



Manual de integração Orientações gerais

1	Conteúdo do documento	2
2	Diretório de Integração	2
3	Nomenclatura dos arquivos	2
4	O que o arquivo deve conter	3
5	Obtenção das informações no arquivo de integração	3
6	Tipos e formatação para os arquivos no formato texto (txt)	4
	Codificação do arquivo	4
	Separador de campos	4
	Final de registro	4
	Alfanumérico	4
	Numérico	4
	Data	5
	Período	5
	Horário	5
7	Tipos e formatação para os arquivos no formato xml	5
8	Validação e retorno de inconsistências	5
9	Arquivos criptografados	5
	Exemplo de algoritmo para criptografia de arquivo texto	6
	Exemplo de algoritmo para criptografia de arquivo xml	7
10	Configurando a origem de dados externos	8

1 Conteúdo do documento

Este documento contém informações para auxílio na integração entre o módulo eSocial SaaS e os dados complementares ao sistema de folha de pagamento, assim como os layouts de integração para a uso no módulo eSocial.

2 Diretório de Integração

O módulo eSocial permite a configuração da forma de integração dos conceitos complementares ao sistema de folha de pagamento através da funcionalidade Dados Externos.

Se o operador do módulo eSocial optar por fazer a integração de um conceito complementar por meio de arquivos, ele deverá informar ao sistema qual o diretório onde os arquivos de integração do referido conceito estarão disponibilizados.

O diretório deverá conter apenas o arquivo a ser importado. O arquivo deve ser desenvolvido de acordo com o layout apresentado no sistema, inclusive sua nomenclatura.

3 Nomenclatura dos arquivos

O nome do arquivo será composto pelo prefixo do conceito contido no arquivo, pela inscrição do empregador, pela data, hora, minuto e segundo da geração do arquivo.

O formato padrão para a data que adotamos foi AAAAMMDD (onde: AAAA = ano; MM = mês; DD = dia) e para a hora, minuto e segundo foi HHMMSS (onde: HH = hora; MM = minutos; SS = segundos). Esse formato possibilita que os arquivos possam ser separados e ordenados pelo próprio sistema operacional, facilitando sua busca, visualização e manipulação. O domínio de horas adotado para um dia vai de 00:00 a 23:59.

Após o nome padrão do arquivo, pode ser informado um texto livre de até 25 caracteres podendo conter espaços e caracteres especiais, desde que esses sejam suportados pelo sistema operacional. Esse atributo não é obrigatório. Em casos de mais de um fornecedor de um mesmo serviço, recomendamos que esse campo seja utilizado para a identificação do fornecedor.

Para a importação de arquivos **XML**, os conceitos relacionados a “**Obrigações Trabalhistas**” e “**Folha de Pagamentos**”, na **exclusão**, deve ser informado o prefixo “EXCLUSAO_” antes do nome padrão.

Caso os arquivos não possuam a nomenclatura definida nesta regra, os arquivos não serão processados.

Exemplos

Sem identificação do fornecedor	
Prefixo do conceito:	HORARIO_
Inscrição do Empregador:	12.345.678/9012-34 (inscrição fictícia)
Data da geração:	01/02/2014
Horário da geração:	08:25:40 (8 horas, 25 minutos e 40 segundos)
Texto livre:	<<Não informado>>
Nome do arquivo:	HORARIO_12345678901234_20140201082540.txt

Prefixo do conceito:	ASO_
Inscrição do Empregador:	12.345.678/9012-34 (inscrição fictícia)
Data da geração:	01/02/2014
Horário da geração:	08:25:40 (8 horas, 25 minutos e 40 segundos)
Texto livre:	<<Não informado>>
Nome do arquivo:	EXCLUSAO_ASO_12345678901234_20140201082540.xml

Com identificação do fornecedor	
Prefixo do conceito:	ATIVIDADES_
Inscrição do Empregador:	12.345.678/9012-34 (inscrição fictícia)
Data da geração:	01/02/2014
Horário da geração:	08:25:40 (8 horas, 25 minutos e 40 segundos)
Texto livre:	EMPRESA FICTICIA
Nome do arquivo:	ATIVIDADES_12345678901234_20140201082540_EMPRESAFICTICIA.xml

4 O que o arquivo deve conter

Deve ser informado no arquivo de integração as informações por trabalhador que ainda não foram informadas em arquivos anteriores, ou seja, cada arquivo de integração deve conter apenas as ações realizadas entre a geração do arquivo anterior e o novo arquivo do conceito.

