

PROJETO INTEGRADOR DE PROGRAMAÇÃO

Relatório de Atividades Desenvolvidas

**Ana Paula M. Cichacz
Edilvan Rodrigues Ferreira
Ketlhin Taynara Nezi Santana**

Relatório de Projeto apresentado à Universidade Estadual do Centro-Oeste – Unicentro/Cedeteg, como quesito para conclusão do 1º semestre do Curso de Tecnologia em Big Data no Agronegócio na disciplina de Projeto Integrador ministrada pelos professores Carolina Paula de Almeida e Luara Peterlini.

Guarapuava
2025

O AgroNewsBot é um robô de notícias criado com o objetivo de facilitar o acesso dos produtores rurais às principais informações do setor agrícola. Com base na percepção de que muitos enfrentam dificuldades para acompanhar notícias relevantes devido à rotina intensa e ao excesso de conteúdo disponível online, propusemos uma solução prática e automatizada. O projeto utiliza web scraping para coletar e enviar notícias diretamente ao celular dos usuários por meio do aplicativo Telegram. Isso garante rapidez, clareza e acesso contínuo às atualizações do setor, contribuindo com a tomada de decisões no campo.

1. Objetivos

1.1 Objetivo Geral

Desenvolver um bot funcional no Telegram, que colete automaticamente notícias de sites especializados do agronegócio, com o objetivo de centralizar e facilitar o acesso à informação para produtores rurais e demais interessados.

1.2 Objetivos Específicos

Durante o desenvolvimento do AgroNewsBot, buscamos aplicar na prática os conhecimentos adquiridos em sala de aula. Um dos principais objetivos específicos foi utilizar técnicas de web scraping com a linguagem Python para realizar a extração automatizada de notícias. Para isso, empregamos bibliotecas como requests, BeautifulSoup, csv e telepot, que possibilitaram a coleta, organização e envio dos dados. Também foi parte do nosso objetivo criar menus interativos dentro do terminal, com a intenção de futuramente integrar totalmente o sistema ao Telegram. Além disso, nos propusemos a permitir que o usuário pudesse escolher a quantidade de notícias que desejava visualizar (entre 5, 10 ou 15), e a desenvolver um menu funcional contendo opções como ajuda, sobre e encerramento do programa. Por fim, trabalhamos na melhoria da experiência do usuário com a inclusão de botões de retorno e comandos personalizados que facilitam a navegação.

2. Metodologia

A metodologia utilizada foi baseada em uma divisão de etapas e tarefas entre os membros da equipe. O desenvolvimento foi feito de forma incremental e colaborativa, com ênfase na

construção prática do sistema. A primeira etapa consistiu na coleta de dados, seguida pelo tratamento das informações, desenvolvimento das funcionalidades do bot e testes. O código foi escrito em Python e executado localmente para validação de sua funcionalidade

2.1 Coleta de Dados com programação em *python*

A coleta de dados foi feita por meio da técnica de web scraping, com a utilização da biblioteca requests para acesso às páginas HTML e BeautifulSoup para extração dos elementos desejados (títulos, links, resumos). As páginas utilizadas foram:

- [Globo Rural – Últimas notícias](#)
- [Canal Rural – Últimas notícias](#)
- [AgroLink – Notícias](#)

Cada notícia extraída foi formatada com título, link e fonte, e salva em um arquivo .csv para possível reaproveitamento posterior. O código ainda oferece opção de filtragem por site e quantidade de notícias.

A seguir, apresentamos um trecho de código responsável por extrair e exibir as últimas notícias do site Globo Rural. Essa função faz parte do processo de coleta e formatação dos dados:

```
def globo_rural():
    print(f'{mensagem1.upper()}')
    for div in noticias_globorural:
        h2 = div.find('h2', class_='feed-post-link gui-color-primary gui-color-hover')
        if h2:
            link_noticia = h2.find('a')
            if link_noticia:
                texto = re.sub(r'\s+', ' ', link_noticia.text).strip()
                link = link_noticia['href']
                print(f'👉 *{texto}*\n🔗 {link}\n\n')
                salvar_csv('Globo Rural', texto, link)
```

Esse trecho demonstra a função `globo_rural()`, que utiliza BeautifulSoup para localizar os elementos HTML das notícias mais recentes no site, extrair os textos com limpeza de espaços e formatar a saída com emojis, negrito e link clicável. Além disso, os dados são armazenados em um arquivo .csv com a função `salvar_csv()`.

O uso de expressões regulares (`re.sub`) foi essencial para garantir que os textos coletados não viessem com espaços quebrados ou caracteres especiais. Essa abordagem foi repetida nas demais funções (`canalrural()` e `agrolink()`), adaptadas à estrutura de cada site.

2.2 Tratamento de Dados

Após a coleta, os dados passaram por uma etapa de limpeza e formatação. Foram utilizados recursos como `strip()` e expressões regulares para remover espaços extras e organizar o conteúdo. Também foi necessário tratar links relativos (como no Agrolink) para garantir URLs completas. As mensagens foram preparadas com negrito, emojis e espaçamento adequado, pensando na clareza de leitura via Telegram.

