



Manual de Administração

Produto: XML Capture		Versão do produto: 4.2.0
Autor: Chayane C. Costa Santos Schmidt Vieira		
Versão do documento: 1.0	Versão do <i>template</i> : 2	Data: 20/02/2013
Documento destinado a: Parceiros NDDigital, técnicos de suporte, analista de T.I.		

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

Histórico de alterações

Data	Versão	Autor	Descrição
20/02/2013	1.0	Chayane C. Costa Santos Schmidt Vieira	Elaboração do documento

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

Índice

1.	Introdução.....	4
2.	Criação das Tabelas	4
2.1	Criação das Tabelas em MS SQL 2008	4
2.2	Criação das Tabelas em Oracle 10g.....	11
3.	Configurações	18
3.1	Editando as Configurações do Serviço de Processamento	19
3.2	Editando as Configurações do Serviço de Monitoramento (Guardian)	19

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

1. Introdução

A solução NDDigital XML Capture 4.2.0 é uma interface *Web* que possibilita obter o *status*, bem como, o XML de Notas Fiscais Eletrônicas, através de parametrizações de consultas. Com essa ferramenta você pode definir o perfil dos usuários (administradores / operacionais), e os locais aonde serão buscados os documentos por ele consultados (SEFAZ / Cold / Robô NDD), além de estabelecer privilégios para visualização e *download* de documentos, assim como, para a geração de relatórios.

Este manual visa auxiliar no processo de administração e configuração do produto.

2. Criação das Tabelas

Para atender a solução NDDigital e-Forms XML Capture é necessária a criação de algumas tabelas específicas para seu funcionamento, cujos *scripts* seguem abaixo:

2.1 Criação das Tabelas em MS SQL 2008

O *script* a seguir deve ser executado junto à base de dados do e-Connector, caso a mesma utilize o SGBD MS SQL 2008.

```

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('TBSTATISTICSCONSULT') AND OBJECTPROPERTY(id, 'IsUserTable') = 1)
    DROP TABLE TBSTATISTICSCONSULT
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('FK_TBCaptureAgent_TBCaptureToken') AND OBJECTPROPERTY(id, 'IsForeignKey') = 1)
    ALTER TABLE TBCaptureAgent DROP CONSTRAINT FK_TBCaptureAgent_TBCaptureToken
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('FK_TBCaptureAgent_TBCAPTURESEARCH') AND OBJECTPROPERTY(id, 'IsForeignKey') = 1)
    ALTER TABLE TBCaptureAgent DROP CONSTRAINT FK_TBCaptureAgent_TBCAPTURESEARCH
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('FK_TBCaptureReport_TBCAPTUREUSER') AND OBJECTPROPERTY(id, 'IsForeignKey') = 1)
    ALTER TABLE TBCaptureReport DROP CONSTRAINT FK_TBCaptureReport_TBCAPTUREUSER
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('FK_TBCaptureReportWork_TBCaptureInstance') AND OBJECTPROPERTY(id, 'IsForeignKey') = 1)
    ALTER TABLE TBCaptureReportWork DROP CONSTRAINT FK_TBCaptureReportWork_TBCaptureInstance
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('FK_TBCaptureReportWork_TBCaptureReport') AND OBJECTPROPERTY(id, 'IsForeignKey') = 1)
    ALTER TABLE TBCaptureReportWork DROP CONSTRAINT FK_TBCaptureReportWork_TBCaptureReport
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('FK_TBCaptureSearch_TBCaptureToken') AND OBJECTPROPERTY(id, 'IsForeignKey') = 1)
    ALTER TABLE TBCaptureSearch DROP CONSTRAINT FK_TBCaptureSearch_TBCaptureToken
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('FK_TBCaptureSearch_TBCAPTUREUSER') AND OBJECTPROPERTY(id, 'IsForeignKey') = 1)
    ALTER TABLE TBCaptureSearch DROP CONSTRAINT FK_TBCaptureSearch_TBCAPTUREUSER
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('FK_TBCaptureSearchResultTBCaptureSearch') AND OBJECTPROPERTY(id, 'IsForeignKey') = 1)
    ALTER TABLE TBCaptureSearchResult DROP CONSTRAINT FK_TBCaptureSearchResultTBCaptureSearch
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('FK_TBCaptureSearchWork_TBCaptureInstance') AND OBJECTPROPERTY(id, 'IsForeignKey') = 1)
    ALTER TABLE TBCaptureSearchWork DROP CONSTRAINT FK_TBCaptureSearchWork_TBCaptureInstance
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('FK_TBCaptureSearchWork_TBCaptureSearch') AND OBJECTPROPERTY(id, 'IsForeignKey') = 1)
    ALTER TABLE TBCaptureSearchWork DROP CONSTRAINT FK_TBCaptureSearchWork_TBCaptureSearch

