



INSTITUTO  
FEDERAL  
Santa Catarina



**PROFEPT**  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

# OBSERVATÓRIO DA EXTENSÃO

Paulo Victor Rebouças Soares  
Igor Thiago Marques Mendonça  
Crislaine Gruber





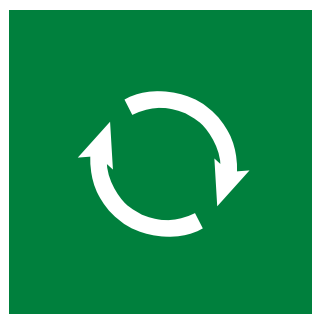
## A EXTENSÃO COMO PROCESSO EDUCATIVO CONSTRÓI PONTES ENTRE ESCOLA E SOCIEDADE



**EXTRAÇÃO**



**TRATAMENTO**



**CARREGAMENTO**



**EXIBIÇÃO**

### PRODUTO EDUCACIONAL

O Observatório da Extensão é um produto educacional em forma de *software* fruto da pesquisa intitulada Observatório da Extensão no Instituto Federal de Santa Catarina como solução para publicização das ações de extensão de interesse da sociedade e do instituto.

### PROGRAMA DE MESTRADO

A pesquisa foi desenvolvida no âmbito do programa em rede nacional de Mestrado Profissional em Educação Profissional e Tecnológica (ProfEPT) coordenado pelo Instituto Federal do Espírito Santo (IFES) e ofertado no câmpus Florianópolis do Instituto Federal de Santa Catarina (IFSC).

**ISBN n° 978-65-83787-04-0**



[48] 3877-9051  
[48] 3877-9000



paulo.soares@ifsc.edu.br  
<https://www.ifsc.edu.br/extensao>



**IFSC Câmpus Florianópolis**

Av. Mauro Ramos, 950 - Centro -  
Florianópolis, Santa Catarina

**Reitoria**

R. Quatorze de Julho, 150 -  
Coqueiros - Florianópolis, Santa  
Catarina

# SUMÁRIO

01 | O Produto

09 | Preparação

16 | Extração

22 | Tratamento

25 | Carregamento

30 | Exibição

34 | Considerações

Referências



# O PRODUTO

OBSERVATÓRIO DA EXTENSÃO

**É PRECISO REFLETIR SOBRE O QUE A  
INFORMAÇÃO SIGNIFICA EM UM CONTEXTO  
E SER CRÍTICO QUANTO À DESCOBERTA**

O PRODUTO EDUCACIONAL 'OBSERVATÓRIO DA EXTENSÃO' FOI VALIDADO NO DIA 17 DE DEZEMBRO DE 2024 POR MEIO DE UM GRUPO FOCAL COMPOSTO POR MEMBROS QUE ESTÃO EM CONSTANTE CONTATO COM A SOCIEDADE E ATUANDO JUNTO À GESTÃO DO IFSC: OUVIDORIA, GESTÃO DE DADOS, GESTÃO DO CONHECIMENTO, COORDENAÇÃO DE EXTENSÃO DE CÂMPUS E PRÓ-REITORIA DE EXTENSÃO.

ESTA SEÇÃO TRAZ UMA APRESENTAÇÃO DO PRODUTO EDUCACIONAL ENQUANTO AS DEMAIS ABORDAM ASPECTOS TÉCNICOS DE COMO IMPLEMENTÁ-LO.

# O Produto

O Observatório da Extensão tem como principal objetivo trazer mais visibilidade às ações de extensão no Instituto Federal de Santa Catarina (IFSC), permitindo uma apresentação clara e acessível dos dados para dois públicos-alvo: a sociedade civil interessada nas ações de extensão e os gestores responsáveis pela gestão da extensão, visando uma melhor tomada de decisão.

A solução em *software* é baseada no Apache Airflow, uma ferramenta robusta para orquestração de fluxos de processamento de dados, que extrai dados do Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA) e os processa para apresentação visual no Looker Studio.

Os dados são apresentados no Looker Studio de forma intuitiva por meio de gráficos, tabelas e visualizações interativas que filtram os dados por ano e câmpus. Essas visualizações permitem que tanto os gestores quanto a sociedade possam acompanhar e entender facilmente os dados da extensão.

O produto educacional foi avaliado por um grupo focal, no dia 17 de dezembro de 2024, composto por servidores do IFSC, que atuam diretamente na gestão das ações de extensão e estão em constante contato com a comunidade. Esse grupo contribuiu com observações valiosas que garantiram que o painel atende às necessidades dos gestores e à transparência necessária para o público externo e interno à instituição, aproximando esses das atividades de extensão.

O produto educacional tem um grande potencial de replicabilidade em outras áreas da Educação Profissional e Tecnológica (EPT) e pode ser facilmente adaptada por outras instituições de ensino. Esse encarte tem por propósito apresentar os passos básicos para replicação da solução, bem como fomentar reflexões para aprimoramento do Observatório da Extensão e incentivar integração de dados entre membros da Rede Federal de Educação Profissional, Científica e Tecnológica.

A concepção do produto educacional surgiu da necessidade de levantar indicadores que atendessem requisitos legais e anseios da sociedade por transparência. Como solução, o Observatório da Extensão possibilitou a publicização de dados por meio de uma arquitetura de *software*/aplicativo para exposição do panorama das atividades de extensão do Instituto Federal de Santa Catarina (IFSC),

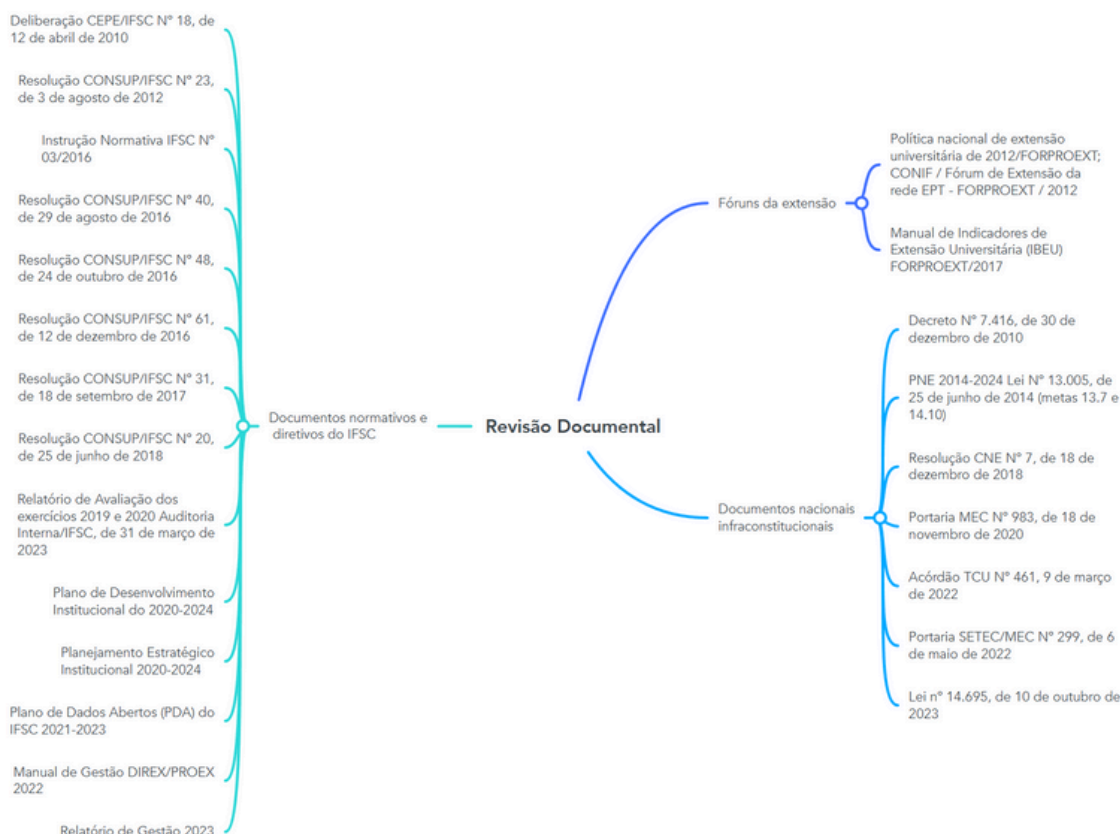
A gestão de dados tem se tornado fundamental para a formulação e o aprimoramento de políticas, bem como para as tomadas de decisão na gestão pública. No contexto brasileiro, as discussões na Rede Federal de Educação Profissional, Científica e Tecnológica abrangem entidades que dentro de seus fóruns tendem a aprimorar a gestão e concepção da extensão para então trazê-las aos câmpus. Dentre uma dessas, destaca-se o FORPROEXT que promoveu a padronização e sistematização de processos em ambientes computadorizados (Rodrigues, 2016; FORPROEXT, 2006).

A implementação de uma sistematização digital de armazenamento e análise de dados deve estar alinhada com os requisitos legais do país. Nesse âmbito destacam-se três leis: **(1)** a lei nº 12.527, de 18 de novembro de 2011, conhecida como lei de acesso à informação (LAI) que sob o princípio da publicidade e transparência estabelece procedimentos de divulgação de ações de órgãos e autarquias dos entes federativos; **(2)** a lei nº 12.965, de 23 de abril de 2014, o marco civil da internet, que estabelece garantias, direitos, deveres e princípios no uso da internet no Brasil, também estabelece diretrizes de governança institucional e pessoal de dados, além de tipificar a soberania dos dados nacionais; **(3)** a lei nº 13.709, de 14 de agosto de 2018, ou lei geral de proteção de dados pessoais (LGPD), que dispõe sobre o tratamento de dados pessoais em qualquer meio, em especial no meio digital, para a proteção do direito de liberdade e privacidade.

