

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - UTFPR

CAMPUS DOIS VIZINHOS

ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS

BRUNA MATTIOLI DE OLIVEIRA

GABRIEL ANDRADE VARGA

PROJETO INTEGRADOR 2

DOIS VIZINHOS

JULHO / 2021

SUMÁRIO

ADMINISTRAÇÃO E GERENCIAMENTO DE BANCO DE DADOS	3
LINGUAGENS DE PROGRAMAÇÃO PARA CIÊNCIA DE DADOS	12
PREPARAÇÃO PEDAGÓGICA	17

1. Administração e Gerenciamento de Banco de Dados

a. Criação de 5 índices

Inicialmente, iremos criar uma tabela extra para junção dos dados e criação de um campo novo:

```
-- Criação de uma base extra no banco de dados
CREATE TABLE desmatamento_completa AS
SELECT id_desmatamento,
d.id_municipio,
nome_municipio,
ano, sigla_uf,
km_hidrografia,
km_area_total,
km_desmatado,
km_n_floresta,
km_nuvem,
km_n_obs,
km_floresta,

CASE WHEN sigla_uf = 'PA' THEN 'Pará'
      WHEN sigla_uf = 'MT' THEN 'Mato Grosso'
      WHEN sigla_uf = 'MA' THEN 'Maranhão'
      WHEN sigla_uf = 'RO' THEN 'Rondônia'
      WHEN sigla_uf = 'TO' THEN 'Tocantins'
      WHEN sigla_uf = 'AC' THEN 'Acre'
      WHEN sigla_uf = 'AM' THEN 'Amazonas'
      WHEN sigla_uf = 'RR' THEN 'Roraima'
      WHEN sigla_uf = 'AP' THEN 'Amapá'
ELSE '' END AS nome_uf

FROM desmatamento AS d, uf AS u, municipio AS m, ano AS a
WHERE d.id_municipio = m.id_municipio AND d.id_ano = a.id_ano AND m.id_uf = u.id_uf;
```

- Índice 1

A criação do primeiro índice será feita numa consulta envolvendo a função WHERE:

```
-- Criação do primeiro índice (particionado)
ANALYZE desmatamento_completa;
```

```
EXPLAIN ANALYZE SELECT nome_municipio, km_area_total
FROM desmatamento_completa
WHERE km_area_total >= 1000 AND km_area_total < 10000;
```

O comando Analyze atualiza as estatísticas do Plano de Consulta da base. Já o comando Explain Analyze nos mostra o Plano de Consulta com informações sobre o tempo de planejamento da consulta e o tempo real de execução da mesma. Neste caso, o filtro foi efetuado de uma maneira não otimizada (via Seq Scan).

Data Output	Explain	Messages	Notifications
	QUERY PLAN text		
1	Seq Scan on desmatamento_completa (cost=0.00..454.40 rows=9073 width=19) (actual time=0.007..2.254 rows=9072 loops=1)		
2	Filter: ((km_area_total >= '1000'::numeric) AND (km_area_total < '10000'::numeric))		
3	Rows Removed by Filter: 6888		
4	Planning Time: 0.115 ms		
5	Execution Time: 2.405 ms		

Para otimizar esta consulta, foi criado um indexador particionado utilizando o filtro anterior e a chave primária da base:

```
CREATE INDEX idx_km_area_total ON desmatamento_completa(id_desmatamento)
WHERE km_area_total >= 1000 AND km_area_total < 10000;
```

Atualizando as estatísticas e verificando o novo Plano de Consulta, vemos que agora o indexador está sendo utilizado:

```
ANALYZE desmatamento_completa;

EXPLAIN ANALYZE SELECT nome_municipio, km_area_total
FROM desmatamento_completa
WHERE km_area_total >= 1000 AND km_area_total < 10000;
```

Data Output	Explain	Messages	Notifications
	QUERY PLAN text		
1	Index Scan using idx_km_area_total on desmatamento_completa (cost=0.29..372.10 rows=9073 width=19) (actual time=0.023..1.366 rows=9072 loops=1)		
2	Planning Time: 0.156 ms		
3	Execution Time: 1.512 ms		

Após a utilização do indexador, houve uma redução de tempo na execução da consulta.