```

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

```

;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('FK_TBCaptureUser_TBCaptureToken') AND
OBJECTPROPERTY(id, 'IsForeignKey') = 1)
ALTER TABLE TBCaptureUser DROP CONSTRAINT FK_TBCaptureUser_TBCaptureToken
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('FK_TBCaptureView_TBCAPTURESEARCH') AND
OBJECTPROPERTY(id, 'IsForeignKey') = 1)
ALTER TABLE TBCaptureView DROP CONSTRAINT FK_TBCaptureView_TBCAPTURESEARCH
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('TBCaptureAgent') AND OBJECTPROPERTY(id,
'IsUserTable') = 1)
DROP TABLE TBCaptureAgent
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('TBCaptureInstance') AND OBJECTPROPERTY(id,
'IsUserTable') = 1)
DROP TABLE TBCaptureInstance
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('TBCaptureReport') AND OBJECTPROPERTY(id,
'IsUserTable') = 1)
DROP TABLE TBCaptureReport
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('TBCaptureReportWork') AND OBJECTPROPERTY(id,
'IsUserTable') = 1)
DROP TABLE TBCaptureReportWork
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('TBCaptureSearch') AND OBJECTPROPERTY(id,
'IsUserTable') = 1)
DROP TABLE TBCaptureSearch
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('TBCaptureSearchResult') AND OBJECTPROPERTY(id,
'IsUserTable') = 1)
DROP TABLE TBCaptureSearchResult
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('TBCaptureSearchWork') AND OBJECTPROPERTY(id,
'IsUserTable') = 1)
DROP TABLE TBCaptureSearchWork
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('TBCaptureToken') AND OBJECTPROPERTY(id,
'IsUserTable') = 1)
DROP TABLE TBCaptureToken
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('TBCaptureUser') AND OBJECTPROPERTY(id,
'IsUserTable') = 1)
DROP TABLE TBCaptureUser
;

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = object_id('TBCaptureView') AND OBJECTPROPERTY(id,
'IsUserTable') = 1)
DROP TABLE TBCaptureView
;

CREATE TABLE TBSTATISTICSCONSULT (
    STATISTICSID bigint identity(1,1) NOT NULL,
    CNPJ bigint NOT NULL,
    DOCTYPE tinyint NOT NULL,
    AMBIENTTYPE smallint NOT NULL,
    PROCSTATUS bit NOT NULL,
    DOCUMENTKEY1 bigint NOT NULL,
    DOCUMENTKEY2 bigint NOT NULL,
    DOCUMENTKEY3 int NOT NULL,
    CONSULTDATE datetime NOT NULL
)
;

CREATE TABLE TBCaptureAgent (
    ID bigint identity(1,1) NOT NULL,
    SEARCHID bigint NOT NULL,
    TOKENID decimal(24) NOT NULL,
    INSERTDATE datetime NOT NULL,
    XMLDOCUMENT varchar(max)
)

```

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

```

;

CREATE TABLE TBCaptureInstance (
    ID smallint identity(1,1) NOT NULL,
    SERVERID smallint,
    LASTUPDATE datetime
)
;

CREATE TABLE TBCaptureReport (
    ID bigint identity(1,1) NOT NULL,
    IDUSER int NOT NULL,
    INSERTDATE datetime NOT NULL,
    FINISHDATE datetime,
    XMLREPORTPROCESS varchar(max),
    XMLREPORTDATAAREPLY varchar(max),
    FILTERXML varchar(max) NOT NULL
)
;

CREATE TABLE TBCaptureReportWork (
    ID bigint identity(1,1) NOT NULL,
    REPORTID bigint NOT NULL,
    SERVERID smallint
)
;

CREATE TABLE TBCaptureSearch (
    ID bigint identity(1,1) NOT NULL,
    IDUSER int NOT NULL,
    INSERTDATE datetime NOT NULL,
    FINISHDATE datetime,
    DOCUMENTKEY1 bigint NOT NULL,
    DOCUMENTKEY2 bigint NOT NULL,
    DOCUMENTKEY3 int NOT NULL,
    XMLREQUESTREPLY varchar(max),
    TOKENID decimal(24) NOT NULL,
    SEARCHSTATUS tinyint NOT NULL,
    OPENSTATUS bit NOT NULL
)
;

CREATE TABLE TBCaptureSearchResult (
    ID bigint identity(1,1) NOT NULL,
    SEARCHID bigint NOT NULL,
    EMIT_CNPJ varchar(20) NOT NULL,
    EMIT_NAME varchar(60),
    EMIT_PHONE varchar(20),
    SearchedInSefaz bit,
    SearchedInCold bit,
    SearchedInRobot bit,
    ExistsDocInCold bit
)
;

CREATE TABLE TBCaptureSearchWork (
    ID bigint identity(1,1) NOT NULL,
    SEARCHID bigint NOT NULL,
    PROFILEXML varchar(max) NOT NULL,
    SERVERID smallint
)
;

CREATE TABLE TBCaptureToken (
    ID bigint identity(1,1) NOT NULL,
    TOKENID decimal(24) NOT NULL,
    TOKEN varchar(100) NOT NULL,
    LICENSEKEY varchar(max)
)
;

CREATE TABLE TBCaptureUser (
    ID int identity(1,1) NOT NULL,
    NAME varchar(50) NOT NULL,
    PASSWORD varchar(100) NOT NULL,
    PROFILETYPE bit NOT NULL,
    PROFILEXML varchar(max),
    TOKENID decimal(24) NOT NULL,
    RESETPASSWORD bit
)
;