No desenvolvimento do Observatório da Extensão também foram considerados documentos normativos e diretivos do IFSC. **(1)** A resolução CGD/IFSC nº 03, de 9 de abril de 2021 que aprova o plano de dados abertos; **(2)** A resolução CGD/IFSC nº 08, de 5 de dezembro de 2022 que aprova a política de segurança da informação. Todavia, esses documentos dão apenas uma direção no tratamento dos dados, para compreender quais dados e informação se requer publicizar foi preciso realizar um levantamento de indicadores.

## INDICADORES OBSERVÁVEIS

Para um levantamento mais acurado das necessidades da sociedade, a pesquisa mapeou os indicadores por meio de três abordagens que se complementam. A primeira deu enfoque na revisão documental oriundos de fóruns da extensão, de documentos nacionais infraconstitucionais, e de normativos e diretivos do IFSC. Já nessa primeira abordagem, por meio da análise de conteúdo, foi possível traçar características da extensão.



Link do mapa: <https://mm.tt/app/map/3608020948?t=aOpCQkbHUB>

A segunda abordagem considerou o que a sociedade externa ao IFSC solicita sobre acesso à informação através do canal oficial de manifestação do governo federal, portal [FalaBR](#) da Controladoria-Geral da União. Esse acesso é exclusivo da ouvidoria do órgão ou autarquia, e de visibilidade restrita à instituição, o que facilita a curadoria dos dados classificados como ‘acesso à informação’.

Não menos importante, existe o buscador com histórico abrangente de solicitações de acesso à informação, o [BuscaLAI](#) do Tribunal de Contas da União, esse é de acesso livre e há uma massa de dados a ser tratada. Como facilitadoras foram utilizadas bibliotecas da linguagem Python para curadoria dos dados. Detalhes do script de programação são encontrados no capítulo ‘Extração’.

A terceira abordagem foi trabalhada com entrevistas semiestruturadas direcionadas aos gestores da extensão da reitoria e dos câmpus a fim de compreender e levantar quais indicadores podem contribuir para a tomada de decisão e planejamento na gestão, e com base na experiência desses com a comunidade quais indicadores podem aproximar o público externo. Para a pesquisa foi retirada uma amostra de 7 entrevistados, sendo 1 de cada região do estado e 1 da reitoria.



Conforme os modelos de gestão e organização de processos vão se aprimorando, conseqüentemente novos indicadores podem surgir. Sendo assim, realizar essas três abordagens ciclicamente ao revisar documentações existentes e analisar novas, curar novas solicitações de acesso à informação e conversar com os gestores de outros câmpus é uma boa prática para aprimorar a visibilidade dos dados da extensão. Todavia, o primeiro ciclo proveniente da pesquisa dá o alicerce para a continuidade da solução. Os indicadores da extensão foram classificados em 13 categorias ou macroindicadores, conforme a ilustração a seguir.



Link do mapa com detalhes: <https://mm.tt/app/map/3607149861?t=21E73Tfifn>

Um produto educacional para ser considerado utilizável precisa ser aprovado, para isso o Observatório da Extensão foi apresentado a um grupo focal composto por servidores do IFSC da ouvidoria, da pró-reitoria de extensão e relações externas, da coordenação de extensão de câmpus, da coordenadoria de gestão de dados, além de convidar a diretoria de gestão do conhecimento. Esses que lidam em sua rotina com dados e informações fizeram uma avaliação positiva do produto, acrescentando apenas 1 indicador que não havia sido identificado. Segue abaixo a imagem do painel principal do Observatório da Extensão inspirado no Acórdão TCU N° 461, 9 de março de 2022.

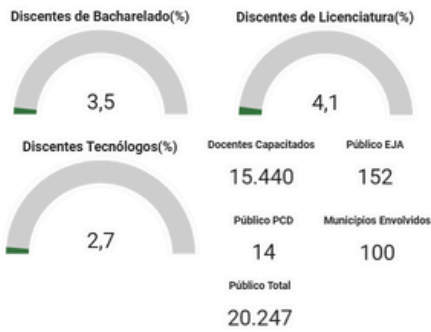




O Painel Glossário de Termos

Ano: 2024 (1)

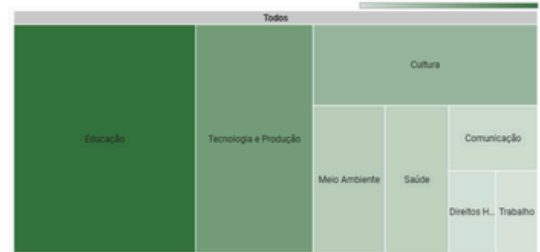
Envolvimento na Extensão



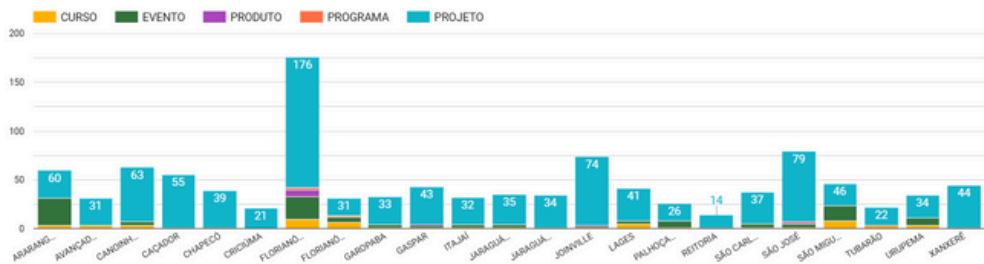
Orçamento Aplicado na Extensão



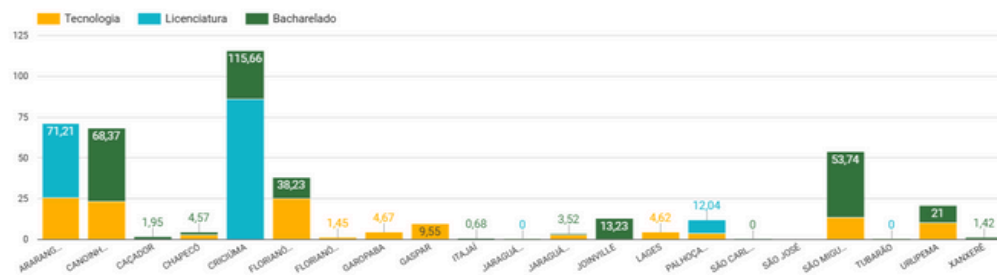
Áreas Temáticas da Extensão



Ações de Extensão por Câmpus e Modalidade



Proporção de Público Externo Envolvido por Discente de Graduação por Câmpus e Tipo de Curso



Data da última atualização: 04/03/2025 10:12:35 | Política de Privacidade

O Produto



Link do Observatório da Extensão:  
<https://lookerstudio.google.com/reporting/2ba39b66-7d69-4fb6-8851-a2a2d8c9c460>

Ao todo foram levantados 114 indicadores dos quais 100 foram possíveis de extrair no todo ou em parte da base de dados do sistema integrado de gestão de atividades acadêmicas (SIGAA) do IFSC. Para atender os 14 indicadores restantes, é necessário que se ocorram customizações no SIGAA. Os indicadores foram apresentados no Observatório da Extensão divididos em 85 painéis navegáveis de acordo com as categorias, e considerando os documentos revisados, cada painel dispõe de filtros para visualização dos dados por câmpus e/ou ano.

Para implementação da arquitetura de *software* foram escritas 5.565 linhas de código para extração e tratamento dos dados, sem considerar a edição de linhas de código e configuração de comandos para funcionamento do ambiente da solução. A reprodução da mesma arquitetura será abordada nos capítulos seguintes.

# PREPARAÇÃO

OBSERVATÓRIO DA EXTENSÃO

**UM AMBIENTE DE SISTEMA DE *SOFTWARE* É  
CONSTITUÍDO POR COMPONENTES QUE  
TENHAM COMPORTAMENTOS E RELAÇÕES  
BEM DEFINIDOS ENTRE SI**

A PREPARAÇÃO É A BASE DA ARQUITETURA DE SOFTWARE, NESSE MOMENTO SÃO TESTADAS AS FUNCIONALIDADES QUE VÃO GARANTIR QUE O SOFTWARE FUNCIONE DE FORMA EFICIENTE E CONFIÁVEL, ALÉM DISSO É FUNDAMENTAL GARANTIR A SEGURANÇA E PROTEÇÃO DOS DADOS.

ESTA SEÇÃO TRAZ CONFIGURAÇÕES PARA REPRODUÇÃO DA ARQUITETURA DO AMBIENTE DE HOSPEDAGEM DO *SOFTWARE*.

O ambiente de *software* foi montado sob o sistema operacional Linux Ubuntu versão 22.04.5 LTS em um mini computador com a seguinte configuração:

- 8 GB de memória principal
- 250 GB de memória secundária
- Processador Intel Core i7-8700T com *clock* de 2.40 GHz
- Placa de vídeo Intel UHD Graphics 630

Foram utilizadas as tecnologias abaixo e suas respectivas versões:

- Docker 27.5.1
- Docker Compose 1.29.2
- Apache Airflow 2.10.3
- PostgreSQL 9.0.13
- DBeaver 24.3.4
- Visual Studio Code 1.96.4
- Python 3.10.12
- PyPI 22.0.2
- Google Chrome 133.0.6943.53 para acesso ao Google Drive, Looker Studio e Google Colaboratory

Para extração e análise de dados foi realizada uma réplica da base de dados do SIGAA com o PostgreSQL em uma máquina virtual mantida pelo IFSC. Por sua vez, para acessá-la disponibilizaram credenciais de acesso via DBeaver. Esse programa é uma ferramenta de gerenciamento de banco de dados, nele é possível configurar múltiplas conexões a outros tipos de banco de dados, bem como visualizar seus relacionamentos e salvar scripts de consulta e edição.