- Índice 2

A criação do segundo índice também será feita numa consulta envolvendo a função WHERE, porém utilizando um agrupamento de base via GROUP BY:

```
ANALYZE desmatamento_completa;

EXPLAIN ANALYZE SELECT sigla_uf, SUM(km_hidrografia) AS total_hidrografia
FROM desmatamento_completa
WHERE sigla_uf = 'PA'
GROUP BY sigla_uf;
```

Data Output	Explain	Messages	Notifications
	QUERY PLAN text		
1	GroupAggregate (cost=0.00..429.63 rows=9 width=35) (actual time=1.458..1.458 rows=1 loops=1)		
2	Group Key: sigla_uf		
3	-> Seq Scan on desmatamento_completa (cost=0.00..414.50 rows=3003 width=8) (actual time=0.005..1.174 rows=3003 loops=1)		
4	Filter: ((sigla_uf)::text = 'PA'::text)		
5	Rows Removed by Filter: 12957		
6	Planning Time: 0.076 ms		
7	Execution Time: 1.469 ms		

Para otimizar esta consulta, será criado um índice particionado utilizando o filtro anterior e a chave primária da base:

```
CREATE INDEX idx_km_hidrografia ON desmatamento_completa(id_desmatamento)
WHERE sigla_uf = 'PA';
```

Atualizando as estatísticas e verificando o novo Plano de Consulta, vemos que agora o indexador está sendo utilizado:

Data Output	Explain	Messages	Notifications
	QUERY PLAN text		
1	GroupAggregate (cost=0.28..150.38 rows=9 width=35) (actual time=0.861..0.861 rows=1 loops=1)		
2	Group Key: sigla_uf		
3	-> Index Scan using idx_km_hidrografia on desmatamento_completa (cost=0.28..135.25 rows=3003 width=8) (actual time=0.022..0.500 rows=3003 loops=1)		
4	Planning Time: 0.100 ms		
5	Execution Time: 0.874 ms		

Após a utilização do indexador, houve uma redução de tempo na execução da consulta.

- Índice 3

A criação do terceiro índice será feita para uma consulta utilizando a chave estrangeira:

```
-- Criação do terceiro índice (índice para a chave estrangeira)
ANALYZE desmatamento;

EXPLAIN ANALYZE SELECT id_municipio, km_floresta
FROM desmatamento
```

```
WHERE id_municipio IN (50, 56, 78, 106, 569);
```

Data Output	Explain	Messages	Notifications
QUERY PLAN text			
1	Seq Scan on desmatamento (cost=0.00..420.35 rows=105 width=10) (actual time=0.011..1.492 rows=105 loops=1)		
2	Filter: (id_municipio = ANY ('{50,56,78,106,569}'::integer[]))		
3	Rows Removed by Filter: 15855		
4	Planning Time: 0.063 ms		
5	Execution Time: 1.500 ms		

Para otimizar esta consulta, será criado um índice utilizando a chave estrangeira da base:

```
CREATE INDEX idx_id_municipio ON desmatamento(id_municipio);
```

Atualizando as estatísticas e verificando o novo Plano de Consulta, vemos que agora o indexador está sendo utilizado:

```
ANALYZE desmatamento;

EXPLAIN ANALYZE SELECT id_municipio, km_floresta
FROM desmatamento
WHERE id_municipio IN (50, 56, 78, 106, 569);
```

Data Output	Explain	Messages	Notifications
QUERY PLAN text			
1	Bitmap Heap Scan on desmatamento (cost=22.24..174.77 rows=105 width=10) (actual time=0.049..0.091 rows=105 loops=1)		
2	Recheck Cond: (id_municipio = ANY ('{50,56,78,106,569}'::integer[]))		
3	Heap Blocks: exact=68		
4	-> Bitmap Index Scan on idx_id_municipio (cost=0.00..22.21 rows=105 width=0) (actual time=0.042..0.042 rows=105 loops=1)		
5	Index Cond: (id_municipio = ANY ('{50,56,78,106,569}'::integer[]))		
6	Planning Time: 0.092 ms		
7	Execution Time: 0.115 ms		

Após a utilização do indexador, houve uma redução de tempo na execução da consulta.