Caso opte pela integração via “*txt*”, todos os dados por trabalhador que devem ser incluídos ou retificados, podem ser informados em um único arquivo. Caso opte pela integração via “*xml*”, os arquivos devem ser encaminhados no layout estabelecido pela receita e contendo somente um *evento* por arquivo.

Não é permitido que um mesmo arquivo contenha mais de um registro referente a uma mesma chave de identificação do conceito tratado dentro de um mesmo grupo formado pelo registro pai. Também não é permitido, em um mesmo arquivo, registros de empregadores distintos. Cada arquivo deve conter os dados do empregador definido no nome do arquivo.

Se o período de apuração ao qual se refere o registro ainda estiver “em aberto” no sistema federal do eSocial, não é necessário sinalizar o registro como “retificador”. Os dados com mesma chave de identificação serão sempre sobrescritos enquanto o período de apuração estiver “em aberto”.

5 Obtenção das informações no arquivo de integração

A busca e carga das informações, por parte do módulo eSocial, é realizada de forma automatizada.

Um serviço de agendamento de tarefas gerencia a busca e obtenção das informações em dias e horários definidos pelo operador do módulo eSocial. Dessa forma, o sistema obterá as informações no diretório de integração, sempre, com dia e horário configurado.

Apesar dessa característica, algumas ações realizadas no módulo eSocial podem fazer com que seja necessário realizar acesso fora dos dias e horários já agendados, tais como atualização de pendências e geração manual. Isso pode ocorrer devido a fluxos de

trabalho do operador que necessitem de uma carga imediata das informações complementares presentes no arquivo.

Finalizada a importação do arquivo, o mesmo será realocado no subdiretório “Processados”.

6 Tipos e formatação para os arquivos no formato texto (txt)

Codificação do arquivo

O arquivo no formato texto, deve ser codificado em ASCII – ISO 8859-1 (Latin-1), não sendo aceitos quaisquer outras codificações de texto nem campos compactados (packed decimal), zonados, binários, ponto flutuante (float point), etc.

Separador de campos

Quando o operador informa que deseja realizar a integração de um conceito através de arquivo, ele indica também qual é o separador de campos que deseja utilizar nos arquivos do conceito em questão.

Os separadores possíveis de seleção são “,” (vírgula), “|” (barra vertical), “;” (ponto e vírgula) e “TAB” (tabulação).

O operador deve estar ciente de que, uma vez selecionado o separador, o mesmo não pode ser utilizado no conteúdo dos campos, pois isso gerará inconsistência na estrutura do arquivo.

Final de registro

Todo registro deve ser finalizado com uma quebra de linha. A quebra de linha deve ser feita com o uso dos caracteres “CR” (Carriage Return – retorno de carro, caractere 13 da tabela ASCII) e “LF” (Line Feed – salto de linha, caractere 10 da tabela ASCII).

Alfanumérico

Podem ser informados letras, números e caracteres especiais. O tamanho máximo permitido para cada campo é informado no layout.

Não é necessário informar espaços, ou qualquer outro caractere, para que o conteúdo do campo atinja o tamanho máximo definido para o mesmo.

Não existe limite mínimo para o campo, ao menos que este seja obrigatório. Um campo alfanumérico obrigatório deve possuir, no mínimo, um caractere diferente de espaço. Um campo alfanumérico não obrigatório pode ser informado vazio.