Embora o Ubuntu seja um sistema operacional leve, ainda assim optou-se na construção do produto educacional adotar a solução Docker que tem por alvo compactar e isolar um *software* em um container utilizando-se do mínimo necessário do sistema operacional para possibilitar a replicabilidade da solução em outros ambientes.

No entanto apenas o Docker não traz praticidade para o ambiente, no caso de *softwares* como o Apache Airflow há uma gama de instruções, outros containers e configurações a serem passadas em uma linha de comando no terminal para funcionamento dela. Para isso foi provido o Docker Compose, que permite vários containers e configurações pré-definidas em um arquivo `.yaml`

```
x-airflow-common:
  &airflow-common
  build: /opt/Airflow
  environment:
    &airflow-common-env
    AIRFLOW__CORE__EXECUTOR: CeleryExecutor
    AIRFLOW__DATABASE__SQL_ALCHEMY_CONN:
    postgresql+psycopg2://airflow:airflow@postgres/airflow
    AIRFLOW__CELERY__RESULT_BACKEND:
    db+postgresql://airflow:airflow@postgres/airflow
    AIRFLOW__CELERY__BROKER_URL:
    redis://:@redis:6379/0
    AIRFLOW__CORE__FERNET_KEY: ""
    AIRFLOW__CORE__DAGS_ARE_PAUSED_AT_CREATION:
    'true'
    AIRFLOW__CORE__LOAD_EXAMPLES: 'false'
    AIRFLOW__API__AUTH_BACKENDS:
    'airflow.api.auth.backend.basic_auth,airflow.api.auth.backe
    nd.session'
    logging-monitoring/check-health.html#scheduler-health-
    check-server
    AIRFLOW__SCHEDULER__ENABLE_HEALTH_CHECK:
    'true'
    _PIP_ADDITIONAL_REQUIREMENTS:
    ${_PIP_ADDITIONAL_REQUIREMENTS:-}
  file
    #AIRFLOW_CONFIG: '/opt/Airflow/config/airflow.cfg'
  volumes:
    - ${AIRFLOW_PROJ_DIR:-.}/dags:/opt/airflow/dags
    - ${AIRFLOW_PROJ_DIR:-.}/logs:/opt/airflow/logs
    - ${AIRFLOW_PROJ_DIR:-.}/plugins:/opt/airflow/plugins
    - ${AIRFLOW_PROJ_DIR:-.}/config:/opt/airflow/config
    - ${AIRFLOW_PROJ_DIR:-.}/config/airflow.cfg:/opt/
    airflow/config/airflow.cfg
    - /opt/Airflow:/opt/Airflow
  user: "${AIRFLOW_UID:-50000}:0"
  depends_on:
    &airflow-common-depends-on
    redis:
      condition: service_healthy
    postgres:
      condition: service_healthy
```

# Preparação

```
services:
  postgres:
    image: postgres:13
    environment:
      POSTGRES_USER: airflow
      POSTGRES_PASSWORD: airflow
      POSTGRES_DB: airflow
    volumes:
      - postgres-db-volume:/var/lib/postgresql/data
    healthcheck:
      test: ["CMD", "pg_isready", "-U", "airflow"]
      interval: 10s
      retries: 5
      start_period: 5s
      restart: always

  redis:
    # Redis is limited to 7.2-bookworm due to licencing
    change
    # https://redis.io/blog/redis-adopts-dual-source-available-licensing/
    image: redis:7.2-bookworm
    expose:
      - 6379
    healthcheck:
      test: ["CMD", "redis-cli", "ping"]
      interval: 10s
      timeout: 30s
      retries: 50
      start_period: 30s
      restart: always

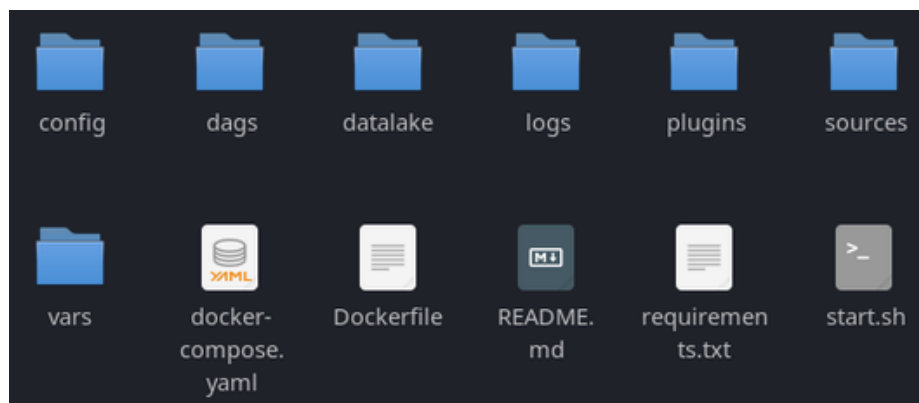
  airflow-webserver:
    <<: *airflow-common
    command: webserver
    ports:
      - "8086:8080"
    healthcheck:
      test: ["CMD", "curl", "--fail", "http://localhost:8086/health"]
      interval: 30s
      timeout: 10s
      retries: 5
      start_period: 30s
      restart: always
    depends_on:
      <<: *airflow-common-depends-on
  airflow-init:
    condition: service_completed_successfully
```

...



O arquivo acima é apenas uma segmentação do arquivo `docker-compose.yaml`, todavia observa-se a diversidade de serviços e suas configurações requeridos para a orquestração do Apache Airflow. Para consultar o arquivo completo acesse o link [https://github.com/pvreboucas/airflow/blob/main/original\\_docker-compose.yaml](https://github.com/pvreboucas/airflow/blob/main/original_docker-compose.yaml)

Em um primeiro momento somente a execução do arquivo não funcionará, antes é preciso criar uma estrutura de diretórios requeridos nas configurações: `dags`, `logs`, `config`, `plugins`. Ainda para esse produto educacional foram criados os diretórios `sources` para demais recursos e `datalake` para guardar os dados tratados. Opcionalmente, ainda há o diretório `vars` para variáveis de ambiente do sistema operacional e `backup` das variáveis da aplicação.



Uma vez criados os diretórios é necessário dar atenção aos arquivos `Dockerfile`, `requirements.txt` e `start.sh`. No `Dockerfile` existirão apenas 3 linhas de comando, a primeira informando qual será a versão do Apache Airflow a ser utilizada, a segunda copia o arquivo `requirements.txt` para dentro do container e a terceira manda executar a instalação de bibliotecas listadas.

```
FROM apache/airflow:2.10.3
COPY requirements.txt /
RUN pip install --no-cache-dir "apache-airflow==${AIRFLOW_VERSION}" -r
/requirements.txt
```

O arquivo requirements.txt é utilizado pelo PyPI, uma ferramenta de índice de pacotes de instalação da linguagem Python, com a lista de bibliotecas extras a serem utilizadas dentro do container. A lista de bibliotecas indexadas encontram-se em <https://pypi.org/>. Para essa o produto educacional foram listadas:

```
pyspark==3.5.3
pendulum==3.0.0
pandas==2.1.4
numpy==1.26.4
requests==2.32.3
google-generativeai==0.8.3
apache-airflow-providers-google==10.25.0
apache-airflow-providers-postgres==5.13.1
```

Por conseguinte, é preciso designar um UID para o Airflow, em seguida executar o arquivo start.sh que contém o comando de início da orquestração do serviço indicando a associação do id de usuário que irá executá-lo ao usuário raiz do container, e condicionando a subir o flower (interface de *workers* das operações das tarefas).

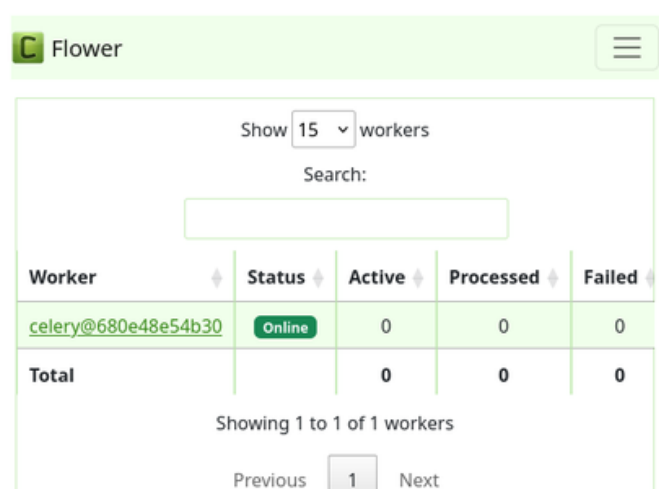
```
AIRFLOW_UID=50000
AIRFLOW_UID=$(id -u) docker-compose --profile flower up -d --build
```

Para conferir se funcionou executa-se o comando:

```

CREATED          STATUS          PORTS          NAMES
5 weeks ago      Up 8 hours (unhealthy)  0.0.0.0:8086->8080/tcp, [::]:8086->8080/tcp  airflow_airflow-webserver_1
5 weeks ago      Up 8 hours (healthy)   8080/tcp      airflow_airflow-worker_1
5 weeks ago      Up 8 hours (healthy)   8080/tcp      airflow_airflow-scheduler_1
5 weeks ago      Up 8 hours (healthy)   0.0.0.0:5555->5555/tcp, [::]:5555->5555/tcp, 8080/tcp  airflow_flower_1
5 weeks ago      Up 8 hours (healthy)   8080/tcp      airflow_airflow-triggerer_1
5 weeks ago      Up 8 hours (healthy)   5432/tcp      airflow_postgres_1
5 weeks ago      Up 8 hours (healthy)   6379/tcp      airflow_redis_1
    
```

Serão listados quais containers estão funcionando, e para acessar o Apache Airflow acesse a uri configurada no docker-compose.yaml - <http://localhost:8080> ou <http://localhost:8086> -. Para consultar o flower acesse <http://localhost:5555>.



