo, 1) + ' and ' +
DataSQL(FimPeriodo, 2) + #13 +
        ' order by 1');
end;

end.
```

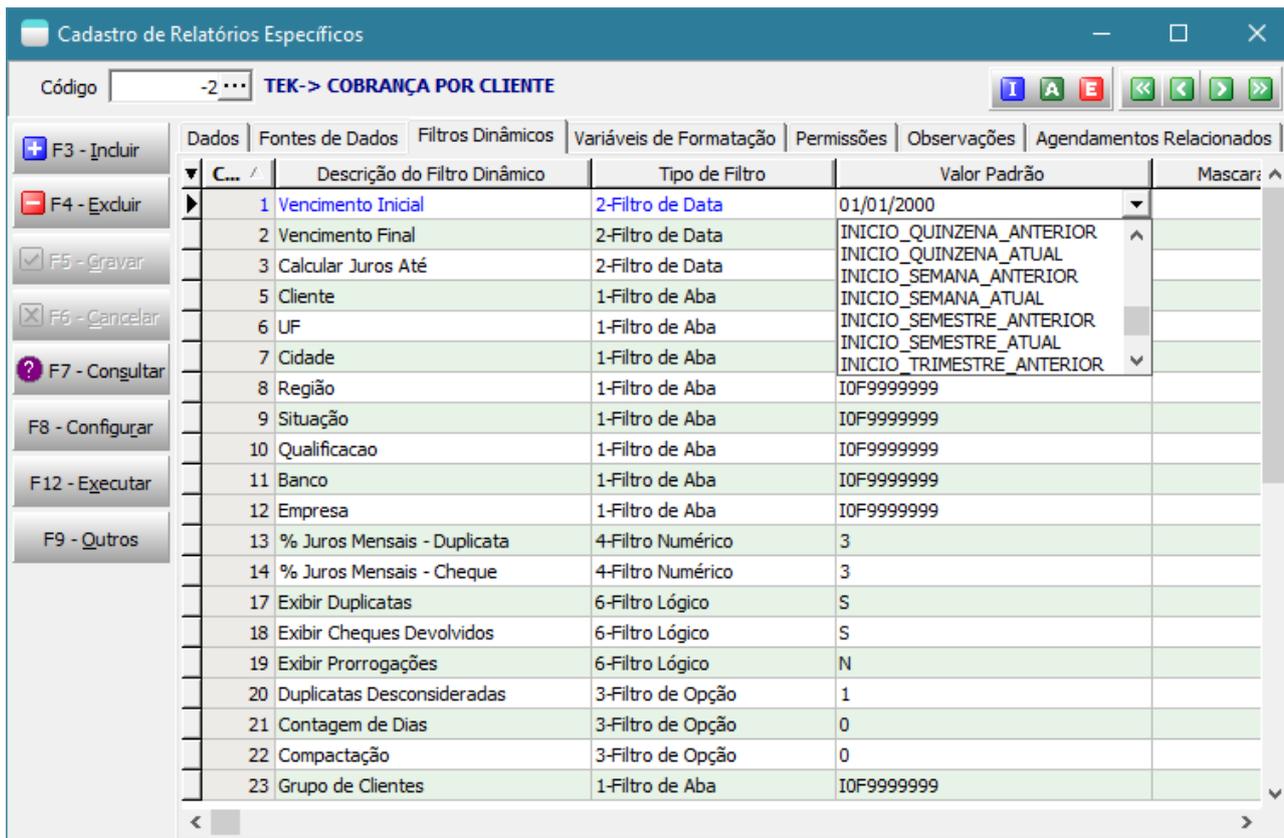
Exemplo de codificação de detalhamento do indicador:

```
unit DetalharProdutividadeDesenvolvedorMesAtual;
uses UnitCodificacaoPrincipalDesteIndicador;

function Main: OleVariant;
begin
    Result := DetalharPeriodo(INICIO_MES_ATUAL, FIM_MES_ATUAL);
end;

end.
```

- Novas constantes de macro substituição dos valores padrões em filtros dinâmicos:



ANTEONTEM

INICIO_SEMESTRE_ANTERIOR
 FIM_SEMESTRE_ANTERIOR
 INICIO_SEMESTRE_ATUAL
 FIM_SEMESTRE_ATUAL

INICIO_BIMESTRE_ANTERIOR
 FIM_BIMESTRE_ANTERIOR
 INICIO_BIMESTRE_ATUAL
 FIM_BIMESTRE_ATUAL

INICIO_TRIMESTRE_ANTERIOR
 FIM_TRIMESTRE_ANTERIOR
 INICIO_TRIMESTRE_ATUAL
 FIM_TRIMESTRE_ATUAL

INICIO_QUINZENA_ANTERIOR
 FIM_QUINZENA_ANTERIOR
 INICIO_QUINZENA_ATUAL
 FIM_QUINZENA_ATUAL

Novos modelos de indicadores padrões disponibilizados

FAT-> ITENS ASSISTÊNCIA - MÊS ANTERIOR - POR ASSISTÊNCIA

FAT-> ITENS ASSISTÊNCIA - MÊS ANTERIOR - POR CLIENTE

FAT-> ITENS ASSISTÊNCIA - MÊS ANTERIOR - POR CONSULTOR

FAT-> ITENS ASSISTÊNCIA - MÊS ANTERIOR - POR PEÇA

FAT-> ITENS ASSISTÊNCIA - MÊS ANTERIOR - POR PRODUTO

FAT-> ITENS ASSISTÊNCIA - MÊS ATUAL - POR ASSISTÊNCIA

FAT-> ITENS ASSISTÊNCIA - MÊS ATUAL - POR CLIENTE

FAT-> ITENS ASSISTÊNCIA - MÊS ATUAL - POR CONSULTOR

FAT-> ITENS ASSISTÊNCIA - MÊS ATUAL - POR PEÇA

FAT-> ITENS ASSISTÊNCIA - MÊS ATUAL - POR PRODUTO

FAT-> QTDE.ITENS ASSISTÊNCIA - MÊS ANTERIOR

FAT-> QTDE.ITENS ASSISTÊNCIA - MÊS ATUAL

FAT-> VLR.ASSISTÊNCIA TÉCNICA - MÊS ANTERIOR

FAT-> VLR.ASSISTÊNCIA TÉCNICA - MÊS ATUAL

FAT-> VLR.ASSISTÊNCIA TÉCNICA - ÚLT.30 DIAS

FAT-> VLR.VENDAS - MÊS ATUAL - POR UF

FAT-> VLR.VENDAS - MÊS ATUAL - POR CIDADE

FAT-> VLR.FATURAMENTO - MÊS ATUAL - POR UF

FAT-> VLR.FATURAMENTO - MÊS ATUAL - POR CIDADE

MADEIRA MADEIRA: QTDE DOC. SEM PEDIDO DE AUTORIZAÇÃO P/ FAT

MADEIRA MADEIRA: QTDE DOC. AGUARDANDO AUTORIZAÇÃO P/ FAT

MADEIRA MADEIRA: QTDE DOC. FATURAMENTO PENDENTE PELA EMPRESA

MAGENTO: DASHBOARD PEDIDOS

FBITS: DASHBOARD PEDIDOS

FBITS: MOVIMENTAÇÕES DE PEDIDOS POR SEMANA - ÚLT. 30 DIAS

ODBC-> % CLIENTES SEM COMPRA - ÚLT.90 DIAS (COMÉRCIO)

ODBC-> VLR.VENDAS MENOS VLR.COMPRAS - MÊS ATUAL (COMÉRCIO)

ODBC-> VLR.VENDAS MENOS DEVOLUÇÕES - MÊS ATUAL (COMÉRCIO)

Novos modelos de relatórios padrões que podem ser agendados

TEK-> VENDAS POR CIDADE

TEK-> FATURAMENTO POR CIDADE

TEK-> FATURAMENTO POR LINHA

TEK-> ESTATÍSTICAS DE METAS DE VENDAS

TEK-> ODBC-> COTAÇÃO X PEDIDOS DE VENDA (COMÉRCIO)

Novos modelos de Armazéns de dados

TEK-> FAT: VENDAS DEDUZINDO CANCELAMENTOS

TEK-> FAT: VENDAS DEDUZINDO CANCELAMENTOS E DEVOLUÇÕES

TEK-> FAT: FATURAMENTO DEDUZINDO DEVOLUÇÕES

TEK-> PCP: MATERIAIS PREVISTOS P/CONSUMO EM OP INICIADA

TEK-> EST: ESTATÍSTICAS DE COMPRAS

TEK-> ODBC-> ENTRADA DE MERCADORIAS (COMÉRCIO)

Novos modelos de Processamentos Específicos

TEK-> HERDAR INDICADORES TEK-SYSTEM

TEK-> BIESSE: GERAR ARQUIVO PLANO DE CORTE OPTPLANNING

Unidade de codificação ajustada/disponibilizada