Númérico

Podem ser informados apenas números, sem espaços ou caracteres especiais. Quando o campo se referir a um conceito com valor decimal (valores monetários, por exemplo) será indicada a quantidade de caracteres que devem ser utilizados para representar os valores decimais.

Deve ser usado o caractere “.” (ponto) para separar a parte decimal da parte inteira. Não deve ser usado nenhum separador para a parte inteira. Não é necessário informar zeros a esquerda para formar o tamanho máximo definido para o campo.

Por exemplo, quando indicado o tamanho “12,2”, deve ser informado um valor numérico com, no máximo, 12 dígitos para o valor inteiro, um “.” para separar o valor decimal e, no máximo, dois dígitos para valor decimal. Dessa forma, o valor monetário R\$ 850,89 deve ser informado “850.89”.

Caso não haja indicação de quantidade de casas decimais, o valor do campo deve ser considerado como sendo um valor inteiro e, dessa forma, não permite o uso do

separador “.”. Ex: O valor inteiro “854545” não pode ser informado como “854545.”.

Data

As datas devem ser informadas no formato AAAAMMDD (ano/mês/dia). Onde AAAA são os quatro dígitos do ano, MM os dois dígitos do mês e DD os dois dígitos do dia.

Período

Para campos que devem indicar período de referência (mês e ano). A informação deve estar no formato AAAAMM onde MM são os dois dígitos referentes ao mês e AAAA são os quatro dígitos onde deve ser informado o ano.

Horário

Os horários (horas e minutos) devem ser informados no formato HHMM, onde HH se refere a hora e MM se refere aos minutos. A informação de horas deve obedecer ao intervalo de 00:00 a 23:59.

7 Tipos e formatação para os arquivos no formato xml

Devem seguir as formatações definidas no leiaute oficial do eSocial, disponibilizado pela Receita Federal do Brasil.

8 Validação e retorno de inconsistências

Durante o carregamento dos dados contidos nos arquivos de integração, serão feitas validações para garantir que o layout definido neste documento foi corretamente seguido e se o conteúdo dos campos é válido.

Se alguma inconsistência for encontrada no arquivo de integração, seja qual for a natureza da mesma, o módulo eSocial apresentará um resumo das inconsistências em um arquivo, com extensão xml, com o mesmo nome do arquivo de origem acrescido do sufixo “_inconsistencias”, no diretório de integração discriminando a linha, o campo e a inconsistência encontrada.

9 Arquivos criptografados

Os arquivos de integração poderão estar criptografados, para isso é necessário configurar quais arquivos estarão criptografados na funcionalidade “Configurações do Módulo” na guia “Segurança”.

Para criptografia de arquivos no módulo é utilizado o algoritmo RSA, onde é gerada uma chave pública para criptografar e uma chave privada para descriptografar, logo para a criptografia de arquivos externos é necessário a chave pública equivalente à chave privada gerada e salva com o auxílio da senha informada pelo usuário na mesma funcionalidade, esta chave pública pode ser exportada e salva em um arquivo xml para que possa ser utilizada na criptografia posteriormente.

Configurações do Módulo

Senha para Criptografia da Chave Privada:

Compartilhar as Configurações de Segurança com Todas as Empresas

Criptografar Arquivos Gerados

<input type="checkbox"/> Arquivo
<input type="checkbox"/> 1000 - Informações do Empregador / Contribuinte / Órgão Público
<input type="checkbox"/> 1005 - Tabela de Estabelecimentos, Obras ou Unidades de Órgãos Públicos
<input type="checkbox"/> 1010 - Tabela de Rubricas
<input type="checkbox"/> 1020 - Tabela de Lotações Tributárias
<input type="checkbox"/> 1030 - Tabela de Cargos/Empregos Públicos - Descontinuado
<input type="checkbox"/> 1035 - Tabela de Carreiras Públicas - Descontinuado
<input type="checkbox"/> 1040 - Tabela de Funções/Cargos em Comissão - Descontinuado
<input type="checkbox"/> 1050 - Tabela de Horários/Turnos de Trabalho - Descontinuado