2.3 Funcionalidades Adicionais Desenvolvidas

- Menu interativo com escolha de site e quantidade de notícias.
- Limite personalizável: o usuário pode escolher visualizar 5, 10 ou 15 notícias.
- Ajuda detalhada com explicação de cada opção do bot.
- Botões de retorno e encerramento para facilitar navegação.
- Exportação automática para arquivo `.csv`.

3. Considerações Finais

O projeto AgroNewsBot atingiu com êxito todos os objetivos propostos. A ferramenta foi desenvolvida, testada e entregue funcionando de forma prática, leve e acessível, proporcionando aos produtores rurais uma nova maneira de se manterem atualizados sobre o setor agrícola por meio do Telegram.

Entre os resultados obtidos, destacam-se a entrega rápida e organizada das notícias, com uma formatação clara e legível. O sistema permite que o usuário escolha quantas notícias deseja visualizar, o que proporciona controle sobre a quantidade de informações recebidas. Além disso, os dados podem ser exportados em formato `.csv`, facilitando seu reaproveitamento. A manutenção do código mostrou-se simples, e o projeto possui boas possibilidades de expansão futura.

Durante a realização do projeto, alguns pontos fortes ficaram evidentes, como a

acessibilidade do bot, sua interface intuitiva, o uso de fontes confiáveis e específicas do agronegócio e a compatibilidade com conexões de internet limitadas, comum em zonas rurais. No entanto, também enfrentamos alguns desafios técnicos. Um dos principais problemas identificados foi que, ao selecionar a opção de “Sair” no Telegram, o bot encerra completamente sua execução e não pode ser reiniciado sem rodar novamente o código manualmente. Outro obstáculo foi a limitação de caracteres imposta pelo Telegram: ao escolher a exibição de 15 notícias por site, a mensagem excede o limite permitido, o que impede seu envio.

Pensando no aprimoramento do projeto, vislumbramos diversas possibilidades de evolução. Pretendemos futuramente hospedar o bot em uma plataforma de nuvem, o que permitirá seu funcionamento contínuo. Também consideramos a integração total com a API oficial do Telegram, a programação de envios automáticos de acordo com temas e horários definidos, a inclusão de novas fontes de informação, além da expansão para outras plataformas como o WhatsApp. Outro passo importante será o desenvolvimento de uma interface gráfica que torne o uso do bot ainda mais intuitivo e acessível

4. Referências Bibliográficas

BeautifulSoup Documentation: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

Requests Documentation: <https://docs.python-requests.org/>

Telepot (Telegram API): <https://telepot.readthedocs.io/en/latest/>

Sites utilizados na coleta:

- <https://globo rural.globo.com/ultimas-noticias/>
- <https://www.canalrural.com.br/ultimas-noticias/>
- <https://www.agrolink.com.br/noticias/>

Vídeo-base utilizado no aprendizado:

- *Extraíndo Dados da Web com Python e BeautifulSoup – AulasDev (YouTube)*

Ferramentas de apoio utilizadas:

- *ChatGPT – OpenAI* (<https://chat.openai.com>): apoio na organização, escrita técnica e explicações de código Python.
- *Gemini – Google AI* (<https://gemini.google.com>): auxílio complementar em sugestões de estrutura e revisão textual.

MATRIZES

A.1. Matriz de habilidades inicial:

FERRAMENTAS	Ana Paula	Edilvan	Ketlin
Python	3	3	3
HTML	3	6	1
MVP	1	2	1
Biblioteca Request	1	3	1
Biblioteca BeautifulSoup	1	3	1
Biblioteca Selenium	1	1	1
GitHub	3	5	3
Colab	1	3	1
GoogleWorkspace	3	3	3
Lógica de Programação	3	5	3

1 - Ouvi Falar

2 - Entendo

3 - Sei fazer com apoio

4 - Sei fazer sem apoio

5 - Sei ensinar

6 - Sei criar

A.2. Matriz de habilidade final e atualizada:

FERRAMENTAS	ANA PAULA	EDILVAN	KETLHIN
Python	3	4	3
HTML	3	6	3
MVP	1	3	2
Biblioteca Request	3	5	3
Biblioteca BeautifulSoup	3	4	3
Biblioteca Selenium	3	2	3
GitHub	3	5	4
Colab	3	5	4
GoogleWorkspace	2	3	3
Logica de Programação	4	5	4

1 - Ouvi Falar

2 - Entendo

3 - Sei fazer com apoio

4 - Sei fazer sem apoio

5 - Sei ensinar

6 - Sei criar

Link postagem no LinkedIn: https://www.linkedin.com/posts/ketlhin-nezi-b41974366_python-telegrambot-webscraping-activity-7349477311072657408-xv2r?utm_medium=ios_app&rcm=ACoAAFrPZGEBFxErN0HZsmO-q7AyQYhuLMd2_Vc&utm_source=social_share_send&utm_campaign=copy_link

Link repositório: <https://github.com/edilvannf/AgroNews-Bot>

Link do Pitch: https://youtu.be/baVDLrrC44s?si=W0lt-EcZMz_qYil2