TEK_CHART_GOOGLE_GEO: De acordo com nova API da Google que permite criar indicadores gráficos com as regiões do Brasil.

TEK_INTEGRACOES: Possui funções úteis comuns em integrações de dados.

TEK_TRANSMISSAO_NFE: Possui função que dispara chamadas para transmissão de NFe.

TEK_COMERCIO_INTEGRACAO: Possui funções úteis para integração com o sistema Comercial da Tek-System.

Novas classes e funções disponibilizadas para interpretação

Arredonde: Permite realizar arredondamentos numéricos conforme norma ABNT.

ArredondarProximoInteiro: Permite realizar arredondamento numéricos para o próximo inteiro.

Confirma: Permite abrir um diálogo para pergunta ao usuário. Muito útil nos processamentos interpretados.

OleVariantParaStream: Permite converter um resultado OleVariant (normalmente vindo do servidor de aplicação) para um Stream. Utilizado, por exemplo, para salvar um relatório processado.

StreamParaOleVariant: Permite converter de uma Stream para um OleVariant. Utilizado, por exemplo, para salvar um arquivo magnético dentro do banco de dados.

ShellExecute: Permite executar chamadas a Shell do Windows para, por exemplo, abrir uma pasta, executar um aplicativo, abrir um arquivo com extensão associada, abrir um link com o navegador padrão, enviar um e-mail usando o gerenciador de e-mail padrão.

CompactarString: Permite compactar uma string utilizando compactação zip. Normalmente utilizada para gravar arquivos XML no banco de dados.

DescompactarString: Permite descompactar uma string que foi previamente compactada com compactação zip.

UrlDownloadToMemoryStream: Permite baixar um arquivo da internet para uma variável Stream. Utilizado em algumas integrações.

EncodeMemoryStreamToBase64: Permite fazer o encode de uma variável stream para base 64. Utilizado em algumas integrações.

Novas constantes disponibilizadas para interpretação

StatusInativo: Código do status configurado como inativo no parâmetro da empresa.

SecaoParametroJson, SecaoEmpresaJson, SecaoUsuarioJson: Contêm informações da conexão atual em formato JSON para serem usados nos processamentos evitando ter que fazer comandos SQL para leitura do parâmetro.

Integrações Homologadas

Cobradas à parte pela Tek-System – Por isto suas codificações não estão disponíveis.

- Madeira Madeira: Site de vendas de móveis.
- Magento: Sistema web de e-commerce de código aberto baseado em PHP e MySQL.
- SkyHub: Centralizador para integração de venda em diversos marketplaces tais como Submarino, Walmart, Americanas, Ponto Frio, CDDiscount, Mercado Livre, Extra, Zoom, Casas Bahia, Shoptime.
- FBits: Plataforma de e-commerce corporativa.

Codificações para testes das classes/funções

- Exemplo de codificação para executar um relatório no meio de um processamento e salvá-lo em diversos formatos. Os filtros poderiam ser passados automaticamente também. Uso da função **OleVariantParaStream**.