Para mais configurações do flower edite/adicione as conexões na interface do Apache Airflow em Admin -> Pool, declare as variáveis abaixo no arquivo airflow.cfg dentro do diretório config, edite a variável AIRFLOW\_CONFIG no arquivo docker-compose.yaml, consulte a documentação da ferramenta e reinicie os containers.

```
[core]

#determina o número máximo de tarefas que podem ser agendadas/executadas
de forma simultânea em um mesmo DAG;
max_active_tasks_per_dag = 4

#determina o número máximo de execuções de DAG (DAG runs) ativas por DAG
que o Scheduler pode criar simultaneamente;
max_active_runs_per_dag = 4

#determina quantas tarefas cada worker do Celery pode executar
simultaneamente.
worker_concurrency = 3
```

Por fim, no primeiro acesso ao Apache Airflow as credenciais por padrão são 'airflow' para login e senha, cuja modificação será solicitada no primeiro acesso. Depois, a página inicial trará a lista de DAGs criadas no diretório dags via Visual Studio Code. O conceito de DAGs é abordado no capítulo de 'Extração'. Para as demais ferramentas é necessário que se crie/utilize uma conta google acessível pelo navegador Chrome.

DAGS

DAG	Owner	Runs	Schedule	Last Run	Next Run
IFSC_EXTENSAO_SIGAA_AREAS_TEMATICAS	airflow	0/0/0/0	@daily		2025-02-11, 00:00:00
IFSC_EXTENSAO_SIGAA_ATIVIDADES	airflow	1/1/0/0	@daily	2025-01-21, 00:00:00	2025-02-11, 00:00:00
IFSC_EXTENSAO_SIGAA_CURRICULARIZACAO	airflow	0/0/0/0	@daily		2025-02-11, 00:00:00
IFSC_EXTENSAO_SIGAA_DISCENTES	airflow	0/0/0/0	@daily		2025-02-11, 00:00:00
IFSC_EXTENSAO_SIGAA_DOCENTES	airflow	0/0/0/0	@daily		2025-02-11, 00:00:00
IFSC_EXTENSAO_SIGAA_INFRAESTRUTURA_RH	airflow	0/0/0/0	@daily		2025-02-11, 00:00:00
IFSC_EXTENSAO_SIGAA_MODALIDADES	airflow	0/0/0/0	@daily		2025-02-11, 00:00:00
IFSC_EXTENSAO_SIGAA_ORCAMENTO	airflow	0/0/0/0	@daily		2025-02-11, 00:00:00
IFSC_EXTENSAO_SIGAA_PARCERIAS	airflow	0/0/0/0	@daily		2025-02-11, 00:00:00

# EXTRAÇÃO

OBSERVATÓRIO DA EXTENSÃO

**OS DADOS SÃO O INSUMO DA ERA DIGITAL,  
E A EXTRAÇÃO É O PROCESSO DE  
ENCONTRAR SUA VERDADEIRA FORMA.**

A EXTRAÇÃO DE DADOS NÃO É UM PROCESSO TRIVIAL, SEMPRE HÁ INTENCIONALIDADE NO ATO DE BUSCAR ALGO. NESSA JORNADA ENCONTRAM-SE PADRÕES, TENDÊNCIAS, OPORTUNIDADES E APRIMORAMENTO CONTÍNUO DE SERVIÇOS DE SISTEMAS DA INFORMAÇÃO.

ESTA SEÇÃO APRESENTA OS CÓDIGOS USADOS NA EXTRAÇÃO DE DADOS DAS PLATAFORMAS FALABR E BUSCALAI, BEM COMO MAPEAR SCRIPTS PARA EXTRAÇÃO DE BASE DE DADOS COM AIRFLOW.

No desenvolvimento deste produto educacional a extração se fez em dois momentos, no levantamento dos indicadores de extensão e na consulta desses na base de dados do SIGAA.

Quanto ao primeiro momento, foi realizada uma consulta na interface gráfica, sem automações, na plataforma do [FalaBR](#) de forma manual, sem intervenção de scripts. Porém, na consulta da [BuscaLai](#) foram baixados arquivos em formato .csv dos anos de 2020 a 2024 para automação por script. Para a construção dos scripts há a opção de utilizar a ferramenta Google Colaboratory no navegador ou o Visual Studio Code localmente. Se a opção for fazer isso localmente, deve-se instalar bibliotecas essenciais para o sistema operacional ubuntu e o python3 com suas respectivas bibliotecas.

```
sudo apt install build-essential gcc make perl dkms curl tcl
sudo apt install python3-pip -y | pip install pandas | pip install openpyxl
```

Para a seleção dos pedidos de informação direcionados ao IFSC foram considerados os arquivos Pedido\_csv\_[2020-2024].csv. Esses arquivos foram condensados em uma estrutura de dados chamada *dataframe* selecionando apenas as linhas onde a coluna OrgaoDestinatario corresponde ao IFSC - Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina e transformado em outro arquivo Pedidos\_IFSC\_2020\_2024.csv.

```
import pandas as pd
import numpy as np
arquivos = ['Pedidos_csv_2020.csv','Pedidos_csv_2021.csv','Pedidos_csv_2022.csv',
'Pedidos_csv_2023.csv','Pedidos_csv_2024.csv']
dataframes = []
for arquivo in arquivos:
    df = pd.read_csv(arquivo, encoding='utf-16', sep=';')
    df['OrgaoDestinatario'] = df['OrgaoDestinatario'].fillna("")
    df_filtrado = df[df['OrgaoDestinatario'].str.contains('IFSC - Instituto Federal
de Educação, Ciência e Tecnologia de Santa Catarina')]
    dataframes.append(df_filtrado)
df_final = pd.concat(dataframes)
df_final.to_csv('Pedidos_IFSC_2020_2024.csv', index=False, encoding='utf-8',
sep=';')
```

Com o arquivo Pedidos\_IFSC\_2020\_2024.csv gerado é possível realizar um rastreamento da palavra extensão ou extensao, em caixa baixa, nas células do *dataframe* de acordo com as colunas selecionadas e então criar o arquivo Pedidos\_Extensao\_IFSC\_2020\_2024.csv.

```
import pandas as pd
df = pd.read_csv("Pedidos_IFSC_2020_2024.csv", encoding='utf-8', sep=';')
colunas = ['DetalhamentoSolicitacao', 'FormaResposta', 'AssuntoPedido',
'SubAssuntoPedido', 'Tag', 'Resposta', 'Decisao', 'EspecificacaoDecisao']
mascara = df[colunas].apply(lambda x:
x.str.lower().str.contains('extensão|extensao')).any(axis=1)
df_filtrado = df[mascara]
df_filtrado.to_csv('Pedidos_Extensao_IFSC_2020_2024.csv', index=False,
encoding='utf-8', sep=';')
```

Por último executa-se o *script* abaixo para transformar o formato do arquivo .csv em .xlsx, assim o conteúdo fica legível no formato de planilha para análise humana.

```
import pandas as pd
arquivo_csv = 'Pedidos_Extensao_IFSC_2020_2024.csv'
arquivo_xlsx = 'Pedidos_Analisados_Extensao_IFSC_2020_2024.xlsx'
df = pd.read_csv(arquivo_csv, sep=';')
df.to_excel(arquivo_xlsx, index=False)
```

Quanto ao segundo momento, usou-se o DBeaver para consultar a base de dados. Como o banco de dados do SIGAA é do tipo relacional, utilizam-se *scripts* SQL baseado na teoria dos conjuntos, portanto a consulta de um indicador pode recorrer a uma ou mais tabelas relacionadas entre si. Como exemplo serão considerados os indicadores de produção extensionista:

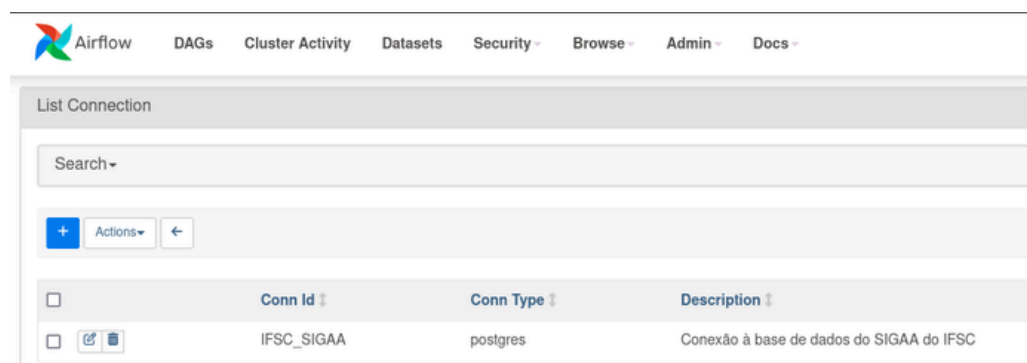
```
select ci.nome as campus, endereco.cep, proj.ano, rae.resultados_qualitativos,
rae.resultados_quantitativos
from extensao.atividade a
inner join extensao.relatorio_acao_extensao rae on rae.id_atividade =
a.id_atividade
inner join projetos.projeto proj on proj.id_projeto = a.id_projeto
inner join comum.unidade u on u.id_unidade = proj.id_unidade
inner join comum.campus_ies ci on ci.id_campus = u.id_campus_ies
inner join comum.endereco endereco on endereco.id_endereco = ci.id_endereco
where rae.resultados_qualitativos != ''
or rae.resultados_quantitativos != ''
group by ci.nome, endereco.cep, proj.ano, rae.resultados_qualitativos,
rae.resultados_quantitativos
order by ci.nome, proj.ano asc;
```