```

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

```

CREATE TABLE TBCaptureView (
    ID bigint identity(1,1) NOT NULL,
    IDSEARCH bigint NOT NULL,
    XMLDOCUMENT varchar(max)
)
;

ALTER TABLE TBSTATISTICSCONSULT ADD CONSTRAINT PK_TBSTATISTICSCONSULT
    PRIMARY KEY CLUSTERED (STATISTICSID)
;

ALTER TABLE TBCaptureAgent ADD CONSTRAINT PK_TBCaptureAgent
    PRIMARY KEY CLUSTERED (ID)
;

ALTER TABLE TBCaptureInstance ADD CONSTRAINT PK_TBCaptureInstance
    PRIMARY KEY CLUSTERED (ID)
;

ALTER TABLE TBCaptureReport ADD CONSTRAINT PK_TBCaptureReport
    PRIMARY KEY CLUSTERED (ID)
;

ALTER TABLE TBCaptureReportWork ADD CONSTRAINT PK_TBCaptureReportWork
    PRIMARY KEY CLUSTERED (ID)
;

ALTER TABLE TBCaptureSearch ADD CONSTRAINT PK_TBCaptureSearch
    PRIMARY KEY CLUSTERED (ID)
;

ALTER TABLE TBCaptureSearchResult ADD CONSTRAINT PK_TBCaptureSearchResult
    PRIMARY KEY CLUSTERED (ID)
;

ALTER TABLE TBCaptureSearchWork ADD CONSTRAINT PK_TBCaptureSearchWork
    PRIMARY KEY CLUSTERED (ID)
;

ALTER TABLE TBCaptureToken ADD CONSTRAINT PK_TBCaptureToken
    PRIMARY KEY CLUSTERED (ID)
;

ALTER TABLE TBCaptureUser ADD CONSTRAINT PK_TBCaptureUser
    PRIMARY KEY CLUSTERED (ID)
;

ALTER TABLE TBCaptureView ADD CONSTRAINT PK_TBCaptureView
    PRIMARY KEY CLUSTERED (ID)
;

CREATE INDEX TBSTATISTICSCONSULT_IDX1
ON TBSTATISTICSCONSULT (CONSULTDATE ASC, PROCSTATUS ASC)
;

CREATE INDEX TBSTATISTICSCONSULTSTATUS_IDX1
ON TBSTATISTICSCONSULT (PROCSTATUS ASC)
;

CREATE NONCLUSTERED INDEX IX_INSERTDATEAGENT
ON TBCaptureAgent (INSERTDATE ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 98) ON [PRIMARY]
;

CREATE NONCLUSTERED INDEX IX_TOKENAGENT
ON TBCaptureAgent (TOKENID ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON, FILLFACTOR = 90) ON [PRIMARY]
;

CREATE NONCLUSTERED INDEX IX_SEARCHAGENT
ON TBCaptureAgent (SEARCHID ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB
= OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON, FILLFACTOR = 90) ON [PRIMARY]
;

CREATE UNIQUE INDEX IX_SERVERID
ON TBCaptureInstance (SERVERID ASC)
;

CREATE NONCLUSTERED INDEX IX_USERREPORT

```

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

```

ON TBCaptureReport (IDUSER ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
;

CREATE NONCLUSTERED INDEX IX_SERVERIDREPORTWORK
ON TBCaptureReportWork (SERVERID ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 98) ON [PRIMARY]
;

CREATE NONCLUSTERED INDEX IX_DOCUMENTKEYSEARCH
ON TBCaptureSearch (DOCUMENTKEY1 ASC, DOCUMENTKEY2 ASC, DOCUMENTKEY3 ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
;

CREATE NONCLUSTERED INDEX IX_TOKENSERACH
ON TBCaptureSearch (TOKENID ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 85) ON [PRIMARY]
;

CREATE NONCLUSTERED INDEX IX_SERVERIDSEARCHWORK
ON TBCaptureSearchWork (SERVERID ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
;

CREATE NONCLUSTERED INDEX IX_SEARCHIDSEARCHWORK
ON TBCaptureSearchWork (SEARCHID ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
;

CREATE NONCLUSTERED INDEX IX_SEARCHIDSEARCHRESULT
ON TBCaptureSearchResult (SEARCHID ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
;

CREATE NONCLUSTERED INDEX IX_REPORTIDREPORTWORK
ON TBCaptureReportWork (REPORTID ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
;

CREATE UNIQUE INDEX IX_TOKENID
ON TBCaptureToken (TOKENID ASC)
;

CREATE NONCLUSTERED INDEX IX_TOKEN
ON TBCaptureUser (TOKENID ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
;

CREATE NONCLUSTERED INDEX IX_SEARCHVIEW
ON TBCaptureView (IDSEARCH ASC) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, FILLFACTOR = 90) ON [PRIMARY]
;

ALTER TABLE TBCaptureAgent ADD CONSTRAINT FK_TBCaptureAgent_TBCaptureToken
FOREIGN KEY (TOKENID) REFERENCES TBCaptureToken (TOKENID)
;
ALTER TABLE TBCaptureAgent ADD CONSTRAINT FK_TBCaptureAgent_TBCAPTURESEARCH
FOREIGN KEY (SEARCHID) REFERENCES TBCaptureSearch (ID)
;

ALTER TABLE TBCaptureReport ADD CONSTRAINT FK_TBCaptureReport_TBCAPTUREUSER
FOREIGN KEY (IDUSER) REFERENCES TBCaptureUser (ID)
;

ALTER TABLE TBCaptureReportWork ADD CONSTRAINT FK_TBCaptureReportWork_TBCaptureInstance
FOREIGN KEY (SERVERID) REFERENCES TBCaptureInstance (SERVERID)
;

ALTER TABLE TBCaptureReportWork ADD CONSTRAINT FK_TBCaptureReportWork_TBCaptureReport
FOREIGN KEY (REPORTID) REFERENCES TBCaptureReport (ID)
;

