



# **Trends.Earth - Guia do desenvolvedor**

***Versão 2.1.8***

**Conservation International**

**06 fev., 2023**

<b>1</b>	<b>Desenvolvimento</b>	<b>2</b>
1.1	Modificando o código QGIS Plugin	2
1.2	Modificando o código de processamento do Earth Engine	6
1.3	A editar modelos de camada vetorial	8
1.4	Gestão dos metadados do conjunto de dados	9
1.5	A atualizar estrutura de relatório	10
1.6	Contribuindo para a documentação	13
<b>2</b>	<b>Changelog</b>	<b>18</b>
2.1	2.1.8 (January 25, 2023)	18
2.2	2.1.6 (December 8, 2022)	18
2.3	2.1.6 (November 21, 2022)	18
2.4	2.1.2 (November 18, 2022)	19
2.5	2.1.0 (November 11, 2022)	19
2.6	2.0.7 (October 31, 2022)	19
2.7	2.0.5 (October 19, 2022)	19
2.8	2.0.3 (October 19, 2022)	19
2.9	2.0.1 (October 13, 2022)	19
2.10	2.0 (July 20, 2022)	20
2.11	1.0.10 (7 de Julho, 2022)	20
2.12	1.0.8 (15 de Outubro, 2021)	20
2.13	1.0.6 (15 de julho de 2021)	20
2.14	1.0.4 (30 de junho de 2021)	20
2.15	1.0.2 (14 de agosto de 2020)	21
2.16	1.0.0 (27 de abril de 2020)	21
2.17	0,98 (2 de abril de 2020)	22
2.18	0,66 (20 de julho de 2019)	22
2.19	0,64 (9 de julho de 2019)	22
2.20	0,62 (27 de janeiro de 2019)	23
2.21	0,60 (3 de dezembro de 2018)	23
2.22	0,58 (11 de agosto de 2018)	23
2.23	0.56.5 (21 de maio de 2018)	23
2.24	0.56.4 (21 de maio de 2018)	23
2.25	0.56.3 (21 de abril de 2018)	24
2.26	0.56.2 (10 de abril de 2018)	24
2.27	0.56.1 (10 de abril de 2018)	24

2.28	0,56 (9 de abril de 2018)	24
2.29	0,54 (8 de abril de 2018)	24
2.30	0.52.1 (21 de março de 2018)	24
2.31	0.52.1 (21 de março de 2018)	25
2.32	0,52 (19 de março de 2018)	25
2.33	0,50 (15 de março de 2018)	25
2.34	0,48 (13 de março de 2018)	25
2.35	0,46 (13 de março de 2018)	25
2.36	0,44 (12 de março de 2018)	25
2.37	0,42 (4 de fevereiro de 2018)	26
2.38	0,40 (4 de fevereiro de 2018)	26
2.39	0,38 (16 de janeiro de 2018)	26
2.40	0,36 (14 de dezembro de 2017)	27
2.41	0,34 (14 de dezembro de 2017)	27
2.42	0,32 (14 de dezembro de 2017)	27
2.43	0,30 (12 de dezembro de 2017)	27
2.44	0,24 (6 de dezembro de 2017)	27
2.45	0,22 (4 de dezembro de 2017)	27
2.46	0,18 (2 de dezembro de 2017)	27
2.47	0,16 (6 de novembro de 2017)	27
2.48	0,14 (25 de outubro de 2017)	27
2.49	0,12 (6 de outubro de 2017)	27



▲TRENDS.EARTH é um software livre e de código aberto, licenciado sob a [GNU General Public License, version 2.0 or later](#).

Há vários componentes na ferramenta ▲TRENDS.EARTH. O primeiro é um plugin QGIS que suporta cálculos de indicadores, acesso a dados brutos, relatórios e produção de mapas impressos. O código para o plugin e mais instruções sobre como instalá-lo, se você deseja modificar o código, estão em [trends.earth](#) Repositório no GitHub.

O ▲TRENDS.EARTH plugin para QGIS é suportado por vários scripts Python diferentes que permitem o cálculo dos vários indicadores no Google Earth Engine (GEE). Esses scripts estão na subpasta «gee» dentro do repositório GitHub. Os scripts GEE são suportados pelo módulo Python *landdegradation*, que inclui código para processar entradas e saídas do plugin, bem como outras funções comuns que suportam o cálculo de integrais NDVI, significância estatística e outro código compartilhado. O código para este módulo está disponível no arquivo *landdegradation* repositório no GitHub.

Mais detalhes abaixo sobre como contribuir com o Trends.Earth trabalhando no código do plugin, modificando o código de processamento ou contribuindo para traduzir o site e o plugin.

## 1.1 Modificando o código QGIS Plugin

### 1.1.1 Fazendo o download do código trends.earth

O código Trends.Earth para os scripts do plugin e do Google Earth Engine que o suporta está localizado no GitHub na seção [trends.earth](#) repositório. Clone este repositório em um local conveniente em sua máquina para garantir que você tenha a versão mais recente do código.

Existem vários ramos diferentes do repositório trends.earth em desenvolvimento ativo. O plug-in suporta oficialmente o QGIS 3 e a maior parte do desenvolvimento está a ocorrer no ramo «develop». O ramo «qgis 2» é a versão mais antiga do plug-in e suporta versões do QGIS 2 a partir da 2.18.

## 1.1.2 Instalando dependências

### Python

O plugin é codificado em Python. Além de ser usado para executar o plugin através do QGIS, o Python também é usado para oferecer suporte ao gerenciamento do plugin (alterando a versão, instalando versões de desenvolvimento etc.). Embora Python esteja incluído no QGIS, você também precisará de uma versão local do Python que possa ser configurada com o software necessário para gerenciar o plugin. A maneira mais fácil de gerenciar várias versões do Python é através da distribuição [Anaconda](#). Para o trabalho de desenvolvimento do plugin, é necessário o Python 3. Para baixar o Python 3.7 (recomendado) no Anaconda, [consulte esta página](#).

### Dependências do Python

Para trabalhar com o código trends.earth, é necessário ter o Invoke instalado em sua máquina, além de vários outros pacotes usados para gerenciar a documentação, traduções, etc. Esses pacotes estão todos listados no arquivo «dev» de requisitos para Trends.Earth, para que eles possam ser instalados navegando em um prompt de comando até a raiz da pasta de código trends.earth e digitando:

```
pip install -r requirements-dev.txt
```

---

**Nota:** Se você estiver usando Anaconda, primeiro deverá ativar um ambiente virtual Python 3.7 antes de executar o comando acima (e qualquer um dos outros comandos de chamada listados na página). Uma maneira de fazer isso é iniciando um «Anaconda prompt», [seguindo as instruções nesta página do Anaconda](#).

---

### PyQt

PyQt5 é o kit de ferramentas gráficas usado pelo QGIS3. Para compilar a interface do usuário do Trends.Earth for QGIS3, você precisa instalar o PyQt5. Este pacote pode ser instalado a partir do pip usando:

```
pip install PyQt5
```

---

**Nota:** PyQt4 é o kit de ferramentas gráficas usado pelo QGIS2. A melhor fonte para este pacote para Windows é o conjunto de pacotes mantidos por Christoph Gohlke na UC Irvine. Para baixar o PyQt4, selecione [o pacote apropriado nesta página](#). Escolha o arquivo apropriado para a versão do Python que você está usando. Por exemplo, se você estiver usando o Python 2.7, escolha a versão com «cp27» no nome do arquivo. Se você estiver usando o Python 3.7, escolha a versão com «cp37» no nome do arquivo. Escolha «amd64» para python de 64 bits e «win32» para python de 32 bits.

Após fazer o download no link acima, use pip para instalá-lo. Por exemplo, para a roda de 64 bits para Python 3.7, você executaria:

```
pip install PyQt4-4.11.4-cp37-cp37m-win_amd64.whl
```

---

### 1.1.3 Alterando a versão do plugin

A convenção para Trends.Earth é que os números de versão que terminam em um número ímpar (por exemplo, 0,65) são versões de desenvolvimento, enquanto as versões que terminam em um número par (por exemplo 0,66) são versões de lançamento. As versões de desenvolvimento do plug-in nunca são liberadas via o repositório QGIS, para que eles nunca sejam vistos pelos usuários normais do plug-in. Versões de desenvolvimento com números ímpares são usadas pela equipe de desenvolvimento do Trends.Earth ao testar novos recursos antes de seu lançamento público.