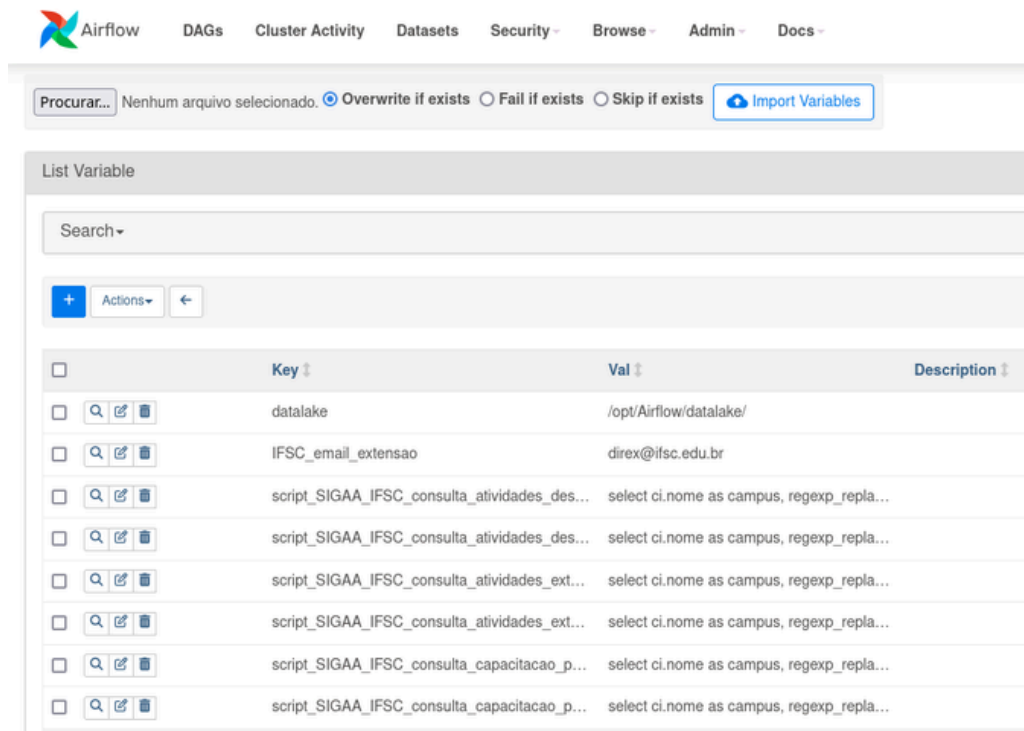
No *script* anterior os dados de interesse para o indicador são extraídos de 4 tabelas diferentes pertencentes a 3 conjuntos ou *schemas* de tabelas distintos. Assim como em outras extrações, para alcançar a tabela de endereços, é preciso agregar a interseção de 2 tabelas extras para encontrar o CEP de campus, totalizando a interação de 6 tabelas no *script* para atender um indicador. Logo, a extração de indicadores da base de dados não é algo simples de realizar sem despendar horas de análise.

Uma vez compreendido como estão organizados os dados deve-se construir os *scripts* SQL para os demais 113 indicadores para, posteriormente, mapeá-los no Apache Airflow. Prosseguir dessa forma permite que a manutenção e revisão dos *scripts* sejam facilitadas quando há mais pessoas envolvidas na gestão de dados.

Para comunicação com o banco de dados o Apache Airflow fornece uma interface de configuração encontrada em Admin -> Connections. Nela adiciona-se uma conexão do tipo postgres com o endereço, porta e autenticação da base de dados do SIGAA.

Para o mapeamento dos *scripts* SQL segue-se para Admin -> Variables. Nessa área é possível criar variáveis para outros propósitos. No que tange aos *scripts*, considerando a escalabilidade da solução, adotou-se a nomenclatura `script_SIGAA_IFSC_*`, assim outras instuições com outros bancos de dados podem ser agregadas.





O Apache Airflow é um orquestrador de fluxos de trabalho de tarefas chamados de *pipelines* que podem ser agendados e/ou programados para executar dentro de rotinas. Os *pipelines* são organizados dentro de DAGs, inspiradas na teoria dos grafos, o acrônimo significa *Directed Acyclic Graph*, em português Grafo Acíclico Direcionado, em outras palavras, o processamento das DAGs é independente umas das outras, com suas tarefas interligadas na forma de grafos com execução estruturada.

Para construir a primeira DAG recorre-se ao Visual Studio Code em Arquivo -> Abrir Pasta e seleciona-se o diretório dags. Nele cria-se um arquivo .py indicando ser da linguagem Python, tendo ela como base utilizam-se bibliotecas do Apache Airflow para estruturar as DAGs.

No exemplo a seguir declara-se uma DAG com o identificador `IFSC_EXTENSAO_SIGAA_PRODUCOES` seguindo a mesma lógica de nomenclatura dos *scripts* à base de dados; Iniciando-se imediatamente a execução com uma frequência diária.

```
from typing import Optional
from airflow import DAG
from airflow.operators.python import PythonOperator
from airflow.providers.postgres.operators.postgres import PostgresOperator
from airflow.providers.smtp.operators.smtp import EmailOperator
from airflow.models import Variable
from airflow.utils.dates import days_ago
from datetime import datetime
import pandas as pd
import os

with DAG(
    dag_id='IFSC_EXTENSAO_SIGAA_PRODUCOES',
    start_date=days_ago(1),
    schedule_interval='@daily',
    catchup=False,
) as dag:

    extrai_resultados_relatorio_extensao = PostgresOperator(
        task_id='extrai_resultados_relatorio_extensao',
        postgres_conn_id='IFSC_SIGAA',
        sql = Variable.get("script_SIGAA_IFSC_consulta_resultados_relatorio_extensao"),
    )

    extrai_resultados_relatorio_extensao
```

Destaca-se ainda a construção da tarefa de identificação `extrai_resultados_relatorio_extensao` utilizando a conexão `IFSC_SIGAA` e o `script` declarado na variável `script_SIGAA_IFSC_consulta_resultados_relatorio_extensao` com a sua chamada de execução presente na última linha. Dessa forma conclui-se o cliço de extração de um indicador.

Ao final do mapeamento dos scripts de extração do banco de dados em variáveis é recomendado que se faça uma exportação desses, pois caso os containers do Apache Airflow sejam apagados não há como recuperar as variáveis mapeadas, para tanto se faz necessário importá-las novamente através do arquivo.

Para mais exemplos de utilização do Apache Airflow acesse o link: <https://github.com/pvreboucas/airflow>



# TRATAMENTO

OBSERVATÓRIO DA EXTENSÃO

**TRANSFORMAR DADOS BRUTOS EM  
CONHECIMENTO É O PRIMEIRO PASSO  
PARA TOMAR DECISÕES ASSERTIVAS E  
ESTRATÉGICAS**

TRATAR OU TRANSFORMAR DADOS NÃO É APENAS UMA TÉCNICA, É UM PROCESSO QUE TRANSMUTA CAOS EM CLAREZA, REQUER DO TRATADOR REFLETIR O QUE É ESSENCIAL NA TRANSFORMAÇÃO DE UM DADO BRUTO EM INFORMAÇÃO CLARA.

ESTA SEÇÃO TRAZ DECISÕES TOMADAS NA CONSTRUÇÃO DE SCRIPTS E DE FILTROS PARA O TRATAMENTO DOS DADOS.

O tratamento de dados não é uma fase bem definida, esse processo pode ocorrer concomitante com a extração e a exibição, a exemplo no *script* SQL de extração da produção extensionista pode-se inserir uma função para retornar somente os números do CEP. Da mesma forma, na exibição com o Looker Studio, para evitar distorções, é possível filtrar o escopo de anos de existência da EPT.

```
select ci.nome as campus, regexp_replace(endereco.cep,'[^0-9]', '', 'g') as cep,
proj.ano, rae.resultados_qualitativos, rae.resultados_quantitativos
from extensao.atividade a
inner join extensao.relatorio_acao_extensao rae on rae.id_atividade =
a.id_atividade
inner join projetos.projeto proj on proj.id_projeto = a.id_projeto
inner join comum.unidade u on u.id_unidade = proj.id_unidade
inner join comum.campus_ies ci on ci.id_campus = u.id_campus_ies
inner join comum.endereco endereco on endereco.id_endereco = ci.id_endereco
where rae.resultados_qualitativos != ''
or rae.resultados_quantitativos != ''
group by ci.nome, endereco.cep, proj.ano, rae.resultados_qualitativos,
rae.resultados_quantitativos
order by ci.nome, proj.ano asc;
```

