

# Empacotando a extensão

Terminada a [montagem do ambiente](#) é possível organizar e preparar sua extensão para distribuição.

Após a preparação inicial do seu projeto, você ainda poderá modificar os arquivos gerados conforme sua necessidade. Eles fazem parte do seu projeto agora e não serão atualizados em futuras versões.

## Organização dos arquivos e pastas

A tabela abaixo relaciona o que você pode ter nas pastas do seu projeto.

Elemento	Pasta	Descrição	Destino	Local de entrega
versao.ini	.	Arquivo que informa a versão da extensão (major version, minor version e release). Além desses, o build_number é passado como parâmetro no empacotamento. <ul style="list-style-type: none"><li>Se a extensão é um plugin Python (.COP), o comando py_gerar_cop se encarrega de gerar o plugin com a versão definida no versao.ini.</li><li>Para outros tipos de extensão Python, é possível usar o comando py_alterar_versao para gerar os arquivos de versão utilizáveis pelo Python.</li><li>Para todos os outros tipos de plugins, executáveis e DLLs é desejável que o desenvolvedor utilize essa versão, pois isso facilita o suporte.</li></ul>	Empacotador	Usado somente durante o empacotamento
arquivo .COP	client	Um arquivo com esta extensão é gerado durante o empacotamento de um plugin Python.	Terminais	<ncr solution>\plugins\<nome_do_plugin>
arquivo .COL	client	Um plugin DLL 'win32' deve ser fornecido com esta extensão para as estações. A compilação e cópia do arquivo na pasta são de responsabilidade do fornecedor.	Terminais	<ncr solution>\plugins\<nome_do_plugin>
arquivo Plugin.*.DLL	client	Um plugin DLL 'assembly', do dotnet, deve ser fornecido com este prefixo e extensão para as estações. A compilação e cópia do arquivo na pasta são de responsabilidade do fornecedor.	Terminais	<ncr solution>\plugins\<nome_do_plugin>
ui.config	client	Arquivo que especifica adições à interface do NCR Colibri nos plugins.	Terminais	<ncr solution>\plugins\<nome_do_plugin>
pasta <b>Server</b>	server	O empacotador adiciona a estrutura completa desta pasta ao destino no servidor	Servidor	<ncr solution>\master\colibri\<nome_da_extensao>
pasta <b>Extras</b>	extras	O empacotador adiciona a estrutura completa desta pasta ao destino nas estações.	Terminais	<ncr solution>\plugins\<nome_do_plugin>\Extras
pasta <b>Reports</b>	reports	O empacotador adiciona a estrutura completa desta pasta ao destino nas estações. Saiba mais em <a href="#">Reports</a> .	Terminais	<ncr solution>\plugins\<nome_do_plugin>\Reports
pasta <b>Templates</b>	templates	O empacotador adiciona a estrutura completa desta pasta ao destino nas estações. Saiba mais em <a href="#">Personalizar templates</a> .	Terminais	<ncr solution>\plugins\<nome_do_plugin>\Reports
pasta <b>_scripts</b>	_scripts	Estrutura de pastas e comandos SQL que gera o arquivo _scripts.zip. Saiba mais em <a href="#">Acesso ao SQL</a> .	Master	Executados contra o banco de dados do produto pos, cbo ou master
pasta <b>_build</b>	_build	Pasta auxiliar que pode ser utilizada com os arquivos usados no empacotamento	Empacotador	Usado somente durante o empacotamento

## Entendendo o manifesto

O *manifesto.dat* é um arquivo JSON incluído na extensão e que lista os arquivos que serão na sua instalação.

O arquivo *manifesto.dat* é gerado a partir do arquivo *manifesto.server* durante o empacotamento. o SDK cria o arquivo *manifesto.server* pré-configurado, mas você pode alterá-lo conforme as suas necessidades.

A tabela abaixo lista as chaves do *manifesto.server* e seus significados.

Chave	Significado
-------	-------------

nome	Nome único da extensão
nome_exibicao	Nome de exibição da extensão
produto	<p>Produto ao qual se aplica:</p> <ul style="list-style-type: none"> <li>• pos - Extensão somente para o POS</li> <li>• cbo - Extensão somente para o Back Office</li> <li>• master - Extensão para POS/Back Office</li> <li>• universal - Não define produto</li> </ul> <div style="background-color: #fff9c4; padding: 10px; margin-top: 10px;"> <p><b>Importante!</b></p> <p>A base de dados padrão onde são aplicados os scripts do pacote varia conforme o produto definido nesta chave.</p> </div>
empresa	Nome da empresa fornecedora
sigla_empresa	Sigla da empresa, com até 10 caracteres alfanuméricos
nivel_sdk	Obrigatória, especifica o nível do SDK, que atualmente é 1
arquivos	Lista de arquivos da extensão.
versoes_bases	<p>Lista de schemas e versões de banco compatíveis com a extensão. Qualquer versão de schema de banco de dados superior o informado no manifesto bloqueia a instalação da extensão pelo Master.</p> <pre>{ "schema": "dbo", "versao": "01.02.03.04" }</pre>
portas	Lista de portas da extensão a serem gerenciadas pelo Master
comandos	Lista de comandos da extensão a serem gerenciados pelo Master. Os comandos poderão ser executados via api, durante a instalação da extensão e ter tipos de agendamentos associados
tipos_agendamento	Lista de tipos de agendamentos associados a comandos da extensão
agendamentos	Lista de agendamentos da extensão
executar	Lista de execuções de comandos

## Gerando um arquivo COP (somente Python)

Para distribuir os plugins em python é necessário gerar o arquivo COP de cada plugin na pasta correta de distribuição (client ou server).

Para isso, **abra um prompt de comando na pasta do SDK** (cmd) e execute o comando

```
inv py-gerar-cop nome-extensao --versao=1.2.3 --build-number=4 --develop=true
```

onde:

<b>nome-extensao</b>	Nome da extensão, que indica a subpasta onde está localizada a extensão que você quer empacotar.
<b>versao</b>	Versao do plugin. Caso não fornecida, será obtida do plugins python ou do arquivo versao.ini
<b>build-number</b>	Número do build que é adicionado à versão da extensão, juntamente com as outras informações de versão obtidas do arquivo <i>versao.ini</i> .
<b>develop</b>	Indica que é um pacote de desenvolvimento (esse é o padrão). Para versão final esse valor deve ser <b>false</b> .

O procedimento gera um arquivo `.cop` na subpasta cliente.

---

## Gerando um pacote da extensão

Para empacotar sua extensão, **abra um prompt de comando na pasta do SDK** (cmd) e execute o comando

```
inv empacotar nome-extensao --build-number=5 --develop=true
```

onde:

<b>nome-extensao</b>	Nome da extensão, que indica a subpasta onde está localizada a extensão que você quer empacotar.
<b>build-number</b>	Número do build que é adicionado à versão da extensão, juntamente com as outras informações de versão obtidas do arquivo <i>verso.ini</i> .
<b>develop</b>	Indica que é um pacote de desenvolvimento (esse é o padrão). Para versão final esse valor deve ser <b>false</b> .

O empacotamento gera um arquivo de extensão `.cmpkg` na subpasta `_build\temp`. Este arquivo pode ser importado no Master como um pacote de distribuição.