Se você deseja fazer alterações no código e baixou uma versão pública do plugin (que termina em um número par), a primeira etapa é atualizar a versão do plugin para o próximo número ímpar seqüencial. Portanto, por exemplo, se você baixou a versão 0.66 do plugin, seria necessário atualizar a versão para 0.67 antes de começar a fazer as alterações. Existem vários locais no código em que a versão é mencionada (assim como em todos os scripts GEE), portanto, há uma tarefa de chamada para ajudar na alteração da versão. Para alterar a versão para 0,67, você deve executar:

```
invoke set-version -v 0.67
```

A execução do comando acima atualizará o número da versão em todos os locais mencionados no código. Para evitar confusão, nunca mude a versão para uma que já foi lançada - sempre AUMENTE o valor da tag da versão para o próximo número ímpar.

### 1.1.4 Testando alterações no plugin

Depois de fazer alterações no código do plugin, você precisará testá-los para garantir que o plugin se comporte conforme o esperado e para garantir que não haja bugs ou erros. O plugin deve passar por testes extensivos antes de ser lançado no repositório QGIS (onde pode ser acessado por outros usuários) para garantir que quaisquer alterações no código não quebrem o plugin.

Para testar as alterações feitas no plugin no QGIS, será necessário instalá-lo localmente. Existem tarefas de chamada que auxiliam nesse processo. A primeira etapa antes da instalação do plug-in é garantir que você o tenha configurado com todas as dependências necessárias para executar no QGIS. Para fazer isso, execute:

```
invoke plugin-setup
```

A tarefa acima só precisa ser executada imediatamente após o download do código trends.earth ou se forem feitas alterações nas dependências do plugin. Por padrão, o `plugin-setup` reutiliza todos os arquivos em cache na sua máquina. Para começar do zero, adicione o sinalizador `-c` (clean) ao comando acima.

Após executar o `plugin-setup`, você está pronto para instalar o plugin na pasta de plug-ins do QGIS em sua máquina. Para fazer isso, execute:

```
invoke plugin-install
```

Depois de correr o comando em cima, precisará de ou 1) reiniciar o QGIS, ou 2) usar o [Plugin Reloader](#) para recarregar o plug-in Trends.Earth, de forma a ver os efeitos das alterações que fez.

Por padrão, `plugin-install` substituirá os arquivos de plugin existentes em sua máquina, mas deixará em vigor quaisquer dados (limites administrativos etc.) que o plugin possa ter baixado. Para começar do zero, adicione o sinalizador `-c` (clean) ao comando acima. Pode ser necessário fechar o QGIS para executar com êxito uma instalação limpa do plugin usando o sinalizador `-c`.

**Nota:** Por padrão, a instalação assume que você deseja instalar o plugin a ser usado no QGIS3. Para instalar o plug-in para uso no QGIS3, adicione o sinalizador `-v 2` ao comando `plugin-install`. Lembre-se de que o plugin pode ou não ser totalmente funcional no QGIS3 - o plugin foi originalmente projetado para o QGIS2 e ainda está sendo testado no QGIS3.

## 1.1.5 Sincronizando e implementando alterações nos binários

Para acelerar os cálculos no Trends.Earth, algumas das ferramentas permitem o uso de binários pré-compilados que foram compilados usando o `numba`. O Numba é um compilador de código aberto que pode compilar código Python e NumPy, tornando-o mais rápido do que quando é executado como Python comum. Para evitar que os usuários do Trends.Earth precisem baixar o Numba e todas as suas dependências, a equipe do Trends.Earth disponibiliza binários pré-compilados para download, se os usuários optarem por instalá-los.

Para gerar binários pré-compilados para o SO, bitness (32/64 bits) e a versão Python que você está executando em sua máquina, use:

```
invoke binaries-compile
```

**Nota:** Você precisará de um compilador C ++ para que o comando acima funcione. No Windows, consulte [esta página do github](#) para detalhes sobre como instalar o compilador Microsoft Visual C ++ necessário para sua versão do Python. No MacOS, você provavelmente precisará instalar o Xcode. No Linux, instale a versão apropriada do GCC.

Para disponibilizar publicamente os binários, eles são distribuídos por meio de um bucket do Amazon Web Services S3. Para fazer upload dos binários gerados com o comando acima no bucket, execute:

```
invoke binaries-sync
```

**Nota:** O comando acima falhará se você não tiver chaves que permitam acesso de gravação ao bucket `trends.earth` no S3.

O comando acima sincronizará cada arquivo binário individual com o S3. No entanto, os usuários da ferramenta baixam os binários como um único arquivo zip vinculado à versão do plug-in que eles estão usando. Para gerar esse arquivo zip para que ele possa ser acessado pelos usuários do Trends.Earth, execute:

```
invoke binaries-deploy
```

**Nota:** O comando acima falhará se você não tiver chaves que permitam acesso de gravação ao bucket `trends.earth` no S3.

## 1.1.6 Construindo um arquivo ZIP contendo o plugin

Existem várias tarefas de chamada para ajudar na criação de um arquivo ZIP para implantar o plugin no repositório QGIS ou para compartilhar a versão de desenvolvimento do plugin com outras pessoas. Para empacotar o plugin e todas as suas dependências em um arquivo ZIP que pode ser instalado seguindo o [procedimento descrito no Leiam do Trends.Earth](#), execute:

```
invoke zipfile-build
```

Este comando criará uma pasta chamada `build` na raiz da pasta de código `trends.earth`, e nessa pasta criará um arquivo chamado `LDMP.zip`. Este arquivo pode ser compartilhado com outras pessoas, que podem usá-lo para [instalar manualmente o Trends.Earth](#). Isso pode ser útil se houver a necessidade de compartilhar os recursos mais recentes com alguém antes que eles estejam disponíveis na versão divulgada publicamente do plugin.



## 1.1.7 Implementando o Arquivo ZIP da versão em desenvolvimento

A página Trends.Earth GitHub fornece um link para um arquivo ZIP que permite que usuários que não sejam desenvolvedores acessem a versão de desenvolvimento do Trends.Earth. Para criar um arquivo ZIP e disponibilizá-lo nessa página (o arquivo ZIP está armazenado no S3), execute:

```
invoke zipfile-deploy
```

Este comando irá empacotar o plugin e copiá-lo para <https://s3.amazonaws.com/trends.earth/sharing/LDMP.zip>.

**Nota:** O comando acima falhará se você não tiver chaves que permitam acesso de gravação ao bucket `trends.earth` no S3.

## 1.2 Modificando o código de processamento do Earth Engine

Os scripts de processamento do Google Earth Engine (GEE) usados pelo Trends.Earth são todos armazenados na pasta «gee», na pasta principal `trends.earth`. Para que esses scripts sejam acessíveis aos usuários do plugin `trends.earth` QGIS, eles precisam ser implantados no serviço `api.trends.earth` que a Conservação Internacional mantém para permitir que os usuários do plugin usem o Earth Engine sem a necessidade de saber como programar ou ter contas de usuário individuais no GEE. A seguir, descrevemos como testar e implantar scripts GEE para serem usados com o Trends.Earth.

### 1.2.1 Configurando dependências

#### trends.earth-CLI

O pacote Python «trends.earth-CLI» é necessário para trabalhar com o servidor `api.trends.earth`. Este pacote está localizado no GitHub no repositório [trends.earth-CLI](#).

O primeiro passo é clonar este repositório na sua máquina. Recomendamos que você clone o repositório na mesma pasta em que salvou o código `trends.earth`. Por exemplo, se você tem uma pasta «Código» em sua máquina, clone ambos repositórios `trends.earth` (com os códigos para o plugin QGIS e scripts GEE associados) e `trends.earth-CLI` repositório na mesma pasta.