Nome: Filtro de Ano

QUANTITATIVO\_PRODUCOES\_EXTENSI...  Mostrar valores sugeridos ao digitar

Excluir 123 Ano Maior que (>) 2030

AND

Excluir 123 Ano Menor que (<) 1900

AND

Todavia, o tratamento de dados mais robusto se dá nos scripts das DAGs. No exemplo abaixo a tarefa `transforma_quantitativo_producoes_extensao` chama a função `classifica_producoes_de_relatorios_extensao` que faz uma varredura no *dataframe* e o compara a um dicionário que categoriza o tipo de produção de acordo com as palavras mapeadas encontradas nas colunas `resultados_qualitativos` e `resultados_quantitativos`. Para estruturar a execução sequencialmente, na última linha nota-se a função de transformação invocada após a de extração.

```

...

extrai_resultados_relatorio_extensao = PostgresOperator(
    task_id='extrai_resultados_relatorio_extensao',
    postgres_conn_id='IFSC_SIGAA',
sql = Variable.get("script_SIGAA_IFSC_consulta_resultados_relatorio_extensao"),
)

def categorizar_produto(texto, palavras_chave) -> Optional[str]:
    if pd.isnull(texto):
        return None
    if pd.isna(texto):
        return None
    texto = texto.lower()
    for categoria, palavras in palavras_chave.items():
        if any(palavra in texto for palavra in palavras):
            return categoria
    return None

def classifica_producoes_de_relatorios_extensao(ti) -> pd.DataFrame:
    result_resultados_relatorio_extensao =
ti.xcom_pull(task_ids='extrai_resultados_relatorio_extensao')
    df_relatorio = pd.DataFrame(result_resultados_relatorio_extensao,
columns=['campus','cep','ano','resultados_qualitativos','resultados_quantitativos']
)
    palavras_chave = {
        "Revistas e Periódicos": ["revista", "periódico", "journal"],
        "Livros ou Capítulos": ["livro", "capítulo", "book", "chapter"],
        "Artigos": ["artigo", "paper", "article"],
        "Mídias de Comunicações em Eventos":
["evento","comunicação","comunicacao"],
        "Produções Audiovisuais": ["audiovisual"],
        "Produções Artísticas": ["artística","arte"]
    }
    df_relatorio['tipo_produto_qualitativo'] = df_relatorio.apply(lambda x:
categorizar_produto(x['resultados_qualitativos'], palavras_chave), axis=1)
    df_relatorio['tipo_produto_quantitativo'] = df_relatorio.apply(lambda x:
categorizar_produto(x['resultados_quantitativos'], palavras_chave), axis=1)
    df_relatorio["tipo_produto"] =
df_relatorio["tipo_produto_qualitativo"].combine_first(df_relatorio["tipo_produto
_quantitativo"])
    df_quantitativo_producoes = df_relatorio.groupby(['campus', 'cep', 'ano',
'tipo_produto']).size().reset_index(name='quantidade')
    return df_quantitativo_producoes

transforma_quantitativo_producoes_extensao = PythonOperator(
    task_id='transforma_quantitativo_producoes_extensao',
    python_callable=classifica_producoes_de_relatorios_extensao
)

extrai_resultados_relatorio_extensao >>
transforma_quantitativo_producoes_extensao

```

# CARREGAMENTO

OBSERVATÓRIO DA EXTENSÃO

**CARREGAR DADOS NÃO É APENAS  
TRANSFERIR INFORMAÇÕES, É GARANTIR  
QUE O CONHECIMENTO FLUA PARA ONDE  
ELE É MAIS NECESSÁRIO**

A CARGA DE DADOS EFICIENTE GARANTE QUE OS DADOS TRATADOS ESTEJAM LEGÍVEIS PARA AS ENGRENAGENS DO SISTEMA QUE IRÃO APRESENTÁ-LOS. O SUCESSO DA CARGA É MAIS DO QUE ENVIAR INFORMAÇÕES DE UM PONTO A OUTRO, É GARANTIR QUE CHEGUEM DE FORMA CORRETA E SEM DISTORÇÕES.

ESTA SEÇÃO TRAZ A ESTRATÉGIA ADOTADA PARA O CARREGAMENTO DE DADOS NA CONSTRUÇÃO DO PRODUTO EDUCACIONAL.

O carregamento dos dados pode ocorrer de diversas formas, por exemplo, é possível carregar dados tratados em outro banco de dados, inclusive de outros tipos e estruturas. Também são possíveis gerar um arquivo do tipo chave-valor, um arquivo tabulado, além de utilizar-se de requisições http para enviar a outro *software* via *world wide web*.

Na elaboração do Observatório da Extensão foram buscadas maneiras de automatizar o carregamento entre Apache Airflow e Looker Studio enviando os dados para a solução Big Query (uma base de dados da plataforma Google), porém devido a esta solução não ser gratuita, ela foi descartada. No entanto, ainda estão declaradas as bibliotecas necessárias para esse carregamento no arquivo requirements.txt.

Outra abordagem estudada foi a de utilização de *plugins* que fazem a conexão direta entre o Looker Studio e o banco de dados PostgreSQL. Todavia, tal abordagem expõe para a plataforma contratada credenciais de acesso à base de dados, além de reduzir o potencial de escalabilidade de solução e de processamento de dados que o Apache Airflow oferece. Além disso, limita a possibilidade de construir *scripts* Python mais robustos para a transformação de dados, como é o caso do indicador de produção extensionista.

Portanto, foi adotada a abordagem de carregar os dados em um arquivo tabulado .csv e armazená-los no diretório datalake, para em seguida transferi-los para o Google Drive. O exemplo a seguir ilustra as tarefas do *script* de carregamento da DAG de produção extensionista, bem como a imagem do grafo gerado pelo Apache Airflow. No *script* observa-se a necessidade de declarar o diretório destino e o nome do arquivo .csv, também há o mapeamento do nome das colunas que serão necessárias ao Looker Studio.

```

...
datalake_extensao_dir = Variable.get("datalake")+ 'extensao/'
datalake_producoes_dir = datalake_extensao_dir+'producoes/'
encoding = "utf-8"
separador = ","
csv_quantitativo_producoes_extensionistas =
datalake_producoes_dir+'IFSC_EXTENSAO_SIGAA_QUANTITATIVO_PRODUCOES_
EXTENSIONISTAS.csv'
lista_arquivos = [csv_quantitativo_producoes_extensionistas]

with DAG(
    ...
    def cria_diretorio():
        if not os.path.exists(datalake_extensao_dir):
            os.makedirs(datalake_extensao_dir)
        elif not os.path.exists(datalake_producoes_dir):
            os.makedirs(datalake_producoes_dir)

    cria_datalake_producoes_extensao = PythonOperator(
        task_id='cria_datalake_producoes_extensao',
        python_callable=cria_diretorio,
    )

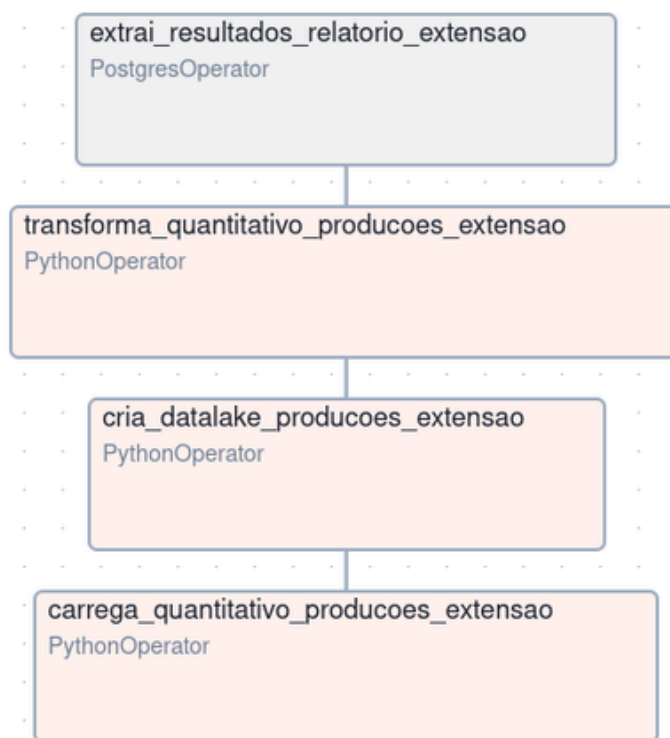
    def cria_planilha_quantitativo_producoes_extensao(ti):
        result_quantitativo_producoes_extensao =
ti.xcom_pull(task_ids='transforma_quantitativo_producoes_extensao')
        df_quantitativo_producoes_extensao =
pd.DataFrame(result_quantitativo_producoes_extensao)
        cabecalho_quantitativo_producoes_extensao =
['Câmpus','CEP','Ano','Categoria','Produções Extensionistas']

        if df_quantitativo_producoes_extensao.empty:
            df_quantitativo_producoes_extensao =
pd.DataFrame(columns=cabecalho_quantitativo_producoes_extensao)
        df_quantitativo_producoes_extensao.to_csv(csv_quantitativo_producoes_extensi
onistas, index=False, encoding=encoding, sep=separador)
        else:
df_quantitativo_producoes_extensao.to_csv(csv_quantitativo_producoes_extensi
onistas, index=False, header=cabecalho_quantitativo_producoes_extensao,
encoding=encoding, sep=separador)

    carrega_quantitativo_producoes_extensao = PythonOperator(
        task_id='carrega_quantitativo_producoes_extensao',
        python_callable=cria_planilha_quantitativo_producoes_extensao
    )

    extrai_resultados_relatorio_extensao >>
transforma_quantitativo_producoes_extensao >>
cria_datalake_producoes_extensao >> carrega_quantitativo_producoes_extensao

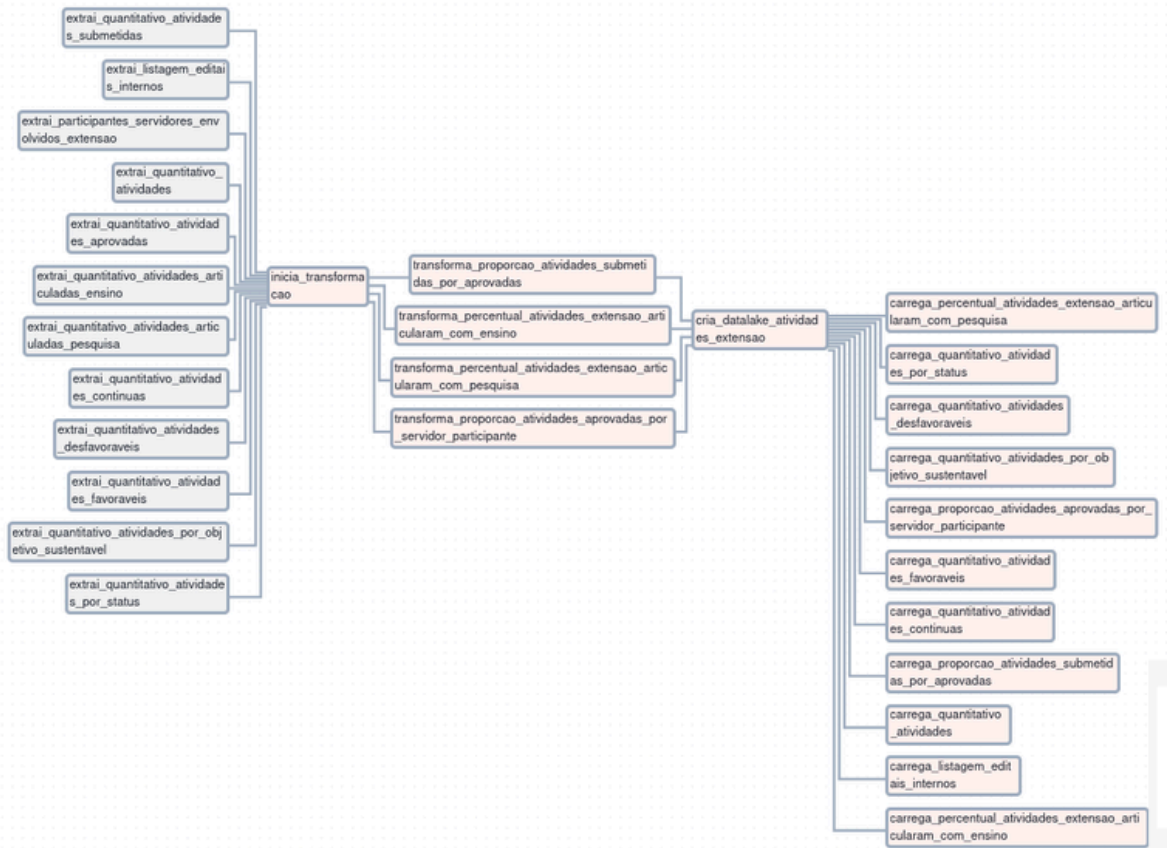
```