```

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

```

ALTER TABLE TBCaptureSearch ADD CONSTRAINT FK_TBCaptureSearch_TBCaptureToken
FOREIGN KEY (TOKENID) REFERENCES TBCaptureToken (TOKENID)
;

ALTER TABLE TBCaptureSearch ADD CONSTRAINT FK_TBCaptureSearch_TBCAPTUREUSER
FOREIGN KEY (IDUSER) REFERENCES TBCaptureUser (ID)
;

ALTER TABLE TBCaptureSearchResult ADD CONSTRAINT FK_TBCaptureSearchResultTBCaptureSearch
FOREIGN KEY (SEARCHID) REFERENCES TBCaptureSearch (ID)
;

ALTER TABLE TBCaptureSearchWork ADD CONSTRAINT FK_TBCaptureSearchWork_TBCaptureInstance
FOREIGN KEY (SERVERID) REFERENCES TBCaptureInstance (SERVERID)
;

ALTER TABLE TBCaptureSearchWork ADD CONSTRAINT FK_TBCaptureSearchWork_TBCaptureSearch
FOREIGN KEY (SEARCHID) REFERENCES TBCaptureSearch (ID)
;

ALTER TABLE TBCaptureUser ADD CONSTRAINT FK_TBCaptureUser_TBCaptureToken
FOREIGN KEY (TOKENID) REFERENCES TBCaptureToken (TOKENID)
;

ALTER TABLE TBCaptureView ADD CONSTRAINT FK_TBCaptureView_TBCAPTURESEARCH
FOREIGN KEY (IDSEARCH) REFERENCES TBCaptureSearch (ID)
;

/***** Object: UserDefinedFunction [dbo].[CreateListTable] Script Date: 07/22/2011 16:08:06 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[CreateListTable]') AND type in
(N'FN', N'IF', N'TF', N'FS', N'FT'))
BEGIN
execute dbo.sp_executesql @statement = N'-- =====
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE FUNCTION [dbo].[CreateListTable] (@list nvarchar(MAX))
RETURNS @tbl TABLE (number bigint NOT NULL) AS
BEGIN
DECLARE @pos int,
        @nextpos int,
        @valuelen int

SELECT @pos = 0, @nextpos = 1

WHILE @nextpos > 0
BEGIN
SELECT @nextpos = charindex(",", @list, @pos + 1)
SELECT @valuelen = CASE WHEN @nextpos > 0
                        THEN @nextpos
                        ELSE len(@list) + 1
                    END - @pos - 1
INSERT @tbl (number)
VALUES (convert(bigint, substring(@list, @pos + 1, @valuelen)))
SELECT @pos = @nextpos
END
RETURN
END
GO
/***** Object: UserDefinedFunction [dbo].[CreateKeyTable] Script Date: 02/28/2011 14:50:06 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[CreateKeyTable]') AND type in
(N'FN', N'IF', N'TF', N'FS', N'FT'))
BEGIN
execute dbo.sp_executesql @statement = N'-- =====
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE FUNCTION [dbo].[CreateKeyTable] (@list nvarchar(MAX))
RETURNS @tbl TABLE (
dockey nvarchar(MAX) NOT NULL,