Quando você configura seu sistema como recomendado acima, o `trends.earth-CLI` funcionará com as tarefas de chamada usadas para gerenciar o `trends.earth` sem nenhuma modificação. No entanto, se você baixar o `trends.earth-CLI` em uma pasta diferente, será necessário adicionar um arquivo chamado «`invoke.yaml`» na raiz do repositório `trends.earth` e, nesse arquivo, informe ao Trends.Earth onde localizar o código `trends.earth-CLI`. Esse arquivo YAML deve ter a seguinte aparência (se você baixou o código no Windows em uma pasta chamada «`C:/Users/azvol/Code/trends.earth-CLI/tecli`»):

```
gee:
  tecli: "C:/Users/azvol/Code/trends.earth-CLI/tecli"
```

Novamente, você **não** precisa adicionar esse arquivo `.yaml` se configurar o sistema conforme recomendado acima.

## docker

O pacote trends.earth-CLI requer `docker` para funcionar. Siga estas instruções para instalar a janela de encaixe no Windows e estas instruções para instalar a janela de encaixe no Mac OS <<https://docs.docker.com/docker-for-mac/install/>>. Se você estiver executando o Linux, siga as instruções nesta página que são apropriados para a distribuição Linux que você está usando.

### 1.2.2 Testando um script do Google Earth Engine localmente

Depois de instalar o pacote trends.earth-CLI, você precisará configurar um arquivo `.tecli.yml` com um token de acesso a uma conta de serviço GEE para testar scripts no GEE. Para configurar a conta de serviço GEE para o `tecli`, primeiro obtenha a chave da sua conta de serviço no formato JSON (no console do Google Cloud) e, em seguida, codifique-a em base64. Forneça a chave codificada base64 ao `tecli` com o seguinte comando:

```
invoke tecli-config set EE_SERVICE_ACCOUNT_JSON key
```

onde «chave» é a chave da conta de serviço no formato JSON codificado em base64.

Ao converter um script que especifica o código a ser executado no GEE de JavaScript para Python, ou ao fazer modificações nesse código, pode ser útil testar o script localmente, sem implementá-lo no servidor `api.trends.earth`. Para fazer isso, use a tarefa de chamada `run`. Por exemplo, para testar o script «`land_cover`», vá para o diretório raiz do código Trends.Earth e, em um prompt de comando, execute:

```
invoke tecli-run land_cover
```

Isso usará o pacote trends.earth-CLI para criar e executar um contêiner de `docker` que tentará executar o script «`land_cover`». Se houver algum erro de sintaxe no script, eles aparecerão quando o contêiner for executado. Antes de enviar um novo script para `api.trends.earth`, sempre verifique se `invoke tecli-run` é capaz de executar o script sem erros.

Ao usar `invoke tecli-run`, você pode receber um erro dizendo:

```
Invalid JWT: Token must be a short-lived token (60 minutes) and in a reasonable timeframe. Check your iat and exp values and use a clock with skew to account for clock differences between systems.
```

Este erro pode ser causado se o relógio no contêiner do `docker` ficar fora de sincronia com o relógio do sistema. Reiniciar `docker` deve corrigir esse erro.

### 1.2.3 Implantando um script GEE em api.trends.earth

Quando você terminar de testar um script GEE e desejar acessá-lo usando o plugin QGIS (e por outros usuários do Trends.Earth), poderá implantá-lo no servidor `api.trends.earth`. A primeira etapa do processo é efetuar login no servidor `api.trends.earth`. Para fazer login, execute:

```
invoke tecli-login
```

Você será solicitado a fornecer um nome de usuário e senha. Estes são os mesmos usuário e a senha que você usa para fazer login no Trends.Earth a partir do plugin QGIS. **Se você não for um administrador, poderá fazer login, mas o comando abaixo falhará.** Para fazer upload de um script (por exemplo, o script «`land_cover`») para o servidor, execute:

```
invoke tecli-publish -s land_cover
```

Se esse script já existir no servidor, você será perguntado se deseja substituir o script existente. Tenha muito cuidado ao fazer o upload de scripts com versões pares, pois esses são scripts publicamente disponíveis, e quaisquer erros que você cometer afetarão qualquer pessoa que esteja usando o plugin. Sempre que você estiver testando, use números de versão de desenvolvimento (números de versão ímpares).

Depois de publicar um script no servidor, você pode usar a tarefa *tecli-info* para verificar o status do script (para saber se ele foi implementado com sucesso - embora a criação do script possa demorar alguns minutos). Para verificar o status de um script implantado, execute:

```
invoke tecli-publish -s land_cover
```

Se você estiver fazendo uma nova versão do plugin e quiser fazer o upload de TODOS os scripts GEE de uma só vez (isso é necessário sempre que o número da versão do plugin for alterado), execute:

```
invoke tecli-publish
```

Novamente - nunca execute o procedimento acima em uma versão divulgada publicamente do plugin, a menos que você pretenda substituir todos os scripts publicamente disponíveis usados pelo plugin.

### 1.3 A editar modelos de camada vetorial

O Trends.Earth permite aos utilizadores digitalizarem novas características vetoriais, para delinear áreas de especial interesse.

Por agora apenas são suportadas camadas «falsos positivos/negativos», mas podem ser adicionadas mais se necessário. Qualquer camada de vector é criada a partir dos ficheiros modelo GeoPackage, que podem ser encontrados dentro da pasta `data/error_recode` do diretório de instalação do plugin. Para cada tipo de vector, existem 6 ficheiros modelo, um para cada idioma oficial da ONU. O código de idioma ISO é adicionado como sufixo ao nome do ficheiro. Isto é necessário para fornecer etiquetas localizadas nos formulários de atributos. Quando a criação da camada de vector é pedida, o QGIS irá procurar o ficheiro modelo tendo em conta a localização do QGIS, como opção alternativa será utilizada a versão em Inglês do ficheiro modelo.

Para alterar o esquema da camada é necessário alterar os ficheiros modelo correspondentes na pasta `data/error_recode` do diretório de instalação do plugin. Além disso, o ficheiro modelo contém embutidos um estilo e configuração de formulário de atributo predefinidos, que serão aplicados automaticamente à camada ao carregar no QGIS.

Para apresentar gráficos no formulário atributo, é usado um widget QML integrado. Os dados para os gráficos estão guardados na tabela de atributos da camada vetorial. Os valores dos campos correspondentes são extraídos com a ajuda de expressões.

O código para gerar gráficos tem este aspeto:

```
import QtQuick 2.0
import QtCharts 2.0

ChartView {
    width: 380
    height:200
    margins {top: 0; bottom: 0; left: 0; right: 0}
    backgroundColor: "#eeeeec"
    legend.alignment: Qt.AlignBottom
    antialiasing: true
    ValueAxis {
        id: valueAxisY
        min: 0
    }
}
```

(continues on next page)

(continuação da página anterior)

```

    max: 100
  }

  BarSeries {
    id: mySeries
    axisY: valueAxisY
    axisX: BarCategoryAxis { categories: ["Productivity", "Land cover", "Soil_
↪organic carbon"] }
    BarSet { label: "Degraded"; color: "#9b2779"; values: [expression.evaluate("\
↪prod_deg\""), expression.evaluate("\land_deg\""), expression.evaluate("\soil_deg\
↪")] }
    BarSet { label: "Improved"; color: "#006500"; values: [expression.evaluate("\
↪prod_imp\""), expression.evaluate("\land_imp\""), expression.evaluate("\soil_imp\
↪")] }
    BarSet { label: "Stable"; color: "#ffffe0"; values: [expression.evaluate("\
↪prod_stab\""), expression.evaluate("\land_stab\""), expression.evaluate("\soil_
↪stab\"")] }
  }
}

```

Para extrair o valor do campo, é usada a função `expression.evaluate("\prod_deg\"")`, o único argumento que aceita é o nome do campo. Para camadas de falsos positivos/negativos, o gráfico contém três indicadores, produtividade, cobertura de solo e carbono orgânico do solo. Para cada indicador, o plug-in mantém três valores da área do polígono: estável, degradado e melhorado. Por exemplo, no caso do indicador produtividade, os campos serão:

- `prod_deg` - produtividade degradada
- `prod_stab` - produtividade estável
- `prod_imp` - produtividade melhorada

A mesma abordagem de nomenclatura é aplicada à cobertura do solo (campos `land_*`) e ao carbono orgânico do solo (campos `soil_*`).