- Índice 4

A criação do quarto índice será feita para uma consulta utilizando um campo discreto:

```
-- Criação do quarto índice (índice para um campo discreto)
ANALYZE municipio;
```

```
EXPLAIN ANALYZE SELECT id_municipio
FROM municipio
WHERE nome_municipio = 'Alta Floresta';
```

Data Output	Explain	Messages	Notifications
<div> <div>▲</div> <div>QUERY PLAN</div> <div>text</div> </div>			
1	Seq Scan on municipio (cost=0.00..15.50 rows=1 width=4) (actual time=0.007..0.048 rows=1 loops=1)		
2	Filter: ((nome_municipio)::text = 'Alta Floresta'::text)		
3	Rows Removed by Filter: 759		
4	Planning Time: 0.043 ms		
5	Execution Time: 0.054 ms		

Para otimizar esta consulta, será criado um índice B-Tree / Bitmap utilizando o campo discreto da base:

```
CREATE EXTENSION btree_gin;
CREATE INDEX idx_nm_municipio_Bitmap ON municipio USING gin (nome_municipio);
```

Atualizando as estatísticas e verificando o novo Plano de Consulta, vemos que agora o indexador está sendo utilizado:

```
ANALYZE municipio;

EXPLAIN ANALYZE SELECT id_municipio
FROM municipio
WHERE nome_municipio = 'Alta Floresta';
```

Data Output	Explain	Messages	Notifications
<div> <div>▲</div> <div>QUERY PLAN</div> <div>text</div> </div>			
1	Bitmap Heap Scan on municipio (cost=8.01..12.02 rows=1 width=4) (actual time=0.013..0.013 rows=1 loops=1)		
2	Recheck Cond: ((nome_municipio)::text = 'Alta Floresta'::text)		
3	Heap Blocks: exact=1		
4	-> Bitmap Index Scan on idx_nm_municipio_bitmap (cost=0.00..8.01 rows=1 width=0) (actual time=0.010..0.010 rows=1 loops=1)		
5	Index Cond: ((nome_municipio)::text = 'Alta Floresta'::text)		
6	Planning Time: 0.074 ms		
7	Execution Time: 0.029 ms		

Após a utilização do indexador, houve uma redução de tempo na execução da consulta.

- Índice 5

A criação do quinto índice será feita para uma consulta utilizando um campo textual:

```
-- Criação do quinto índice (índice para um campo textual)
ANALYZE desmatamento_completa;

EXPLAIN ANALYZE SELECT km_desmatado, nome_municipio, ano
FROM desmatamento_completa
WHERE nome_municipio LIKE '%São%';
```

Data Output	Explain	Messages	Notifications
QUERY PLAN			
text			
1	Seq Scan on desmatamento_completa (cost=0.00..414.50 rows=1112 width=24) (actual time=0.006..1.379 rows=1092 loops=1)		
2	Filter: ((nome_municipio)::text ~~ '%São% '::text)		
3	Rows Removed by Filter: 14868		
4	Planning Time: 0.083 ms		
5	Execution Time: 1.404 ms		

Para otimizar esta consulta, será criado um índice Full Text Search utilizando o campo textual da base:

```
CREATE EXTENSION pg_trgm;
CREATE INDEX idx_nome_municipio_gin ON desmatamento_completa USING GIN (nome_municipio gin_trgm_ops);
```