```

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

```

docprotocol bigint NOT NULL,
dckey01 bigint NOT NULL,
dckey02 bigint NOT NULL,
dckey03 bigint NOT NULL) AS
BEGIN
    DECLARE @pos      int,
            @nextpos  int,
            @valuelen int,
            @documentkey nvarchar(MAX),
            @documenttype nvarchar(2),
            @protocol bigint,
            @dockey01 bigint,
            @dockey02 bigint,
            @dockey03 bigint

    SELECT @pos = 0, @nextpos = 1

    WHILE @nextpos > 0
    BEGIN

        SELECT @nextpos = charindex(",", @list, @pos + 1)
        SELECT @valuelen = CASE WHEN @nextpos > 0
                                THEN @nextpos
                                ELSE len(@list) + 1
                            END - @pos - 1

        SELECT @documentkey = substring(@list, @pos + 1, @valuelen)
        IF len(@documentkey) > 44
        BEGIN
            SELECT @protocol = convert(bigint, substring(@documentkey,45,len(@documentkey)))
            SELECT @documentkey = substring(@documentkey, 1,44)
        END
        ELSE
            SELECT @protocol = 0
        SELECT @documenttype = substring(@documentkey,21,2)
        IF @documenttype = "55"
            SELECT @documentkey = "NFe" + @documentkey
        ELSE
            SELECT @documentkey = "CTe" + @documentkey
            SELECT @dockey01 = CONVERT(BIGINT, SUBSTRING(@documentkey, 4,18))
            SELECT @dockey02 = CONVERT(BIGINT, SUBSTRING(@documentkey, 22,18))
            SELECT @dockey03 = CONVERT(INT, SUBSTRING(@documentkey, 40,9))
            INSERT @tbl (dockey, docprotocol, dckey01, dckey02, dckey03) VALUES (@documentkey, @protocol, @dockey01,
@dockey02, @dockey03)
            SELECT @pos = @nextpos
        END
    END
    RETURN
END
GO
/***** Object: UserDefinedFunction [dbo].[CreateComposeValues]  Script Date: 02/28/2011 14:50:06 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[CreateComposeValues]') AND type in
(N'FN', N'IF', N'TF', N'FS', N'FT'))
BEGIN
    execute dbo.sp_executesql @statement = N'-- =====
-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE FUNCTION [dbo].[CreateComposeValues] (@list nvarchar(MAX))
    RETURNS @tbl TABLE (doccnpj bigint NOT NULL, docnumber int NOT NULL, docserie smallint NOT NULL) AS
BEGIN
    DECLARE @pos      int,
            @nextpos  int,
            @valuelen int,
            @document nvarchar(MAX),
            @vcnpj     bigint,
            @vnumber   int,
            @vserie    smallint

    SELECT @pos = 0, @nextpos = 1

    WHILE @nextpos > 0
    BEGIN

        SELECT @nextpos = charindex(",", @list, @pos + 1)
        SELECT @valuelen = CASE WHEN @nextpos > 0

```

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

```

        THEN @nextpos
        ELSE len(@list) + 1
    END - @pos - 1

    SELECT @document = substring(@list, @pos + 1, @valuelen)
    SELECT @vcnpj = convert(bigint, substring(@document, 1, 14))
    SELECT @vnumber = convert(int, substring(@document, 15, 9))
    SELECT @vserie = convert(smallint, substring(@document, 24, 3))

    INSERT @tbl (doccnpj, docnumber, docserie) VALUES (@vcnpj, @vnumber, @vserie)

    SELECT @pos = @nextpos
END
RETURN
END
,
END

```

2.2 Criação das Tabelas em Oracle 10g

O *script* a seguir deve ser executado junto à base de dados do e-Connector, caso a mesma utilize o SGBD Oracle.

```

CREATE TABLE TBSTATISTICSCONSULT (
    STATISTICSID number(20,0) NOT NULL,
    CNPJ number(20,0) NOT NULL,
    DOCTYPE number(3,0) NOT NULL,
    AMBIENTTYPE number(5,0) NOT NULL,
    PROCSTATUS number(1,0) NOT NULL,
    DOCUMENTKEY1 number(20,0) NOT NULL,
    DOCUMENTKEY2 number(20,0) NOT NULL,
    DOCUMENTKEY3 number(10,0) NOT NULL,
    CONSULTDATE TIMESTAMP(6) NOT NULL
)
/
CREATE TABLE TBCaptureAgent (
    ID number(20,0) NOT NULL,
    SEARCHID number(20,0) NOT NULL,
    TOKENID decimal(24) NOT NULL,
    INSERTDATE TIMESTAMP(6) NOT NULL,
    XMLDOCUMENT CLOB
)
/
CREATE TABLE TBCaptureInstance (
    ID number(5,0) NOT NULL,
    SERVERID number(5,0),
    LASTUPDATE TIMESTAMP(6)
)
/
CREATE TABLE TBCaptureReport (
    ID number(20,0) NOT NULL,
    IDUSER number(10,0) NOT NULL,
    INSERTDATE TIMESTAMP(6) NOT NULL,
    FINISHDATE TIMESTAMP(6),
    XMLREPORTPROCESS CLOB,
    XMLREPORTDATAAREPLY CLOB,
    FILTERXML CLOB NOT NULL
)
/
CREATE TABLE TBCaptureReportWork (
    ID number(20,0) NOT NULL,
    REPORTID number(20,0) NOT NULL,
    SERVERID number(5,0)
)
/
CREATE TABLE TBCaptureSearch (
    ID number(20,0) NOT NULL,
    IDUSER number(10,0) NOT NULL,
    INSERTDATE TIMESTAMP(6) NOT NULL,
    FINISHDATE TIMESTAMP(6),
    DOCUMENTKEY1 number(18,0) NOT NULL,
    DOCUMENTKEY2 number(18,0) NOT NULL,
    DOCUMENTKEY3 number(8,0) NOT NULL,
    XMLREQUESTREPLY CLOB,
    TOKENID decimal(24) NOT NULL,
    SEARCHSTATUS number(3,0) NOT NULL,
    OPENSTATUS number(1,0) NOT NULL
)