```
unit ProcessamentoEspecifico;

var CDSFiltros: TClientDataSet;

const
  RelatorioASerProcessado = 1;
  CaminhoSalvamentoArquivo = 'c:\temp';

procedure Main;
var
  Rel_Ole: OleVariant;
  MS: TMemoryStream;
  Extensao: String;
begin
  CDSFiltros := TClientDataSet.Create;
  MS := TMemoryStream.Create;
  try
    if ConfirmarFiltros then
      begin
        Rel_Ole := ExecuteMethods('TSMcadGR_Relatorio.RelatorioPronto',
[RelatorioASerProcessado, Filtro('Extensão'), CDSFiltros.Data]);
        OleVariantParaStream(Rel_Ole, MS);
        case Filtro('Extensão') of
          0: Extensao := 'pdf';
          1: Extensao := 'fp3';
          2: Extensao := 'txt';
          3: Extensao := 'html';
          4: Extensao := 'jpeg';
          5: Extensao := 'CSV';
        end;
        MS.SaveToFile(CaminhoSalvamentoArquivo + '\relatorio.' + Extensao);
        ShowMessage('Arquivo criado com sucesso');
      end;
    finally
      MS.Free;
      CDSFiltros.Free;
    end;
  end;
end;

function ConfirmarFiltros: Boolean;
begin
  CDSFiltros.Data := EstruturaDeFiltrosDinamicos;
  // Filtros do relatório
  {01} IncluirFiltroDinamico(CDSFiltros, 'ABA', cTipoFiltro_Aba, 'I0F0', '', 'SBanco',
'');
  {02} IncluirFiltroDinamico(CDSFiltros, 'DATA', cTipoFiltro_Data, '22/12/2015', '', '',
'');
  {03} IncluirFiltroDinamico(CDSFiltros, 'OPCAO', cTipoFiltro_Opcao, '0', '', '',
'0'#13'1'#13'2');
  {04} IncluirFiltroDinamico(CDSFiltros, 'NUMERICO', cTipoFiltro_Numerico, '0', '', '',
'');
  {05} IncluirFiltroDinamico(CDSFiltros, 'TEXT0', cTipoFiltro_Texto, 'Abc', '', '', '');
  {06} IncluirFiltroDinamico(CDSFiltros, 'LOGICO', cTipoFiltro_Logico, 'N', '', '', '');

  // Filtros do processamento
  {07} IncluirFiltroDinamico(CDSFiltros, 'Extensão', cTipoFiltro_Opcao, '0', '', '',
'pdf'#13'fr'#13'txt'#13'html'#13'jpeg'#13'CSV');

  CDSFiltros.Data := ExecutarFiltroDinamico(CDSFiltros.Data);

  Result := (not CDSFiltros.IsEmpty);
end;
end.
```

- Exemplo de uso da função **StreamParaOleVariant** anexando um arquivo ao pedido.

```
procedure Main;
var MS: TMemoryStream;
begin
  MS := TMemoryStream.Create;
  try
    MS.LoadFromFile('C:\Tek-System\Atendimento\temp\atraente.jpg');
    ExecuteMethods('TSMCadDocumentoBase.AnexarArquivo',
      [93939, 'contrato.JPG', StreamParaOleVariant(MS)]);
  finally
    MS.Free;
  end;
end;
```

- Exemplo de uso das funções **CompactarString** e **DescompactarString**.

```
procedure Main;
var
  S1: String;
  S2: OleVariant;
begin
  S1 := 'Texto sem compactação';
  S2 := CompactarString(S1);

  ShowMessage(S1);
  ShowMessage(S2);
  ShowMessage(DescompactarString(S2));
end;
```

- Exemplo de chamadas de **execução da Shell do Windows**, permitindo executar protocolos associados, aplicativos externos e abrir arquivos cujas extensões estejam registradas no Windows:

```
procedure Main;
begin
  ShellExecute('mailto:denisuba@gmail.com'); // abre o gerenciador de email padrão.
  ShellExecute('http://www.teksystem.com.br/'); // abre um site no browser padrão
  ShellExecute('c:\temp'); // abre a pasta temp
  ShellExecute('c:\Temp\relatorio.pdf'); // abre um arquivo com extensão
  associada.
  ShellExecute('c:\util\PrevisaoRelatorio.exe'); // Executa um aplicativo externo.
end;
```

- Exemplo de uso das funções **UrlDownloadToMemoryStream** e **EncodeMemoryStreamToBase64**.

```
procedure Main;
const link = 'http://www.teksystem.com.br/site/images/logo_tek_system.png';
begin
  MostrarLogTexto(ConverterURLImagemEmBase64(link));
end;

function ConverterURLImagemEmBase64(url_imagem: String): String;
var Stream: TMemoryStream;
begin
  Stream := TMemoryStream.Create;
  try
    UrlDownloadToMemoryStream(url_imagem, Stream);
    Result := Troca(EncodeMemoryStreamToBase64(Stream), #13#10, '');
  finally
    Stream.Free;
  end;
end;
```

- Exemplo de uso de novas **funções / constantes relacionados ao tempo**.