O cálculo da percentagem de área é feito com a função de expressão personalizada, o seu código pode ser encontrado no ficheiro `charts.py`, no diretório de raiz do plug-in. A função está otimizada para funcionar com polígonos grandes e utiliza o seguinte fluxo de trabalho. Para uma dada geometria, encontra uma `bbox` e extrai o subconjunto de matriz, usando esta `bbox`. Executa a rasterização geométrica na memória e aplica-a como máscara para o cálculo matricial. Depois conta o número de pixels que têm um valor específico e calcula percentagem. Uma vez que a contagem de pixels está construída sobre funções vetoriais `numpy`, é muito rápida, mesmo para polígonos de grande dimensão.

Na primeira tentativa de editar uma camada vetorial, será apresentado um diálogo ao utilizador, onde este deverá escolher os conjuntos de dados a usar para os indicadores. Depois o plug-in irá configurar os valores de expressão por defeito para todos os campos indicadores, para que o valor seja atualizado em cada alteração de geometria.

## 1.4 Gestão dos metadados do conjunto de dados

Os metadados dos conjuntos de dados são armazenados no formato QGIS QMD. Estes ficheiros QMD podem ser criados para cada matriz individualmente e também para todo o conjunto de dados. O diálogo do editor de metadados é aberto a partir do menu **Editar metadados** na plataforma Trends.Earth.

Quando o conjunto de dados é exportado para ZIP, a conversão para ISO XML é realizada usando transformação XSLT. As transformações correspondentes estão localizadas na sub-pasta `data\xsl` da pasta de instalação do plug-in.

## 1.5 A atualizar estrutura de relatório

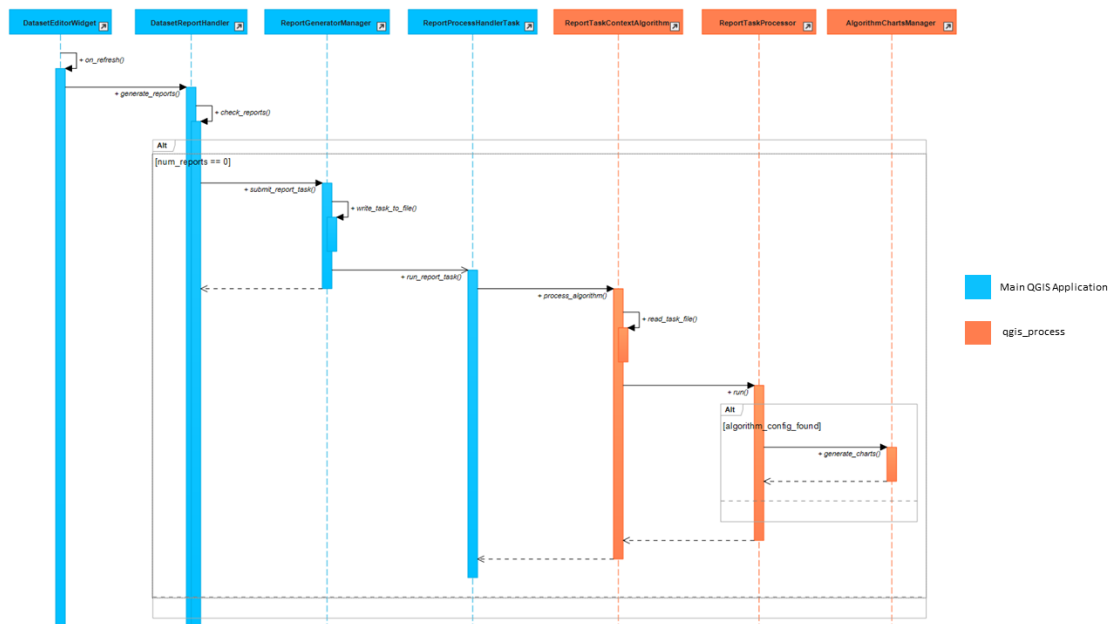
### 1.5.1 Resumo da estrutura de relatório

A Estrutura de Relatórios foi concebida para ser extensível, proporcionando ao mesmo tempo interatividade ao utilizador, através de operações sem bloqueio. A Estrutura baseia-se fortemente nas classes `QgsProject` e `QgsPrintLayout`, que não são seguras, daí a utilização de `qgis_process` para fazer o trabalho pesado de gerar os relatórios (e gráficos). Pode encontrar mais informação sobre `qgis_process` [aqui](#).

A caixa de ferramentas executa dois passos principais quando gera relatórios (e gráficos) para as camadas por defeito num trabalho:

1. Cria um objeto `ReportTaskContext` que constitui um objeto `ReportConfiguration` (ver: [ref:config\\_report\\_params](#)) e um objeto `Job` que é representado no painel de **Conjuntos de dados**. Este objeto `ReportTaskContext` é serializado para um ficheiro JSON e depois passado como um dos argumentos num objeto `ReportProcessHandlerTask` (que herda de `QgsTask`).
2. O objeto `ReportProcessHandlerTask` inicia uma instância separada de `qgis_process` e passa o caminho para o ficheiro JSON como uma entrada para o algoritmo de processamento `trendsearch:reporttask`. Este é um wrapper fino que desserializa o ficheiro para o objeto `ReportTaskContext` e o passa para um objeto `ReportTaskProcessor`, que é responsável por gerar os relatórios e o projeto QGIS do trabalho. Para algoritmos que requerem gráficos, o objeto `ReportTaskProcessor` passa o objeto do trabalho para um objeto `AlgorithmChartsManager` que verifica se existe uma configuração de gráficos definida para o algoritmo do trabalho. Se existir, gera os gráficos correspondentes como ficheiros PNG. (Ver [Adicionar Configurações de Gráfico](#) para mais informações sobre configurações de gráficos)

O diagrama em baixo apresenta uma ilustração de alto nível deste processo:



\* *Clique na imagem para uma vista ampliada.*

**Nota:** Alguns dos nomes de funções no diagrama em cima foram simplificados para fins ilustrativos. As classes cima mencionadas podem ser encontradas nos módulos `LDMP.reports` e `LDMP.processing_provider.report`.

## 1.5.2 A adicionar variáveis de layout do relatório

As variáveis do relatório fornecem informações de contexto relacionadas com um trabalho, camada (ou banda) ou `report_settings` durante o processo de execução do relatório. Atualmente, a caixa de ferramentas suporta as variáveis listadas na secção `layout_expr_vars`.

Cada variável é definida como uma `namedtuple` no módulo `LDMP.reports.expressions` e é subsequentemente atualizada e avaliada pelo objeto `ReportTaskProcessor`.

Siga as orientações em baixo sobre como adicionar novas variáveis de trabalho ou de camada atual.