```

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

```

/
CREATE TABLE TBCaptureSearchResult (
    ID number(20,0) NOT NULL,
    SEARCHID number(20,0) NOT NULL,
    EMIT_CNPJ varchar2(20) NOT NULL,
    EMIT_NAME varchar2(60),
    EMIT_PHONE varchar2(20),
    SearchedInSefaz number(1,0),
    SearchedInCold number(1,0),
    SearchedInRobot number(1,0),
    ExistsDocInCold number(1,0)
)
/
CREATE TABLE TBCaptureSearchWork (
    ID number(20,0) NOT NULL,
    SEARCHID number(20,0) NOT NULL,
    PROFILEXML CLOB NOT NULL,
    SERVERID number(5,0)
)
/
CREATE TABLE TBCaptureToken (
    ID number(20,0) NOT NULL,
    TOKENID decimal(24) NOT NULL,
    TOKEN varchar2(100) NOT NULL,
    LICENSEKEY CLOB
)
/
CREATE TABLE TBCaptureUser (
    ID number(10,0) NOT NULL,
    NAME varchar2(50) NOT NULL,
    PASSWORD varchar2(100) NOT NULL,
    PROFILETYPE number(1,0) NOT NULL,
    PROFILEXML CLOB,
    TOKENID decimal(24) NOT NULL,
    RESETPASSWORD number(1,0)
)
/
CREATE TABLE TBCaptureView (
    ID number(20,0) NOT NULL,
    IDSEARCH number(20,0) NOT NULL,
    XMLDOCUMENT CLOB
)
/
-----
-- File created - Quinta-feira-Janeiro-27-2011
-----
-- DDL for Type NDD_COMPOSEVALUEOBJECT
-----

CREATE OR REPLACE TYPE "NDD_COMPOSEVALUEOBJECT" AS OBJECT
(
    doccnpj NUMBER(14,0), docnumber NUMBER(10,0), docserie NUMBER(3,0)
);
/
-----
-- DDL for Type NDD_COMPOSEVALUEOBJECT_TABLE
-----
CREATE OR REPLACE TYPE "NDD_COMPOSEVALUEOBJECT_TABLE" AS TABLE OF NDD_COMPOSEVALUEOBJECT;
/
-----
-- DDL for Type NDD_KEYTABLEOBJECT
-----
CREATE OR REPLACE TYPE "NDD_KEYTABLEOBJECT" AS OBJECT
(
    dockey varchar(50),
    docprotocol NUMBER(19,0),
    dckey01 NUMBER(19,0),
    dckey02 NUMBER(19,0),
    dckey03 NUMBER(19,0)
);
/
-----
-- DDL for Type NDD_KEYTABLEOBJECT_TABLE
-----
CREATE OR REPLACE TYPE "NDD_KEYTABLEOBJECT_TABLE" AS TABLE OF NDD_KEYTABLEOBJECT;
/
-----
-- DDL for Type NDD_NUMBEROBJECT
-----
CREATE OR REPLACE TYPE "NDD_NUMBEROBJECT" AS OBJECT
(VNUMBER NUMBER(19,0));