Arquivos de Dados Externos Criptografados

<input type="checkbox"/> Arquivo
<input type="checkbox"/> Ambiente de Trabalho
<input type="checkbox"/> Aquisição de Produção Rural
<input type="checkbox"/> Comercialização da Produção Rural Pessoa Física
<input type="checkbox"/> Comunicação de Acidente de Trabalho (CAT)
<input type="checkbox"/> Condições Ambientais do Trabalho - Agentes Nocivos
<input type="checkbox"/> Contratação de Trabalhadores Avulsos Não Portuários
<input type="checkbox"/> Exame Toxicológico do Motorista Profissional
<input type="checkbox"/> Horário e Turno de Trabalho

Exemplo de algoritmo para criptografia de arquivo texto

Para a criptografia do conteúdo de um arquivo texto, é utilizado um algoritmo de criptografia em blocos. Segue abaixo um exemplo do mesmo na linguagem de programação C#.

```

/// <summary>
/// Criptografa o texto em blocos.
/// </summary>
/// <param name="textoASerCriptografado">O texto a ser criptografado.</param>
/// <param name="caminhoDoArquivoXmlChavePublica">O caminho do arquivo xml com a chave pública.</param>
/// <returns>Texto criptografado.</returns>
public string Criptografa(string textoASerCriptografado, string caminhoDoArquivoXmlChavePublica)
{
    const int TAMANHO_MAXIMO_DA_CHAVE = 1024;
    var algoritmoRSA = new RSACryptoServiceProvider();
    var tamanhoDaChaveEmBytes = TAMANHO_MAXIMO_DA_CHAVE / 8;

    // Obtém o vetor de bytes correspondente ao texto que deve ser criptografado.
    var arquivo = Encoding.UTF32.GetBytes(textoASerCriptografado);
    var tamanhoMaximo = tamanhoDaChaveEmBytes - 42;
    var tamanhoDoArquivo = arquivo.Length;

    // Obtém a quantidade de blocos contidas no texto.
    var quantidadeDeBlocos = tamanhoDoArquivo / tamanhoMaximo;
    var stringBuilder = new StringBuilder();

    // Carrega a chave pública obtida na exportação.
    algoritmoRSA.FromXmlString(File.ReadAllText(caminhoDoArquivoXmlChavePublica));

    for (var contador = 0; contador <= quantidadeDeBlocos; contador++)
  
```

```

{
    // Obtém o bloco para criptografia.
    byte[] blocoDeBytes = new byte[(tamanhoDoArquivo - tamanhoMaximo * contador > tamanhoMaximo) ?
        tamanhoMaximo : tamanhoDoArquivo - tamanhoMaximo * contador];
    Buffer.BlockCopy(arquivo, tamanhoMaximo * contador, blocoDeBytes, 0, blocoDeBytes.Length);
    byte[] bytesCriptografados = algoritmoRSA.Encrypt(blocoDeBytes, false);

    // Inverte o vetor de bytes.
    Array.Reverse(bytesCriptografados);
    stringBuilder.Append(Convert.ToBase64String(bytesCriptografados));
}

// Retorna o texto criptografado.
return stringBuilder.ToString();
}