### Variável de trabalho

Permite que a informação sobre o trabalho atual - a ser executado - seja adicionada ao layout de um relatório. A informação sobre cada variável de trabalho é encapsulada num objeto `JobAttrVarInfo` que tem quatro atributos:

Nome de atributo	Descrição	Tipo de dados	Valor por defeito
<code>job_attr</code>	Nome de atributo de um objeto <code>Job</code> , como usado numa notação dot. Por exemplo, <code>id</code> corresponde a <code>job.id</code> . Pode até usar a notação dot para referir-se a atributos em classes internas de cadeias, por exemplo, <code>results.uri.uri</code> .	String	N/D
<code>var_name</code>	Nome da variável de layout do relatório. Deve ter o prefixo <code>te_job_</code> .	String	N/D
<code>default_value</code>	Valor por defeito a usar para <code>var_name</code> , aplicado principalmente ao desenhar layouts.	objeto	objeto
<code>fmt_func</code>	Um objeto de função que será usado para converter o valor do atributo do trabalho para um formato compatível com expressões QGIS. Por exemplo, <code>str</code> pode ser usado para converter o valor do <code>id</code> de um trabalho, de UUID para string. Também pode usar funções lambda aqui.	objeto de função	Nenhum

O fragmento de código em baixo mostra como adicionar uma variável `te_job_result_name` que corresponde a `job.results.name`.

```
# LDMP/reports/expressions.py
def _job_attr_var_mapping() -> typing.List[JobAttrVarInfo]:
    return [
        ...
        JobAttrVarInfo('results.name', 'te_job_result_name', '', str),
        ...
    ]
```

### Variável de Camada

Fornecer informação sobre a camada raster atual a ser executada. Esta informação de variável está encapsulada num objeto `LayerVarInfo` que consiste em três atributos:

Nome de atributo	Descrição	Tipo de dados	Valor por defeito
<code>var_name</code>	Nome do variável de esquema de relatório of the report layout variable. Deve ter como prefixo <code>te_current_layer_</code> .	String	N/D
<code>default_value</code>	Valor por defeito a usar para <code>var_name</code> , aplicado principalmente ao desenhar layouts.	objeto	objeto
<code>fmt_func</code>	Um objeto de função que será utilizado para extrair e/ou converter um valor de um objeto <code>QgsRasterLayer</code> para um formato compatível com expressões QGIS. Pode utilizar funções lambda aqui. Por exemplo, <code>lambda layer: layer.name()</code> devolve o nome da camada.	objeto de função	Nenhum

O fragmento de código em baixo mostra como adicionar uma variável `te_current_layer_height` que corresponde à altura do camada raster.

```
# LDMP/reports/expressions.py
def _current_job_layer_var_mapping() -> typing.List[LayerVarInfo]:
    return [
        ...
        LayerVarInfo(
            'te_current_layer_height',
            '',
            lambda layer: layer.height()
        )
        ...
    ]
```

**Nota:** Estas variáveis estão apenas disponíveis no âmbito do esquema.

### 1.5.3 Adicionar Configurações de Gráfico

Os gráficos podem ser agrupados utilizando o objeto de configuração de gráficos que corresponde a um algoritmo específico. A definição de uma nova configuração de gráficos é um processo com três passos:

1. Crie uma nova classe de gráficos que herda de `BaseChart` no módulo `LDMP.reports.charts`. Implemente a função `export` para especificar o tipo de gráfico, propriedades, etc utilizando a biblioteca Python `Plotly` incluída com o QGIS. Por fim, dentro da função `export`, chame a função `save_image` para escrever o código de objeto `Figure Plotly` como um ficheiro de imagem, utilizando qualquer um dos formatos suportados pela classe `QImageWriter` de Qt. Também pode especificar o caminho em relação com diretório de resultado de raiz, que também está disponível como um atributo na classe base. Veja o fragmento de código em baixo:

```
# LDMP/reports/charts.py
class MyCustomChart(BaseChart):
    def export(self) -> typing.Tuple[bool, list]:
        status = True
        messages = []

        # Create chart Figure using Plotly and set properties
        fig = go.Figure(...)

        # Add warning or error messages
        messages.append('Colour list not supported.')
```

(continues on next page)

(continuação da página anterior)

```

# Set image path in dataset's reports folder
img_path = f'{self.root_output_dir}/chart-NDVI.png'

# Save image and append its path
self.save_image(fig, img_path)
self._paths.append(img_path)

return status, messages

```

Pode consultar a classe `UniqueValuesPieChart` para um exemplo mais completo.

2. Crie uma classe configuração de gráfico `chart configuration class` que herda de `BaseAlgorithmChartsConfiguration` e implemente a função `_add_charts`. A classe de configuração de gráfico define basicamente quais os gráficos que serão utilizados para um dado algoritmo. O atributo `layer_band_infos` é uma lista de objetos `LayerBandInfo` que contém a camada e dados `band_info` necessários para produzir os gráficos. Pode consultar a classe `LandCoverChartsConfiguration` para um exemplo mais completo.
3. Por fim, mapeie um algoritmo (nome) à classe de configuração de gráficos correspondente na classe `AlgorithmChartsManager` tal como apresentado em baixo:

```

# LDMP/reports/charts.py
class AlgorithmChartsManager:
    def _set_default_chart_config_types(self):
        ...
        self.add_alg_chart_config('land-cover', LandCoverChartsConfiguration)
        self.add_alg_chart_config('productivity', ↵
↵MyCustomLandProductivityChartsConfiguration)
        ...

```

A classe `AlgorithmChartsManager`, que é instanciada no objeto `ReportTaskProcessor`, irá criar um novo objeto de configuração de gráficos para o algoritmo de um trabalho correspondente quando os relatórios estiverem a ser gerados.

## 1.6 Contribuindo para a documentação

### 1.6.1 Sumário

A documentação para Trends.Earth é produzida usando o `Sphinx`, e está escrito em `reStructuredText` <<http://docutils.sourceforge.net/rst.html>>\_ formato. Se você não for familiarizado com nenhuma dessas ferramentas, consulte a documentação para obter mais informações sobre como elas são usadas.

A documentação do Trends.Earth é armazenada na pasta «docs», no diretório principal `trends.earth`. Dentro dessa pasta, há vários arquivos e pastas importantes que você deve conhecer:

- `build`: contém a documentação da build para `trends.earth` (no formato PDF e HTML). Observe que ele só aparecerá na sua máquina depois de executar a tarefa de chamada `docs-build`.
- `i18n`: contém traduções da documentação para outros idiomas. Os arquivos aqui são normalmente processados automaticamente usando tarefas com o comando «`invoke`», portanto, você nunca deve ter motivos para modificar nada nesta pasta.
- `recursos`: contém todos os recursos (principalmente imagens ou PDFs) mencionados na documentação. Atualmente, existe apenas uma pasta («`EN`», em inglês), pois todas as imagens na documentação são da versão



em inglês do plugin - pastas adicionais apropriadas podem ser adicionadas em «resources» com códigos de duas letras identificando o idioma para incluir imagens específicas para um idioma específico.

- source: contém os arquivos de origem reStructuredText que definem a documentação (esse é o texto em inglês real da documentação e os arquivos que você provavelmente precisará modificar).

## 1.6.2 Instalando dependências

### Dependências do Python

Para trabalhar com a documentação, você deve ter os pacotes invoke, Sphinx, sphinx-intl e sphinx-rtd-theme (o tema do site Trends.Earth) instalados em sua máquina. Esses pacotes estão todos listados no arquivo de requisitos «dev» do Trends.Earth, para que possam ser instalados navegando em um prompt de comando até a raiz da pasta de código trends.earth e digitando:

```
pip install -r requirements-dev.txt
```

### LaTeX

O LaTeX é usado para produzir saídas em PDF da documentação para Trends.Earth.

Para instalar LaTeX no Windows, siga o processo descrito aqui para instalar a distribuição proTeXt do LaTeX a partir do arquivo zip disponível aqui. O instalador do LaTeX é bastante grande (vários GB), portanto, pode levar algum tempo para baixar e instalar.

No MacOS, o MacTeX é uma boa opção e pode ser instalado seguindo as instruções aqui.

No Linux, a instalação do LaTeX deve ser muito mais fácil - use o gerenciador de pacotes da sua distribuição para encontrar e instalar qualquer distribuição do LaTeX incluída por padrão.

### Qt Linguist

O Qt Linguist também é necessário para extrair strings do código e da GUI para tradução. O comando «lrelease» deve estar disponível e no seu caminho. Tente tentar

```
lrelease
```

dentro de uma janela do terminal. Se o arquivo não for encontrado, você precisará instalar o Qt Linguist. Esta página é uma fonte de instaladores para o Qt Linguist. Depois de instalar o Qt Linguist, adicione a pasta que contém lrelease ao seu caminho para que o script de chamada Trends.Earth possa encontrá-lo.