```

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

```

/
-----
-- DDL for Type NDD_NUMBEROBJECT_TABLE
-----
CREATE OR REPLACE TYPE "NDD_NUMBEROBJECT_TABLE" AS TABLE OF ndd_numberobject;
/
ALTER TABLE TBCaptureAgent ADD CONSTRAINT PK_TBCaptureAgent
PRIMARY KEY (ID)
/
ALTER TABLE TBCaptureInstance ADD CONSTRAINT PK_TBCaptureInstance
PRIMARY KEY (ID)
/
alter table TBCaptureInstance add constraint PK_SERVERID_UNIQUE unique("SERVERID")
/
ALTER TABLE TBCaptureReport ADD CONSTRAINT PK_TBCaptureReport
PRIMARY KEY (ID)
/
ALTER TABLE TBCaptureReportWork ADD CONSTRAINT PK_TBCaptureReportWork
PRIMARY KEY (ID)
/
ALTER TABLE TBCaptureSearch ADD CONSTRAINT PK_TBCaptureSearch
PRIMARY KEY (ID)
/
ALTER TABLE TBCaptureSearchResult ADD CONSTRAINT PK_TBCaptureSearchResult
PRIMARY KEY (ID)
/
ALTER TABLE TBCaptureSearchWork ADD CONSTRAINT PK_TBCaptureSearchWork
PRIMARY KEY (ID)
/
ALTER TABLE TBCaptureToken ADD CONSTRAINT PK_TBCaptureToken
PRIMARY KEY (ID)
/
alter table TBCaptureToken add constraint PK_TOKENID_UNIQUE unique("TOKENID")
/
ALTER TABLE TBCaptureUser ADD CONSTRAINT PK_TBCaptureUser
PRIMARY KEY (ID)
/
ALTER TABLE TBCaptureView ADD CONSTRAINT PK_TBCaptureView
PRIMARY KEY (ID)
/
ALTER TABLE TBSTATISTICSCONSULT ADD CONSTRAINT PK_TBSTATISTICSCONSULT
PRIMARY KEY (STATISTICSID)
/
CREATE INDEX IX_INSERTDATEAGENT ON TBCaptureAgent
(INSERTDATE ASC)
/
CREATE INDEX IX_SEARCHAGENT ON TBCaptureAgent
(SEARCHID ASC)
/
CREATE INDEX IX_TOKENAGENT ON TBCaptureAgent
(TOKENID ASC)
/
CREATE INDEX IX_USERREPORT ON TBCaptureReport
(IDUSER ASC)
/
CREATE INDEX IX_REPORTIDREPORTWORK ON TBCaptureReportWork
(REPORTID ASC)
/
CREATE INDEX IX_SERVERIDREPORTWORK ON TBCaptureReportWork
(SERVERID ASC)
/
CREATE INDEX IX_DOCUMENTKEYSEARCH ON TBCaptureSearch
(DOCUMENTKEY1 ASC, DOCUMENTKEY2 ASC, DOCUMENTKEY3 ASC)
/
CREATE INDEX IX_TOKENSERACH ON TBCaptureSearch
(TOKENID ASC)
/
CREATE INDEX IX_SEARCHIDSEARCHRESULT ON TBCaptureSearchResult
(SEARCHID ASC)
/
CREATE INDEX IX_SEARCHIDSEARCHWORK ON TBCaptureSearchWork
(SEARCHID ASC)
/
CREATE INDEX IX_SERVERIDSEARCHWORK ON TBCaptureSearchWork
(SERVERID ASC)
/
CREATE INDEX IX_TOKEN ON TBCaptureUser
(TOKENID ASC)
/
CREATE INDEX IX_SEARCHVIEW ON TBCaptureView
(IDSEARCH ASC)
/

```


Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

```

v_pos := 0;
v_nextpos := 1;
v_documentkey := '';
v_documenttype := '';
v_protocol := 0;
v_dockey01 := 0;
v_dockey02 := 0;
v_dockey03 := 0;
WHILE v_nextpos > 0
LOOP
  v_nextpos := INSTR(plist, ',', v_pos + 1);
  v_valuelen := (CASE WHEN v_nextpos > 0 THEN v_nextpos
    ELSE LENGTH(plist) + 1
    END) - v_pos - 1;
  v_documentkey := SUBSTR(plist, v_pos + 1, v_valuelen);
  if length(v_documentkey) > 44 then
    v_protocol := to_number(substr(v_documentkey, 45, length(v_documentkey)));
    v_documentkey := substr(v_documentkey, 1, 44);
  else
    v_protocol := 0;
  end if;
  v_documenttype := substr(v_documentkey, 21, 2);
  if v_documenttype = '55' then
    v_documentkey := 'NFe' || v_documentkey;
  else
    v_documentkey := 'CTe' || v_documentkey;
  end if;
  v_dockey01 := TO_NUMBER(SUBSTR(v_documentkey, 4, 18));
  v_dockey02 := TO_NUMBER(SUBSTR(v_documentkey, 22, 18));
  v_dockey03 := TO_NUMBER(SUBSTR(v_documentkey, 40, 8));
  PIPE ROW(ndd_keytableobject(substr(v_documentkey,1,length(v_documentkey)), v_protocol, v_dockey01,
v_dockey02, v_dockey03));
  V_pos := V_nextpos;
END LOOP
RETURN;
END;

/

-----
-- DDL for Function CREATELISTTABLE
-----

CREATE OR REPLACE FUNCTION "CREATELISTTABLE" (plist CLOB)
RETURN ndd_numberobject_table PIPELINED AS

  v_pos number(19,0);
  v_nextpos number(19,0);
  v_valuelen number(19,0);
  v_numberval number(19,0);
BEGIN
  v_pos := 0;
  v_nextpos := 1;
  v_numberval := 0;
  WHILE v_nextpos > 0
  LOOP
    v_nextpos := INSTR(plist, ',', v_pos + 1);
    v_valuelen := (CASE WHEN v_nextpos > 0 THEN v_nextpos
      ELSE LENGTH(plist) + 1
      END) - v_pos - 1;
    v_numberval := TO_NUMBER(SUBSTR(plist, v_pos + 1, v_valuelen));
    PIPE ROW(ndd_numberobject(v_numberval));
    V_pos := V_nextpos;
  END LOOP
  RETURN;
END;

/