Atualizando as estatísticas e verificando o novo Plano de Consulta, vemos que agora o indexador está sendo utilizado:

Data Output	Explain	Messages	Notifications
QUERY PLAN			
text			
1	Bitmap Heap Scan on desmatamento_completa (cost=20.61..249.51 rows=1112 width=24) (actual time=0.083..0.386 rows=1092 loops=1)		
2	Recheck Cond: ((nome_municipio)::text ~~ '%São% '::text)		
3	Rows Removed by Index Recheck: 21		
4	Heap Blocks: exact=214		
5	-> Bitmap Index Scan on idx_nome_municipio_gin (cost=0.00..20.34 rows=1112 width=0) (actual time=0.068..0.068 rows=1113 loops=1)		
6	Index Cond: ((nome_municipio)::text ~~ '%São% '::text)		
7	Planning Time: 0.255 ms		
8	Execution Time: 0.421 ms		

Após a utilização do indexador, houve uma redução de tempo na execução da consulta.

b. Implementação de 1 trigger

- Criação de uma tabela auxiliar para a utilização do Trigger:

```
CREATE TABLE desmatAtualizado (
```



```

id_desmatamento SERIAL PRIMARY KEY,
km_hidrografia NUMERIC(30,2),
km_area_total NUMERIC(30,2),
km_desmatado NUMERIC(30,2),
km_n_floresta NUMERIC(30,2),
km_nuvem NUMERIC(30,2),
km_n_obs NUMERIC(30,2),
km_floresta NUMERIC(30,2),
id_municipio INTEGER,
id_ano INTEGER
);

```

- Criação da Trigger

```

CREATE OR REPLACE FUNCTION atualizaDesmatamento()
RETURNS trigger AS $$
BEGIN
    INSERT INTO desmatAtualizado VALUES (OLD.id_desmatamento, OLD.km_hidrografia, OLD.km_area_total, OLD.km_desmatado, OLD.km_n_floresta,
    OLD.km_nuvem, OLD.km_n_obs, OLD.km_floresta, OLD.id_municipio, OLD.id_ano);
    RETURN NULL;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER TrigDesmatamento AFTER UPDATE ON desmatamento
FOR EACH ROW EXECUTE PROCEDURE atualizaDesmatamento();

```

A função existe para guardar os registros da tabela “desmatamento”, após ser feito UPDATE em algum registro. Após o UPDATE na tabela “desmatamento”, a trigger “TrigDesmatamento” executa a função atualizaDesmantamento(), que insere o registro que existia antes de ser atualizado na tabela “desmatAtualizado”. Dessa forma o registro anterior não é perdido ao ser atualizado e fica registrado para que possa ser consultado no futuro.

c. Criação de 3 usuários

Definição do usuário Analista de Dados: pode consultar todas as tabelas para fazer as análises, porém não pode deletar nem atualizar as tabelas já existentes.

- Criação do usuário:

```

CREATE USER analista_dados;

```

- Atribuição da permissão de consulta em todas as bases e campos do Banco de Dados:

```
GRANT SELECT (id_ano, ano) ON ano TO analista_dados;

GRANT SELECT (id_desmatamento, km_hidrografia, km_area_total, km_desmatado, km_n_floresta,
km_nuvem, km_n_obs, km_floresta, id_municipio, id_ano) ON desmatamento TO analista_dados;

GRANT SELECT (id_municipio, nome_municipio, id_uf) ON municipio TO analista_dados;

GRANT SELECT (id_uf, sigla_uf) ON uf TO analista_dados;
```

Definição do usuário Analista de Banco de Dados: tem acesso a todos os privilégios disponíveis em todas as tabelas do Banco de Dados. Ele é responsável pela manutenção e otimização do BD.