## 1.6.3 Atualizando e preparando a documentação

Depois de instalar os requisitos do pacote sphinx, você estará pronto para começar a modificar a documentação. Os arquivos a serem modificados estão localizados na pasta «docs\source». Depois de fazer alterações nesses arquivos, você precisará criar a documentação para visualizar os resultados. Existem duas versões da documentação do Trends.Earth: uma versão HTML (usada para o site) e uma versão PDF (para download offline). Para criar a documentação do Trends.Earth, use a tarefa «docs-build» do invoke. Por padrão, esta tarefa cria a documentação completa do Trends.Earth, em HTML e PDF, para todos os idiomas suportados. Isso pode levar algum tempo para ser executado (até algumas horas). Se você está apenas testando os resultados de algumas pequenas alterações na documentação, geralmente é melhor usar a opção -f (de «fast» = rápido). Esta opção criará apenas a documentação HTML em inglês, que deve levar apenas alguns segundos. Para criar usando a opção rápida, execute:

```
invoke docs-build -f
```

O comando acima levará alguns segundos para ser executado e, se você procurar em «docs\build\html\en», verá a versão HTML da documentação. Carregue o arquivo «index.html» em um navegador da Web para ver como fica.

Para criar a documentação completa, para todos os idiomas, em PDF e HTML (lembre-se de que isso pode levar algumas horas para ser concluído), execute:

```
invoke docs-build
```

Após executar o comando acima, você verá (em inglês) a documentação HTML em «docs\build\html\en», e os PDFs da documentação em «docs\build\html\en\pdfs».

Se você quiser testar um idioma específico (testar traduções, por exemplo), poderá especificar um código de duas letras identificando o idioma para criar apenas os documentos para esse idioma. Por exemplo, para criar apenas a documentação em espanhol, execute:

```
invoke docs-build -l es
```

Observe que as opções podem ser combinadas; portanto, você pode usar a opção rápida para criar apenas a versão HTML da documentação em espanhol executando:

```
invoke docs-build -f -l es
```

Ao criar a documentação completa para o site, é uma boa idéia remover primeiro todas as versões antigas da documentação, pois elas podem conter arquivos que não são mais usados na documentação atualizada. Para fazer isso, use a opção `-c` (clean):

```
invoke docs-build -c
```

Em geral, a preparação dos documentos DEVE ser concluído sem erros se você planeja compartilhar a documentação ou publicá-la no site. No entanto, ao testar localmente, convém ignorar os erros de documentação que aparecem somente em alguns idiomas (devido a erros de sintaxe decorrentes de erros de tradução etc.) e continuar a compilar a documentação restante, independentemente de haver erros. Para fazer isso, use a opção `-i` (ignorar erros):

```
invoke docs-build -i
```

Sempre que você fizer alterações no texto da documentação, é uma boa ideia enviar as últimas atualizações para o Transifex, para que possam ser traduzidas. Para atualizar as seqüências de caracteres no Transifex com novas alterações, execute:

```
invoke translate-push
```

---

**Nota:** Para executar com êxito o comando acima, você precisará da chave da conta transifex Trends.Earth.

---

## 1.6.4 Preparando a documentação para publicar a nova versão

Antes de liberar a nova documentação, sempre baixe as últimas traduções do Transifex para que todas as traduções estejam atualizadas. Para fazer isso, execute:

```
invoke translate-pull
```

Para criar uma versão pública da documentação (no site ou em PDF), você deve compilar toda a documentação usando `docs-build` sem parâmetros adicionais:

```
invoke docs-build
```

Esse processo deve ser concluído com êxito, sem erros. Se ocorrer algum erro durante o processo, revise a mensagem de erro e faça as modificações necessárias para permitir que a compilação seja concluída com êxito. Depois que a compilação é concluída sem erros, os arquivos estão prontos para serem implementados no site.

---

**Nota:** Ambos os comandos acima também têm opções `-f` (force) que forcem a baixar ou subir as últimas traduções do ou para o Transifex (respectivamente). Use essas opções apenas se tiver MUITO certo do que está fazendo, pois elas podem substituir completamente as traduções no Transifex, resultando em perda de trabalho dos tradutores se as traduções mais recentes ainda não foram submetidas para o repositório github.

---

## 1.6.5 Adicionando novo texto de documentação

Quaisquer novos arquivos `.rst` adicionados à documentação precisam ser adicionados a vários arquivos de configuração para garantir que apareçam no menu de navegação, que sejam traduzidos corretamente, e (para tutoriais) para garantir que sejam gerados em PDF para que possam ser baixados para uso offline.

- `docs\source\index.rst`: adicione novos arquivos `.rst` no local apropriado aqui para garantir que eles estejam vinculados no menu de navegação.
- `.tx\config`: lista os novos arquivos `.rst` aqui (no mesmo formato dos outros arquivos já incluídos) para tornar o software de tradução ciente deles, para que possam ser traduzidos
- `docs\source\conf.py`: se você deseja gerar um arquivo PDF da página do site, deve listar essa página na lista `latex_documents`. Normalmente, fazemos isso apenas para as páginas de tutoriais que queremos disponibilizar aos participantes do workshop em PDFs individuais. Todas as páginas do site serão incluídas na versão PDF do site como um todo, independentemente de estar na lista `latex_documents`.

## 1.6.6 Adicionando novas imagens ou outros recursos

Quaisquer novas imagens ou outros recursos (PDFs etc.) necessários à documentação devem ser adicionados em «`docs/resources/en`». Se desejar, é possível fazer upload de versões diferentes de uma imagem para que a imagem apareça com as traduções adequadas. Isso pode ser útil se você quiser mostrar a interface da GUI no idioma apropriado, por exemplo. Para fazer isso, primeiro faça upload de uma cópia da imagem para «`docs/resources/en`» (com o texto em inglês). Em seguida, crie uma cópia da imagem com texto traduzido e coloque-a na pasta apropriada para esse idioma (por exemplo, uma imagem mostrando traduções em espanhol seria exibida em «`docs/resources/es`»). A versão em inglês da imagem será usada como padrão para todos os idiomas para os quais uma versão nativa da imagem não é fornecida, enquanto uma versão localizada será usada quando disponível.

---

**Nota:** Há outra pasta, `docs\source\static`, usada para reter recursos temporariamente enquanto executa os scripts que preparam a documentação do Trends.Earth. Você pode ter imagens listadas nessa pasta, se alguma vez construiu a documentação nessa máquina. **Esta pasta nunca deve ser usada para adicionar novos recursos** - os novos

recursos sempre devem estar em `docs\\resources\\en` ou, para imagens traduzidas, na pasta específica do idioma apropriada em `docs\\resources`.

---

### 1.6.7 Contribuindo como tradutor

As traduções para o plugin QGIS e também para este site são gerenciadas pelo [transifex](#). Se você quiser contribuir com a tradução do plugin e da documentação (e gostaríamos muito de ter sua ajuda!), pode solicitar para se juntar a nossa equipe por meio de [transifex](#), ou enviando um email para [trends.earth@conservation.org](mailto:trends.earth@conservation.org).

Esta página lista o histórico das versões de [▲TRENDS.EARTH](#).

### 2.1 2.1.8 (January 25, 2023)

- Output JSON field for Prais containing updated false positive/negative calculations.
- Fixes #752 - false positive/negative datasets not showing up in non-English versions

### 2.2 2.1.6 (December 8, 2022)

- Bumps maximum number of land cover classes to 45 (useful particularly for users seeking to use Corine legend)
- Fix issue that was arising on Mac/Linux due to missing QgsPanelWidget ui file
- Various more minor bug fixes

### 2.3 2.1.6 (November 21, 2022)

- Fixes #691 (issue that arose if the base data directory was cleared). When base directory is cleared plugin will now reset the value to the default.
- Fix python error coming up when settings window was closed while datasets screen was not open.

## 2.4 2.1.2 (November 18, 2022)

- Fixes #724 - ensure max number of land cover classes is set to 38 by default (ignoring the previously set default value in qsettings, which used to be 32).
- Fixes #722 - file name casing when code refers to land\_cover\_transition\_matrix\_unccd.json.
- Fixes #726 - make dataset year visible in datasets window for imported datasets

## 2.5 2.1.0 (November 11, 2022)

- Moved plugin settings to live within main QGIS plugin settings window
- Bumped marshmallow-dataclass to 8.5.10 and marshmallow to 3.18.0. This necessitated changes in trends.earth-schemas, so it is possible some old jobs may not load in this new version (suggest re-running them if need be).