```

3. Configurações

Os serviços de processamento, bem como, de monitoramento (Guardian), possuem arquivos de configurações com valores que definem a execução dos mesmos. Os valores padrões de configuração dos serviços, atribuídos automaticamente durante a instalação, são baseados nos requisitos mínimos de *hardware* especificados no manual de instalação do produto. Porém, é possível alterar estas definições buscando melhorias de desempenho da solução, neste caso

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

sugerimos que o ambiente onde a solução encontra-se disponível seja analisado, a fim de avaliar qual a melhor configuração a ser aplicada.

Para editar as configurações dos serviços utilizados pela solução, siga os passos detalhados nas próximas seções.

3.1 Editando as Configurações do Serviço de Processamento

O serviço de processamento é responsável por capturar na base de dados os registros de consultas e relatórios a serem realizados, processá-los e retornar o resultado ao usuário solicitante.

Algumas configurações determinam o comportamento do serviço de processamento, podendo estas, serem modificadas. Para isso, acesse o diretório de instalação do produto, geralmente disponível em "**<raiz>\Arquivos de programas\NDDigital**", e localize o subdiretório "**Processing**". Encontre o arquivo de configuração, este obrigatoriamente deve ter o mesmo nome do serviço, normalmente relacionado como "**NDDigital.XMLCapture.Processing.Service.exe.config**". Este arquivo conterá as seguintes informações:

- **LogPath:** Caminho onde serão gerados os *logs* de processamento da solução;
- **IDService:** Identificador do serviço de processamento. O produto permite que mais de um serviço de processamento seja executado sobre a mesma base de dados, mediante a necessidade e o volume de dados a serem tratados. O instalador do produto registra apenas um único serviço sendo que, quando necessário, os demais devem ser registrados manualmente. Neste caso, cada serviço de processamento deve possuir um ID único que o identifique;
- **ItemQuantity:** Quantidade de *threads* a serem executadas simultaneamente, sendo possível configurar até 15 *threads* por serviço;
- **ServiceType:** Define o tipo de serviço. O serviço de processamento pode ser configurado para processar:
 - **1:** apenas pesquisas;
 - **2:** apenas relatórios;
 - **0:** pesquisas e relatórios.

Considerando isso pode-se, por exemplo, possuir um serviço para processar apenas as pesquisas efetuadas e outro apenas para processar os relatórios gerados, ou ainda, um único serviço que processe tanto pesquisas quanto relatórios. Isto vai depender do volume de processamento e das necessidades do utilizador;

- **WaitDataBase:** Determina o intervalo de tempo, em segundos, que o sistema vai aguardar para tentar uma nova conexão com a base de dados, caso a mesma encontre-se indisponível;
- **WaitNothing:** Intervalo de tempo, em segundos, que o serviço de processamento irá aguardar para verificar se existem consultas ou relatórios a serem processados;
- **UseDocSearchService:** Determina se o serviço de processamento do DocSearch será utilizado.

Depois de alteradas as configurações do serviço de processamento, reinicie o mesmo.

3.2 Editando as Configurações do Serviço de Monitoramento (Guardian)

O serviço de monitoramento, Guardian, é responsável por garantir que as consultas e relatórios estejam sendo direcionados a serviços de processamento ativos.

Assim como o processamento, o serviço de monitoramento também dispõe de configurações editáveis. Para alterar as configurações do Guardian, acesse o diretório de instalação do produto, geralmente disponível em "**<raiz>\Arquivos de programas\NDDigital**", e localize o subdiretório "**Guardian**". Encontre o arquivo de configuração do serviço de monitoramento, este obrigatoriamente deve ter o mesmo nome do serviço, normalmente relacionado como "**NDDigital.XMLCapture.Guardian.Service.exe.config**". Não pode existir mais de um serviço guardião para uma mesma base de dados do XML Capture. As definições da execução do serviço podem ser editadas, considerando:

- **LogPath:** Caminho onde serão gerados os *logs* do serviço de monitoramento;
- **NothingUpdateTime:** Intervalo de tempo, em segundos, que o serviço de monitoramento irá aguardar para indicar que o serviço de processamento está parado ou deixou de funcionar;

Código do Documento MDS024	Responsável Analista de Testes	Data Última Revisão 24/08/2011	Versão Última Revisão 2
-------------------------------	-----------------------------------	-----------------------------------	----------------------------

- **WaitDBTime:** : Determina o intervalo de tempo, em segundos, que o sistema vai aguardar para tentar uma nova conexão com a base de dados, caso a mesma encontre-se indisponível;
- **SleepTime:** Intervalo de tempo, em segundos, que o serviço irá aguardar caso não tenha nenhum registro a processar.

Depois de alteradas as configurações do serviço do Guardian, reinicie o mesmo.