```

Exemplo de algoritmo para criptografia de arquivo xml

Para a criptografia do conteúdo de um arquivo xml, a TAG padrão eSocial deve ser substituída pela TAG eSocialCriptografado. Segue abaixo um exemplo do algoritmo utilizado na linguagem de programação C#.

```

/// <summary>
/// Criptografa o documento XML.
/// </summary>
/// <param name="documentoXml">O documento XML.</param>
/// <param name="nomeDaChave">O nome da chave utilizada.</param>
/// <param name="caminhoDoArquivoXmlChavePublica">O caminho do arquivo XML com a chave pública.</param>
public void Criptografa(XmlDocument documentoXml, string nomeDaChave, string caminhoDoArquivoXmlChavePublica)
{
    const string ELEMENTO_A_SER_CRIPTOGRAFADO = "eSocial";
    const string ELEMENTO_GERADO_NA_CRIPTOGRAFIA = "eSocialCriptografado";
    RSACryptoServiceProvider algoritmoRSA = null;
    RijndaelManaged chaveDaSessao = null;

    // Cria uma instância do elemento especificado.
    XmlElement elementoParaCriptografia =
        documentoXml.GetElementsByTagName(ELEMENTO_A_SER_CRIPTOGRAFADO)[0] as XmlElement;

    // Caso o elemento não exista retorna exceção.
    If(elementoParaCriptografia == null)
    {
        throw new XmlException("Elemento não encontrado.");
    }

    try
    {
        algoritmoRSA = new RSACryptoServiceProvider();

        // Carrega a chave pública obtida na exportação.
        algoritmoRSA.FromXmlString(File.ReadAllText(caminhoDoArquivoXmlChavePublica));

        // Cria uma chave Rijndael de 256bit.
        chaveDaSessao = new RijndaelManaged();
        chaveDaSessao.KeySize = 256;

        // Cria uma nova instância de EncryptedXml e a usa para criptografar
        // o elemento XML com uma nova chave randômica simétrica.
        EncryptedXml eXml = new EncryptedXml();

        byte[] elementoCriptografado = eXml.EncryptData(elementoParaCriptografia, chaveDaSessao, false);

        // Cria um objeto EncryptedData e o popula com a informação criptografa.
        EncryptedData dadosCriptografados = new EncryptedData();
        dadosCriptografados.Type = EncryptedXml.XmlEncElementUrl;
        dadosCriptografados.Id = ELEMENTO_GERADO_NA_CRIPTOGRAFIA;

        // Cria um elemento EncryptionMethod para receber qual algoritmo foi usado para criptografar.
        dadosCriptografados.EncryptionMethod = new EncryptionMethod(EncryptedXml.XmlEncAES256Url);

        // Criptografa a chave de sessão e o adiciona no elemento EncryptedKey.
        EncryptedKey chaveCriptografada = new EncryptedKey();
    }
}

```

```

byte[] chaveCriptografadaEmBytes = EncryptedXml.EncryptKey(chaveDaSessao.Key, algoritmoRSA, false);
chaveCriptografada.CipherData = new CipherData(chaveCriptografadaEmBytes);
chaveCriptografada.EncryptionMethod = new EncryptionMethod(EncryptedXml.XmlEncRSA15Url);

// Cria uma nova referência do elemento.
// Esse elemento é opcional e indica qual elemento criptografado que utiliza essa chave
// Um documento XML pode ter vários elementos criptografados utilizando várias chaves.
DataReferencereferenciaDoElemento=newDataReference();

// Especifica a tag do elemento criptografado.
referenciaDoElemento.Uri="#" + ELEMENTO_GERADO_NA_CRIPTOGRAFIA;

// Adiciona a referência do elemento na chave criptografada.
chaveCriptografada.AddReference(referenciaDoElemento);

//Adiciona a chave criptografada nos dados criptografados.
dadosCriptografados.KeyInfo.AddClause(newKeyInfoEncryptedKey(chaveCriptografada));

// Especifica as informações da chave como nome da chave RSA.
// Cria uma nova informação de chave.
KeyInfoName informacoesDaChave = new KeyInfoName();

// Especifica o nome da chave.
informacoesDaChave.Value = nomeDaChave;

// Adiciona as informações da chave no elemento da chave.
chaveCriptografada.KeyInfo.AddClause(informacoesDaChave);

// Adiciona o elemento criptografado.
dadosCriptografados.CipherData.CipherValue = elementoCriptografado;

// Substitui o elemento do documento original pelo elemento criptografado.
EncryptedXml.ReplaceElement(elementoParaCriptografia, dadosCriptografados, false);
}
finally
{
    if(chaveDaSessao != null)
    {
        chaveDaSessao.Clear();
    }

    if(algoritmoRSA!=null)
    {
        algoritmoRSA.Clear();
    }
}
}
}