## 2.6 2.0.7 (October 31, 2022)

- Enable Excel table generation for SDG 15.3.1 when using custom classes - all the various minor bugs should now be worked out.
- Limit report generation to 3 tries - then quit trying to generate reports until next restart (prevents Trends.Earth continually spawning processes for report generation when they are failing).

## 2.7 2.0.5 (October 19, 2022)

- Fix SDG summary generation when using JRC LPD or WOCAT LPD.

## 2.8 2.0.3 (October 19, 2022)

- Allow bypassing reset of land cover legend when choosing defaults in the SDG tool.
- Disable excel generation when child legend does not have 7 classes.
- Fix report generation when using custom legend.

## 2.9 2.0.1 (October 13, 2022)

- Added ability to completely customize the land cover legend used in Trends.Earth, including the number, name, color, and coding of each class in the legend. This applies to all calculatons using land cover data, including the SDG 15.3.1 tool, and the productivity, soil organic carbon, and land cover indicator tools.
- Reworked land cover import tool to allow importing data and assigning classes to custom legend.
- Misc minor bug fixes.
- Add ability to include false positive/negative dataset in UNCCD report. Next release will include further functionality to apply and report on false positive/negatives.

- This experimental release disables the creation of Excel output tables for the SDG 15.3.1 tool. This functionality will be present in the next (stable) release.

## 2.10 2.0 (July 20, 2022)

- Um interface completamente novo com ecrãs de navegação e carregamento de conjuntos de dados simplificados
- Documentação e website atualizados
- A Trends.Earth está agora traduzida para todas os idiomas oficiais da UN, sendo que estão também planeadas traduções para Português, Suaíli e Persa (para o novo interface).
- Novas funções para suportar o mais recente ciclo de reportagem da UNCCD (incluindo suporte direto para exportar resultador para carregamento no sistema UNCCD PRAIS)
- Novas funções que suportam a avaliação do risco, vulnerabilidade e exposição de seca
- Novas funções de ferramentas de séries de tempo (suporta agora “plotting restrend”, WUE, etc.)

## 2.11 1.0.10 (7 de Julho, 2022)

- Resolver problema com TransformDirection relacionado com mudanças em QGIS 3.22+

## 2.12 1.0.8 (15 de Outubro, 2021)

- Resolver problema com a gravação da última camada para o cálculo de ODS (relacionado com problemas #500 e #505)

## 2.13 1.0.6 (15 de julho de 2021)

- Remova trends.earth-schemas como submódulo e instale através de setup.py (muito mais limpo para gestão de versões, desenvolvimento, etc.)
- Corrigido bug ao carregar dados NDVI de 2020 em código GEE (nome do ativo não estava definido corretamente, por isso 2020 não era carregado)
- Várias tarefas de invocação adicionadas para ajudar no desenvolvimento/lançamento de plug-ins.

## 2.14 1.0.4 (30 de junho de 2021)

- Adicionar WorldPop e «Gridded Population of the World» versão 4 (GPWv4) aos conjuntos de dados disponíveis a partir do Trends.Earth
- Atualizar para permitir acesso à cobertura do solo da ESA-CCI até 2020
- Atualizar conjuntos de dados GPCC e GPCP
- Corrigir bug quando são usados tampões não inteiros
- Correções de pequenos erros de documentação

## 2.15 1.0.2 (14 de agosto de 2020)

- Corrigir código de área urbana para permitir processamento de AOI com áreas entre 10 000 e 25 000 km<sup>2</sup>.
- Adicionar os últimos dados do MERRA2 (até 2019).
- Remover a limitação de área máxima da ferramenta de transferência.
- Incrementar versão httplib2 to 0.18.0.
- Atualizar de GPCC V6 para GPCC V7
- Adicionar dados de desflorestação de 2019 de Hansen et al.
- Atualizar para o código de cores mais recente para dados de degradação (tendo em conta questões de daltonismo vermelho/verde).
- Adicionar suporte para mais tipos de dados em ficheiros de forma de entrada (adicionar PointZ, MultiPoint, MultiPointZ, PolygonZ, MultiPolygonZ).
- Correções de vários bugs para resolver erros de Python que estavam a surgir com algumas mensagens do QMessageBox.

## 2.16 1.0.0 (27 de abril de 2020)

- Adicione capacidade de baixar e usar binários pré-compilados (compilados com o Numba) para acelerar alguns cálculos locais. No momento, isso está disponível apenas para o cálculo da tabela de resumo do ODS 15.3.1, mas eventualmente ele também será expandido para outras ferramentas.
- Relacionado ao item acima, agora existe um botão de configurações «avançadas» na janela de configurações, que permitirá aos usuários baixar binários pré-compilados e ativar ou desativar o log detalhado de mensagens enquanto a ferramenta está em execução. Essas mensagens de log podem ser úteis ao tentar solucionar problemas, se você encontrar alguma.
- Melhora as verificações de validade da geometria em arquivos de entrada e fornece uma mensagem de erro em vez de lançar uma exceção quando houver erros de geometria.
- Corrigir o processamento usando novos dados MODIS (os arquivos não foram atualizados corretamente na última versão)
- Corrigir o código de cores das coberturas da terra na janela de agregação de classes de cobertura da terra
- Adiciona informações de versão mais detalhadas ao diálogo sobre.
- Adiciona detalhes adicionais à ferramenta de download de dados.
- Adiciona o ID do trabalho à janela de downloads para facilitar o relatório de erros.
- Pequenas correções de bugs (classificação de trabalhos e tabelas de downloads).



## 2.17 0,98 (2 de abril de 2020)

- Primeira versão do QGIS3 - muitas correções para atualizar para a API Qt5 e QGIS3.
- Atualiza todas as dependências do plugin para as versões mais recentes a partir de janeiro de 2020.
- Corrige a ferramenta de download de dados para ter 1style padrão para todos os conjuntos de dados disponíveis.
- Começa a mudar para as estruturas QgsProcessing e QgsTask - atualmente apenas a ferramenta carbon é migrada, mas todas as ferramentas serão migradas antes da versão 1.0./
- Formatação de limpeza da planilha de saída da ferramenta de carbono para tornar mais claro o significado de cada coluna.
- Atualiza todos os scripts GEE para usar a versão mais recente da API GEE (0.1.213).
- Salva mais configurações escolhidas nas caixas de diálogo das ferramentas nas sessões do QGIS.
- Limpa o código de buffer, para usar as projeções de Área Igual de Lambert centradas nos centróides de polígono para buffer.
- Movemos a documentação para a pasta docs na raiz do repositório trends.earth.
- Adicionamos mais detalhes sobre como contribuir para o desenvolvimento de Trends.Earth,
- Limpamos o repositório removendo os arquivos de tradução compilados e adicionando esses tipos de arquivo ao arquivo «.gitignore».
- Altera o nome do projeto Transifex para «trendsearch».
- Várias compatibilidade e pequenas correções de bugs.

## 2.18 0,66 (20 de julho de 2019)

- Limitamos a área máxima para tarefas a 10.000.000 km<sup>2</sup>, exceto para tarefas na área urbana, que é limitada a 10.000 km<sup>2</sup>.
- Adicionamos a seção de plano de fundo ao ODS 11.3.1 e atualize o tutorial do ODS 11.3.1.
- Atualizamos o código ODS 11.3.1 para incluir 1998 na série (internamente durante o cálculo) para filtrar o ruído desde o início da série urbana.
- Corrijimos as restrições de data na ferramenta tudo-em-um do ODS 15.3.1 para considerar a disponibilidade do ESA e do MODIS.
- Adiciona uma seção ao leia-me sobre como instalar as versões do Github.
- Atualiza e revisa as traduções em espanhol, atualize as traduções do Google para outros idiomas.