```

procedure Main;
begin
  AbrirWebBrowser (
    'Semestre: ' + IntToStr(SemestreDoAno(Date)) + '<br>' +
    'Trimestre: ' + IntToStr(TrimestreDoAno(Date)) + '<br>' +
    'Bimestre: ' + IntToStr(BimestreDoAno(Date)) + '<br>' +
    'Quinzena: ' + IntToStr(QuinzenaDoMes(Date)) + '<br>'
    '<br>' +
    'InicioDoSemestre: ' + DateToStr(InicioDoSemestre(Date)) + '<br>'
    'InicioDoTrimestre: ' + DateToStr(InicioDoTrimestre(Date)) + '<br>'
    'InicioDoBimestre: ' + DateToStr(InicioDoBimestre(Date)) + '<br>'
    'InicioDaQuinzena: ' + DateToStr(InicioDaQuinzena(Date)) + '<br>'
    '<br>' +
    'FimDoSemestre: ' + DateToStr(FimDoSemestre(Date)) + '<br>'
    'FimDoTrimestre: ' + DateToStr(FimDoTrimestre(Date)) + '<br>'
    'FimDoBimestre: ' + DateToStr(FimDoBimestre(Date)) + '<br>'
    'FimDaQuinzena: ' + DateToStr(FimDaQuinzena(Date)) + '<br>'
    '<br>' +
    'INICIO_SEMESTRE_ANTERIOR: ' + DateToStr(INICIO_SEMESTRE_ANTERIOR) + '<br>' +
    'FIM_SEMESTRE_ANTERIOR: ' + DateToStr(FIM_SEMESTRE_ANTERIOR) + '<br>' +
    'INICIO_SEMESTRE_ATUAL: ' + DateToStr(INICIO_SEMESTRE_ATUAL) + '<br>' +
    'FIM_SEMESTRE_ATUAL: ' + DateToStr(FIM_SEMESTRE_ATUAL) + '<br>' +
    '<br>' +
    'INICIO_TRIMESTRE_ANTERIOR: ' + DateToStr(INICIO_TRIMESTRE_ANTERIOR) + '<br>' +
    'FIM_TRIMESTRE_ANTERIOR: ' + DateToStr(FIM_TRIMESTRE_ANTERIOR) + '<br>' +
    'INICIO_TRIMESTRE_ATUAL: ' + DateToStr(INICIO_TRIMESTRE_ATUAL) + '<br>' +
    'FIM_TRIMESTRE_ATUAL: ' + DateToStr(FIM_TRIMESTRE_ATUAL) + '<br>' +
    '<br>' +
    'INICIO_BIMESTRE_ANTERIOR: ' + DateToStr(INICIO_BIMESTRE_ANTERIOR) + '<br>' +
    'FIM_BIMESTRE_ANTERIOR: ' + DateToStr(FIM_BIMESTRE_ANTERIOR) + '<br>' +
    'INICIO_BIMESTRE_ATUAL: ' + DateToStr(INICIO_BIMESTRE_ATUAL) + '<br>' +
    'FIM_BIMESTRE_ATUAL: ' + DateToStr(FIM_BIMESTRE_ATUAL) + '<br>' +
    '<br>' +
    'INICIO_QUINZENA_ANTERIOR: ' + DateToStr(INICIO_QUINZENA_ANTERIOR) + '<br>' +
    'FIM_QUINZENA_ANTERIOR: ' + DateToStr(FIM_QUINZENA_ANTERIOR) + '<br>' +
    'INICIO_QUINZENA_ATUAL: ' + DateToStr(INICIO_QUINZENA_ATUAL) + '<br>' +
    'FIM_QUINZENA_ATUAL: ' + DateToStr(FIM_QUINZENA_ATUAL));
end;

```

- Exemplo de processamento que executa um site com carga tardia criando um botão para **autopreenchimento dos campos usando JavaScript**.