Ao todo foram criadas 13 DAGs para cada categoria de indicador de extensão, essas geraram 72 arquivos tabulados para 100 indicadores extraídos do SIGAA. Indica-se que as DAGs mantenham apenas o necessário, À medida que novos indicadores são identificados, a complexidade de manutenção da DAG cresce exponencialmente, pois é preciso cobrir as etapas de extração, tratamento e carregamento com uma ou mais tarefas (*tasks*) para cada etapa por indicador. É possível tratar isso com a programação orientada a objetos, no entanto, tal abordagem pode ser tornar mais complexa que a anterior à medida que o Apache Airflow integra outras instituições.

Para exemplificar a dimensão dos ciclos de extração, ilustra-se a seguir a imagem do grafo de atividades de extensão. Neste caso, como há uma extensa gama de indicadores, as suas tarefas se tornam extensas também, conseqüentemente com uma complexidade maior para estruturar seu processamento. Logo, a rastreabilidade pode se tornar um problema. Na imagem seguinte cada coluna representa o trabalho de um *worker* agendado para executar as tarefas ilustradas em quadrados. A cor vermelha indica que houve erro na execução, o verde sucesso, e o amarelo que a tarefa anterior teve interrupção.

# Carregamento



# EXIBIÇÃO

OBSERVATÓRIO DA EXTENSÃO

## **DADOS BEM APRESENTADOS SE TORNAM FERRAMENTAS DE TRANSFORMAÇÃO**

OS DADOS TABELADOS POR SI NEM SEMPRE TRAZEM UMA COMUNICAÇÃO CLARA E OBSERVÁVEL. POR CONTA DISSO, A EXIBIÇÃO DOS DADOS EM GRÁFICOS COM A AJUDA DE FILTROS OS TORNAM ACESSÍVEIS E SIGNIFICATIVOS. LOGO, ELES FALAM COM QUEM OS OBSERVA.

ESTA SEÇÃO TRAZ OS PASSOS TOMADOS PARA APRESENTAÇÃO DOS DADOS DO PRODUTO EDUCACIONAL.

Uma vez os dados estejam tabulados nos arquivos .csv e salvos no google drive, convertem-se os mesmos para o formato .gsheet. Acessa-se o Looker Studio com a conta google conectada e no painel criado faz-se a conexão aos dados em Adicionar Dados -> Planilhas Google. Uma vez adicionada um arquivo como fonte de dados, nas próximas vezes basta sobreescrever ou incrementar os dados atualizados.

Ao adicionar um arquivo como uma fonte de dados ,por vezes, é necessário editá-la em Recurso -> Gerenciar Fonte de Dados Adicionadas para melhor aproveitamento do Looker Studio. Por exemplo, ao editar a fonte de dados de produções extensionistas o Looker Studio já identifica as colunas conforme o nome dado no carregamento dos dados via Apache Airflow, mas alguns dados como CEP e Ano são identificados como tipo Texto, logo é coerente ajustá-los para tipo Geolocalização e Número respectivamente. Também nessa área é possível adicionar campos customizáveis (fx), como o CEP-Câmpus, que agrega os dois dados com a função CONCAT(CAST(CEP AS STRING), " - ", Câmpus) para maior eficiência no uso de gráficos no formato mapa.

Campo ↓	Tipo ↓	Agregação padrão ↓
<b>DIMENSÕES (6)</b>		
Ano	123 Número	Nenhum
Câmpus	ABC Texto	Nenhum
Categoria	ABC Texto	Nenhum
CEP	🌐 CEP	Nenhum
CEP-Câmpus <small>fx</small>	ABC Texto	Nenhum
Produções Extensionistas	123 Número	Soma
<b>MÉTRICAS (1)</b>		
Record Count	123 Número	Automático

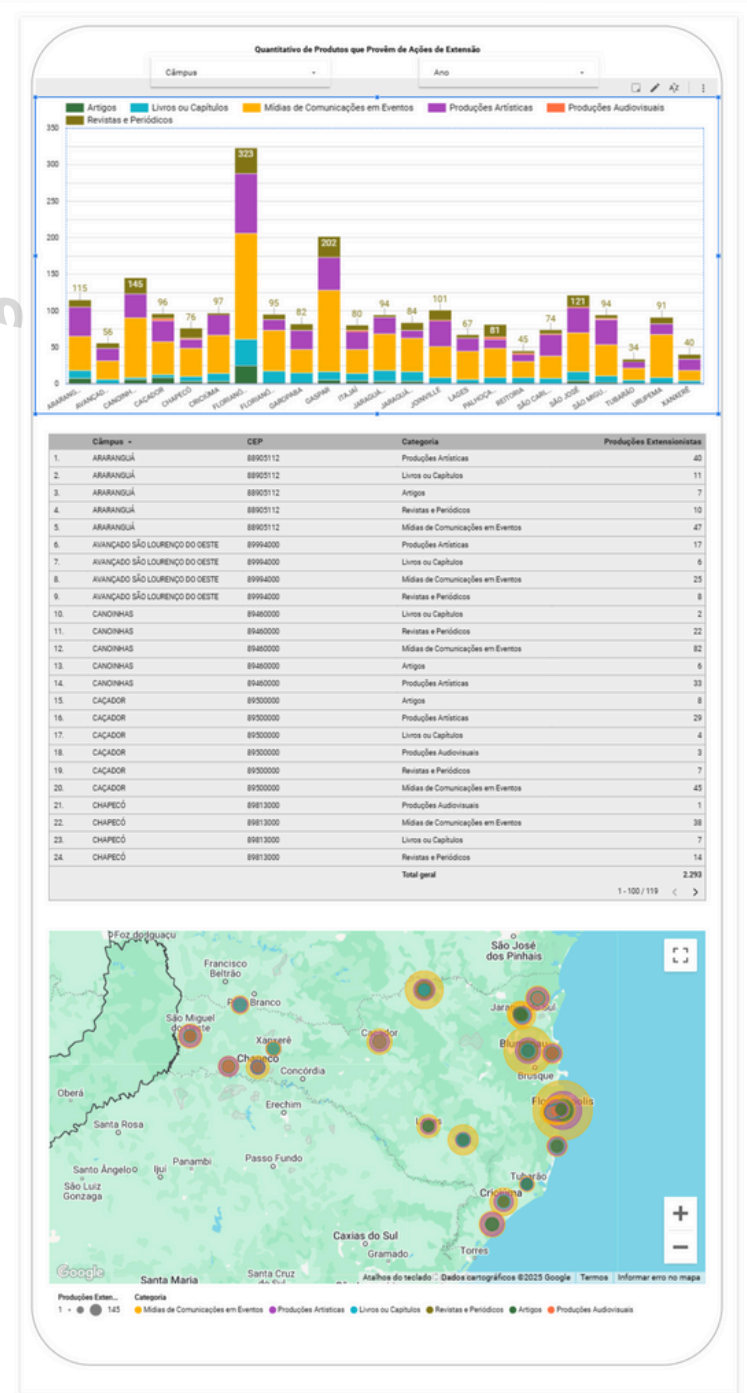
Com as fontes de dados cadastradas é possível construir o esquema de páginas em Página -> Gerenciar Páginas, assim de acordo com o layout escolhido o menu será exibido como uma barra superior horizontal ou uma barra lateral vertical de acordo com as categorias.