## 2.19 0,64 (9 de julho de 2019)

- Corrige o manuseio de NoData na ferramenta de carbono total.
- Adiciona suporte para os dados Hansen 2018 na ferramenta de carbono total.
- Adiciona suporte para dados globais de biomassa de 30m de carbono total do Wood's Hole
- Define o ano final máximo para que uma ferramenta ODS 15.3.1 de uma etapa seja 2015 (correspondendo aos dados da ESA).

- Torna a produtividade do Trends.Earth o conjunto de dados padrão na ferramenta de uma etapa do ODS para 15.3.1.

## 2.20 0,62 (27 de janeiro de 2019)

- Adiciona uma ferramenta experimental para mapear possíveis retornos de carbono de intervenções alternativas de restauração.
- Adiciona dados MODIS 2018.
- Correções diversas no dimensionamento de janelas para janelas da GUI.
- Atualiz para o openpyxl mais recente - corrige o carregamento do logotipo Trends.Earth nas tabelas de resumo.
- Adiciona a lista de publicações para ajudar os documentos.

## 2.21 0,60 (3 de dezembro de 2018)

- Acrescenta o cálculo da mudança na área urbana e na taxa de crescimento populacional (ODS 11.3.1)
- Corrige alturas padrão do botão/campo de entrada Adicionar seleção de cidade para AOI
- Adiciona buffer opcional da AOI

## 2.22 0,58 (11 de agosto de 2018)

- Adiciona uma seção de teste à página de cálculos
- Adiciona versão de teste do carbono total (acima e abaixo do solo) e das emissões devido ao desmatamento
- Pequenas correções de bugs, incluindo polígonos inválidos nas AOIs de entrada

## 2.23 0.56.5 (21 de maio de 2018)

- Corrige o erro com a importação de LPD solicitando um ano de dados.

## 2.24 0.56.4 (21 de maio de 2018)

- Sempre redefina os dados importados para a resolução mais alta.
- Corrige as zonas climáticas personalizadas de importação do SOC para usar um conjunto de dados expandido das zonas climáticas para eliminar nenhum dado.
- Atualiza MOD16A2 com os dados mais recentes.
- Forçar entrada da data na importação de dados SOC e LC
- Adicione saídas globais Trends.Earth à ferramenta de download.
- Corrija o manuseio de valores NULL nas legendas.

## 2.25 0.56.3 (21 de abril de 2018)

- Corrija o cálculo das tabelas de resumo das AOIs divididas no 180º meridiano (Fiji, Rússia etc.).
- Modifique o cálculo do estado para que áreas com mudanças de magnitude muito pequenas na integral NDVI ( $<0,01$  unidades NDVI durante todo o período) sejam consideradas estáveis.

## 2.26 0.56.2 (10 de abril de 2018)

- Pequenas correções unicode.

## 2.27 0.56.1 (10 de abril de 2018)

- Corrija o erro do marshmallow na carga do plugin

## 2.28 0,56 (9 de abril de 2018)

- Corrigir problema com rasterização de dados (rasters vazios na saída)
- Forçar o usuário a escolher a resolução de saída ao rasterizar um vetor
- Suporte ao cálculo da degradação do SOC a partir de dados SOC e LC personalizados

## 2.29 0,54 (8 de abril de 2018)

- Suporte ao carregamento de dados LPD, SOC e LC personalizados.
- Estilos de limpeza para que correspondam a maps.trends.earth
- Atualiza pyopenxl
- Adicione ícones de importação/carregamento a todas as caixas seletoras de camadas

## 2.30 0.52.1 (21 de março de 2018)

- Pequenas correções de bugs durante a oficina de Antalya.

### **2.31 0,52.1 (21 de março de 2018)**

- Pequenas correções de bugs durante a oficina de Antalya.

### **2.32 0,52 (19 de março de 2018)**

- Limpe o código de processamento da AOI.

### **2.33 0,50 (15 de março de 2018)**

- Passe a exceção se relacionado apenas à adição do logotipo Trends.Earth no arquivo do Excel.
- Várias pequenas correções de bugs.

### **2.34 0,48 (13 de março de 2018)**

- Corrigir formatação da tabela

### **2.35 0,46 (13 de março de 2018)**

- Suporte ao cálculo da tabela de relatórios com várias geometrias (Fiji, Rússia)
- Adicionar tabelas LPD e LC à guia da planilha UNCCD
- Limpe a mensagem de aviso na ferramenta de importação LPD
- Corrigir o carregamento final combinado da camada de produtividade da TE
- Corrigir tarefas de download (ainda sem estilos)

### **2.36 0,44 (12 de março de 2018)**

- Adicionar JRC LPD
- Adicionar ferramenta para fazer upload de dados personalizados de cobertura do solo
- Adicionar ferramenta para fazer upload de dados de produtividade personalizados
- Adicione que o upload personalizado do SOC será lançado em breve
- Ferramenta Adicionar para adicionar mapas base usando dados do Natural Earth
- Adicione a ferramenta tudo-em-um-passo para calcular os três subindicadores de uma só vez
- Renomeie a classe «Solo Exposto» para «Outra classe de cobertura» para obter consistência com a UNCCD
- Atualizar documentos
- Atualize para o marshmallow 3.0.0b7
- Mova o código GEE para o repositório principal trends.earth

- Melhore o manuseio de AOIs, principalmente quando os shapefiles são usados para entrada
- Manipule saídas de vários arquivos do GEE, agrupando-as em VRTs
- Suporte ao processamento de dados para países que cruzam o 180º meridiano
- Melhorar a formatação da tabela de resumo
- A partir de agora, as versões do script GEE serão correspondidas à versão do plugin

### 2.37 0,42 (4 de fevereiro de 2018)

- Corrigir falha na alteração da agregação de LC (devido setEnabled no rótulo removido)

### 2.38 0,40 (4 de fevereiro de 2018)

- Remova o uso do modo para o indicador de cobertura do solo.
- Combine a tabela de resumo e as ferramentas de criação de mapa de indicadores SDG.
- Adicione stub para onde o produto JRC LPD estará disponível.
- Salve o subindicador de produtividade como banda 2 no arquivo do indicador SDG.
- Bump script GEE para v0.3.
- Corrija o erro devido à divisão por zero na geração da tabela de resumo quando uma classe tiver área zero.
- O padrão é MODIS para cálculos de produtividade.

### 2.39 0,38 (16 de janeiro de 2018)

- Adicionar cálculo anual de carbono orgânico do solo
- Limpe o código de processamento da AOI, permita vários polígonos de entrada nas AOIs do shapefile
- Adicionar sombreamento ao lado dos itens da tabela de agregação da cobertura da terra
- Corrija o problema do firstShow na tabela de agregação
- Revise a saída da tabela de resumo para fornecer mais informações sobre cada um dos três indicadores
- Adicione conjuntos de dados suplementares ao desempenho, estado, cobertura do solo e produção de carbono orgânico do solo.
- Atualize nenhum dado e mascaramento para ser consistentemente -32768 (sem dados) e -32767 (dados mascarados)
- Permitir nomeação de downloads de arquivos
- Adicione ícone ao menu da barra de ferramentas, corrija o nome do plugin.
- Refatorar o código de estilo da camada para extrair informações da banda da saída GEE.
- Adicione uma ferramenta para carregar os conjuntos de dados trends.earth existentes no QGIS.
- Fixar limites de data de cobertura do solo - não permita que datas inválidas sejam selecionadas a partir dos dados da CCI.

**2.40 0,36 (14 de dezembro de 2017)**

- Corrija o problema com showEvent na ferramenta de criação de relatórios de mapas.

**2.41 0,34 (14 de dezembro de 2017)**

**2.42 0,32 (14 de dezembro de 2017)**

**2.43 0,30 (12 de dezembro de 2017)**

**2.44 0,24 (6 de dezembro de 2017)**

**2.45 0,22 (4 de dezembro de 2017)**

**2.46 0,18 (2 de dezembro de 2017)**

**2.47 0,16 (6 de novembro de 2017)**

**2.48 0,14 (25 de outubro de 2017)**

**2.49 0,12 (6 de outubro de 2017)**