```

unit ProcessamentoEspecifico;

procedure Main;
begin
  AbrirWebBrowser (
    'https://www3.bcb.gov.br/CALCIDADA0/publico/exibirFormCorrecaoValores.do?method=exibirFormCorrecaoValores&aba=3',
    '',
    False,
    'Calculadora de Poupança',
    'document.getElementsByName("dataInicial")[0].value = "15/12/2015";' + #13 +
    'document.getElementsByName("dataFinal")[0].value = "15/12/2016";' + #13 +
    'document.getElementsByName("valorCorrecao")[0].value = 100000.00;' + #13 +
    'document.getElementsByName("regraNova")[0].checked = true;' + #13 +
    'document.forms[0].submit();');
end;
end.

```

- Exemplo de **chamada da API de Integração de dentro do sistema de BI**. Por exemplo, em um indicador que exibirá informações de outra empresa parceira.

```

procedure Main;

const
  HostPorta           = 'http://127.0.0.1:8080';
  CodIndicadorInteresse = {68} {322} {660} 991;
  Token              = '1B6D1FF9-E49A-481C-9489-BCED632E8247';

var sJSON, sAutenticacao, sIndicadores, sTipoIndicador, sValorIndicador: String;
begin
  // 1) Buscar a autenticação através do token:
  sJSON := EnderecoWebProcessado2(
    HostPorta + '/auth/login/bi/1',
    'GET', '', 0, 0,
    'token: ' + Token);
  sAutenticacao := ExtractQuotedString(sJSON, '');

  // 2) Buscar lista de indicadores:
  sJSON := EnderecoWebProcessado2(
    HostPorta + '/api/bi/indicadores',
    'GET', '', 0, 0,
    'Authorization: ' + sAutenticacao);
  sIndicadores := TJSONObject.ParseJSONValue(sJSON).ToString;
  MostrarListaIndicadores(sIndicadores);

  // 3) Buscar o valor de um indicador:
  sJSON := EnderecoWebProcessado2(
    HostPorta + '/api/bi/indicadores/' + IntToStr(CodIndicadorInteresse),
    'POST', '{"cnpj_cpf":"751.481.536-87"}', 0, 0,
    'Authorization: ' + sAutenticacao);
  sJSON := TJSONObject.ParseJSONValue(sJSON).ToString;
  sTipoIndicador := GetValueJson(sJSON, 'tipo');
  sValorIndicador := GetValueJson(sJSON, 'valor');
  if Pos(sTipoIndicador, '3, 4, 5, 6, 7, 8') > 0 then
  begin
    sValorIndicador := JSONFormatado(sValorIndicador);
    MostrarLogTexto(sValorIndicador);
  end
  else
    AbrirWebBrowser(sValorIndicador);
end;

procedure MostrarListaIndicadores(JSON: String);
var CDS: TClientDataSet;
begin
  // MostrarLogTexto(JSONFormatado(JSON));
  CDS := TClientDataSet.Create;
  try
    CDS.FieldDefs.Clear;
    CDS.FieldDefs.Add('codigo',          ftInteger, 0, False);
    CDS.FieldDefs.Add('descricao',      ftString, 60, False);
    CDS.FieldDefs.Add('tipo',           ftInteger, 0, False);
    CDS.FieldDefs.Add('limiteInferior',  ftCurrency, 0, False);
    CDS.FieldDefs.Add('limiteSuperior',  ftCurrency, 0, False);
    CDS.FieldDefs.Add('condicaoInferior', ftInteger, 0, False);
    CDS.FieldDefs.Add('condicaoSuperior', ftInteger, 0, False);
    CDS.CreateDataSet;

    JSON := '{"indicadores": ' + JSON + '>';

    JSontoCDS(JSON, [CDS], 'indicadores');

    MostrarCDS(CDS);
  finally
    CDS.Free;
  end;
end;
end;

```

Denis Pereira Raymundo

Certified Delphi Developer
Professional Coach of Life Coaching
Especialista em Gestão e Manutenção de Tecnologia da Informação
Bacharel em Ciência da Computação
Licenciado em Matemática
Técnico em Processamento de Dados

Gerente de Sistemas

www.teksystem.com.br

Prêmios: Top Móbile - Segmento: Fornecedores de Softwares p/Setor Moveleiro

- 1ª lugar (2013)

- 2ª lugar (2012, 2014, 2015 e 2016)

- 3ª lugar (2009)



"O seu Deus o ensina e o instrui acerca do que há de fazer" Is 28.26