O tema e layout do projeto irão pré-definir alguns elementos estéticos, como tamanho da página, largura, *design* dos gráficos, esquema de cores, etc. No entanto, é provável que uma página precise de customizações específicas. Para isso há a opção em Página -> configurações da página atual.

Após a definição visual, há que se optar qual(is) gráficos irão compor a exibição dos dados. Neste produto educacional, com exceção das páginas de apresentação, optou-se por apresentá-los de 3 formas, a primeira por meio do gráfico de colunas simples e empilhadas que dá destaque a uma percepção numérica, a segunda escolhida foi a tabela que permite a análise detalhada dos dados, e a terceira a opção de usar o mapa balão, pois a EPT tem a característica de ter instituições com câmpus espalhados pelo território estadual, o mesmo ocorre com o IFSC.

Assim, conclui-se sucintamente os passos necessários para exibição de dados em painéis e o ciclo de extração, transformação e carga. Na imagem a seguir é apresentada a página do quantitativo da produção extensionista, nela há dois filtros de dados por câmpus e ano que afetam os 3 gráficos: colunas empilhadas, tabela e mapa balão, todos interativos. Observa-se também à direita dois menus de customização do gráfico de colunas compostas, o mais externo com a lista de campos cadastrados na fonte de dados QUANTITATIVO\_PRODUCOES\_EXTENSIONISTAS, e o menu colunar mais interno com as configurações, dimensões e agregações do gráfico selecionado.

# Exibição



### Gráfico

CONFIGURAÇÃO ESTILO

Fonte de dados: QUANTITATIVO\_PRODUCOES\_EXTEN...

COMBINAR DADOS

Dimensão de período: Adicionar dimensão

Dimensão: **MC Câmpus**

Detalhar:

Dimensão detalhada: **MC Categoria**

Métrica: **SUM Produções Extensionistas**

Métricas opcionais:

Controles deslizantes de métrica:

Classificar: **MC Câmpus**

Decrescente

Crescente

Classificação secundária: **MC Categoria**

Decrescente

Crescente

Filtro: Filtros Por Este Chart

ADICIONAR UM FILTRO

Interações de gráfico

Cruzamento de filtros:

Mudar classificação:

Zoom:

Adicionar um campo

Adicionar um parâmetro

Adicionar dados

Data da última atualização: 13/02/2023 16:14:08

# CONSIDERAÇÕES

OBSERVATÓRIO DA EXTENSÃO

**A REFLEXÃO DO QUE FOI APREENDIDO É O  
INÍCIO DOS PRÓXIMOS PASSOS**

AQUI ESTÁ O SEU PONTO DE PARTIDA  
PARA A MATURAÇÃO DE IDEIAS PARA  
NOVOS PROJETOS, LEIA VISLUMBRANDO AS  
POSSIBILIDADES DE INOVAÇÃO NO SEU  
AMBIENTE, SEJA UM AGENTE DE  
TRANSFORMAÇÃO!

ESTA SEÇÃO TRAZ BREVES  
CONSIDERAÇÕES A RESPEITO DAS  
IMPRESSÕES, REPLICABILIDADE E  
ESCALABILIDADE DO PRODUTO  
EDUCACIONAL

O Observatório da Extensão apresenta um significativo avanço na extração e visualização dos dados no IFSC, proporcionando uma análise mais acurada dos dados da extensão por câmpus e ano, oferecendo uma visualização do histórico para melhor análise pela gestão, apontando lacunas existentes na customização do SIGAA, na cultura digital dentro da instituição e na ampliação da solução para outras frentes da educação profissional e tecnológica.

Apesar de intuitiva na navegabilidade entre páginas, revela desafios iniciais para novos usuários, especialmente no que diz respeito à compreensão de termos e à usabilidade dos filtros, o que pode ser atribuído à fase de adaptação ao sistema. Para isso foi elaborado um painel de glossário de termos, porém, ainda há margem para melhorias de *design*(UI) e experiência do usuário (UX) na adoção de outras ferramentas de visualização de dados.

Não obstante, a automatização entre o processo de carregamento e exibição dos dados se mostra crucial para continuidade da solução, para tanto há que rever o contrato das ferramentas google ou considerar outra de uso livre que resolva esse gargalo. Sanar isso é relevante sobretudo em contextos de transição de funções de gestão, minimizando riscos de inconsistências na carga dos dados, essas que devem ser constantemente revisadas para desvelamento das informações condizentes com a realidade.

O Observatório da Extensão não apenas facilitou a afirmação de percepções tácitas, como a predominância de docentes em determinadas modalidades de extensão, também instigou a revisão de processos internos e políticas de registro de informações, apontando para a necessidade de reforçar a cultura de uso do sistema. A qual não deve ser direcionada para estimular um sistema de vigilância, e sim um escudo para a EPT.

A transparência dos dados, corrobora a importância do painel no atendimento às exigências da LAI e na aproximação do IFSC com a comunidade externa, evidenciando seu potencial como ferramenta estratégica de comunicação e melhoria contínua das práticas institucionais e dos mecanismos democráticos. Por fim, além da solução ensinar a construção de painéis do ensino e da pesquisa, possibilita a replicabilidade da solução em outras instituições independente do sistema de registro de dados que utilizam, também traz o potencial institucionalização da ferramenta em escala nacional mantido pelo Ministério da Educação, com o objetivo de fortalecer a educação e o preparo da população para o mundo do trabalho.

- AIRFLOW. **What is Airflow?** Disponível em <https://airflow.apache.org/docs/apache-airflow/2.10.3/index.html>.
- AIRFLOW. **Docker image for Apache Airflow.** Disponível em <https://airflow.apache.org/docs/docker-stack/index.html>.
- AIRFLOW. **Core Concepts.** Disponível em <https://airflow.apache.org/docs/apache-airflow/2.10.3/core-concepts/index.html>.
- AIRFLOW. **Taskflow.** Disponível em <https://airflow.apache.org/docs/apache-airflow/2.10.3/core-concepts/taskflow.html>.
- BARBIERI, Carlos. **Governança de Dados: práticas, conceitos e novos caminhos.** Rio de Janeiro, RJ: Alta Books, 2019.
- BRASIL. **Lei nº 12.527, de 18 de novembro de 2011.** Lei de Acesso à Informação. Disponível em [https://www.planalto.gov.br/ccivil\\_03/\\_ato2011-2014/2011/lei/l12527.htm](https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm).
- BRASIL. **Lei nº 12.965, de 23 de abril de 2014.** Marco Civil da Internet. Disponível em [https://www.planalto.gov.br/ccivil\\_03/\\_ato2011-2014/2014/lei/l12965.htm](https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm).
- BRASIL. **Lei nº 13.709, de 14 de agosto de 2018.** Lei Geral de Proteção de Dados Pessoais. Disponível em [https://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2018/lei/l13709.htm](https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm).
- IFSC. **Plano de Dados Abertos 2021-2023. 2021.** Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina. Disponível em: <https://www.ifsc.edu.br/12-dados-abertos>.
- GOOGLE. **Documentação da família de produtos do Looker.** Disponível em <https://cloud.google.com/looker/docs?hl=pt-br>.
- RODRIGUES, Valéria M. **O Fórum de pró-reitores de extensão e sua contribuição no debate sobre a extensão universitária.** Revista Educação e Políticas em Debate, [S. l.], v. 4, n. 2, 2016. DOI: 10.14393/REPOD-v4n2a2015-34562. Disponível em: <https://seer.ufu.br/index.php/revistaeducaopoliticas/article/view/34562>. Acesso em: 29 jun. 2024.
- FORPROEXT. **Indissociabilidade entre ensino-pesquisa-extensão e flexibilização curricular: uma visão da extensão.** Porto Alegre: UFRGS; Brasília: MEC/SESu, 2006. Disponível em: [http://www.uemg.br/downloads/indissociabilidade\\_ensino\\_pesquisa\\_extensao.pdf](http://www.uemg.br/downloads/indissociabilidade_ensino_pesquisa_extensao.pdf). Acesso em 5 de abril de 2024.
- TCU. **Acórdão 461/2022.** Indicadores de Gestão e Desempenho da Rede Federal de Educação. Tribunal de Contas da União, 2022. Disponível em [https://pesquisa.apps.tcu.gov.br/documento/acordao-completo/\\*/NUMACORDAO%253A461%2520ANOACORDAO%253A2022/DTRELEVANCIA%2520desc%252C%2520NUMACORDAOINT%2520desc/0/%2520](https://pesquisa.apps.tcu.gov.br/documento/acordao-completo/*/NUMACORDAO%253A461%2520ANOACORDAO%253A2022/DTRELEVANCIA%2520desc%252C%2520NUMACORDAOINT%2520desc/0/%2520). Acesso em 11 abril 2024.