```

10 Configurando a origem de dados externos

Execute os passos a seguir para realizar a correta configuração da origem de dados externos:

1. Na funcionalidade “Configuração da Origem de Dados”, indicar o diretório onde estarão os arquivos para integração. Este diretório deve ser compartilhado para que todos os demais operadores possam executar as tarefas considerando os arquivos armazenados.

Configuração da Origem dos Dados Sincronizador

Utilizar diretório padrão para importação de arquivos

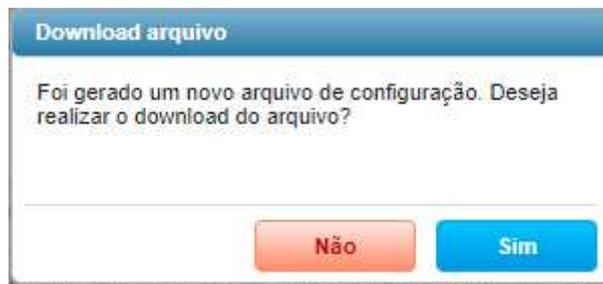
Tabela de Processos Administrativos/Judiciais

Tipo / Separador: Diretório:

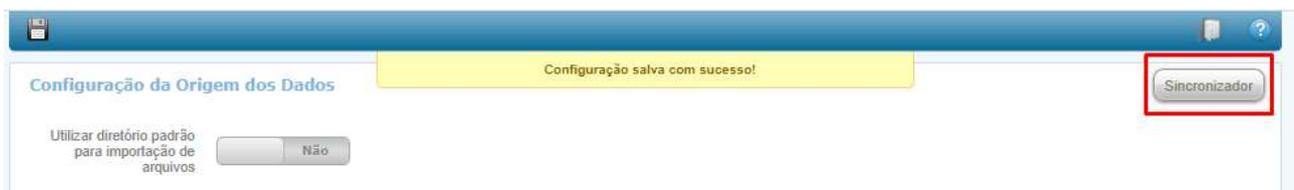
Rubricas ?

Tipo / Separador: Diretório:

2. Ao salvar a configuração, será apresentado a opção para realizar o download do arquivo Config.xml. O arquivo deve ser baixado para viabilizar a integração.



3. Através do botão Sincronizador, realizar o download do arquivo, descompactá-lo e mover o arquivo Config.xml para esta pasta.



Nome	Data de modificação	Tipo	Tamanho
Config	14/07/2021 12:01	Documento XML	2 KB
LGeSocialService	14/07/2021 12:01	Aplicativo	288 KB
MetroFramework.dll	14/07/2021 12:01	Extensão de aplica...	345 KB
MetroFramework.Fonts.dll	14/07/2021 12:01	Extensão de aplica...	657 KB
Microsoft.WindowsAzure.Storage.dll	14/07/2021 12:01	Extensão de aplica...	1.096 KB
Newtonsoft.Json.dll	14/07/2021 12:01	Extensão de aplica...	493 KB

4. Executar o arquivo LGeSocialService, marcar a checkbox "Gerar Log" e clique em "Monitorar arquivos externos". O arquivo de log gerado, apresentará o andamento do processo de download.



5. Após finalizada a sincronização dos arquivos, acesse a tarefa que processará os eventos contendo os dados inseridos nos arquivos e faça a sua execução. O acompanhamento poderá ser realizado através da funcionalidade “Gerenciador de tarefas” do eSocial.
6. Caso o processamento resulte em possíveis inconsistências, será possível consultá-las no mesmo diretório indicado no passo 1, através da pasta PROCESSADOS.