- Criação do usuário:

```
CREATE USER analista_banco_dados;
```

- Atribuição da permissão de todos os privilégios em todas as bases e campos do Banco de Dados:

```
GRANT ALL ON ano TO analista_banco_dados;
GRANT ALL ON desmatamento TO analista_banco_dados;
GRANT ALL ON municipio TO analista_banco_dados;
GRANT ALL ON uf TO analista_banco_dados;
```

Definição do usuário Gestor Município: pode consultar, inserir, atualizar ou deletar informações da tabela municipio.

- Criação do usuário:

```
CREATE USER gestor_municipio;
```

- Atribuição da permissão de consulta, inserção, atualização ou exclusão de informações na tabela municipio:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON municipio TO gestor_municipio;
```

d. Simulação de 2 transações

- Transação 1

```
-- TRANSAÇÃO 1
BEGIN;
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
INSERT INTO desmatamento VALUES (6,10, 100, 30, 30, 10, 1100, 20, 1, 1);
SELECT * FROM desmatamento WHERE id_desmatamento = 6;
COMMIT;
```

A transação 1 tem o nível de isolamento o Read Committed, insere registros da tabela “desmatamento” e faz a consulta do registro inserido na tabela “desmatamento”. Depois é executado o Commit para que a transação altere de fato o Banco de Dados.

- Transação 2

```
-- TRANSAÇÃO 2
BEGIN;
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SELECT * FROM municipio WHERE nome_municipio = 'São Félix do Xingu';
UPDATE municipio SET nome_municipio = 'Sao Felix do Xingu' WHERE nome_municipio = 'São Félix do Xingu';
COMMIT;
```

A transação 2 tem o nível de isolamento o Read Committed, faz uma consulta aos registros da tabela “municipio” que são iguais a um determinado texto, depois faz a alteração do nome deste município. Depois é executado o Commit para que a transação altere de fato o Banco de Dados.

2. Linguagens de Programação para Ciência de Dados

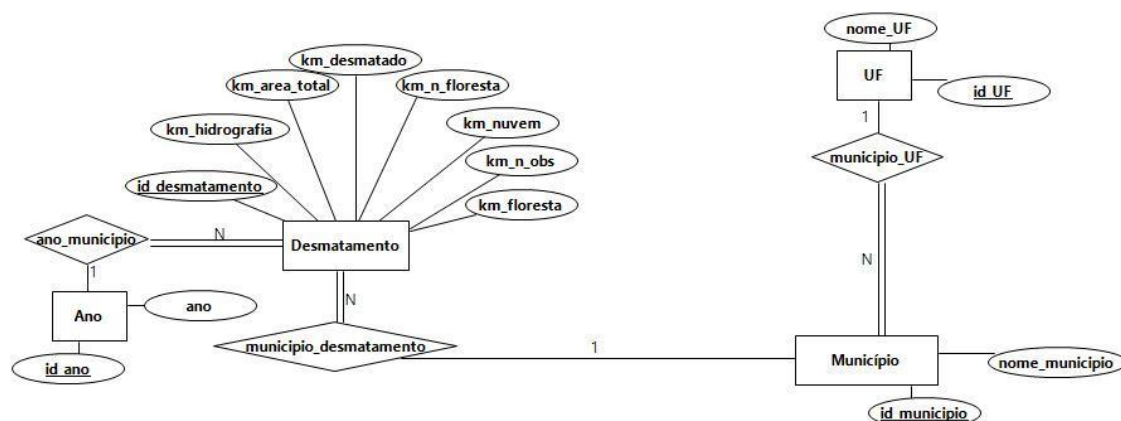
A fonte de base de dados utilizada para a realização deste Projeto Integrador (PI) é proveniente do projeto PRODES Amazônia ([PRODES - Desflorestamento nos Municípios \(inpe.br\)](http://prodes.inpe.br)). Neste projeto, é realizado o monitoramento por satélites do desmatamento por corte raso na Amazônia Legal.

Para a construção da base, foram consolidadas as informações de desmatamento do período de 2000 até 2020. Selecionamos as variáveis que ajudam a verificar a evolução do desmatamento nos Estados e Municípios da Amazônia Legal.

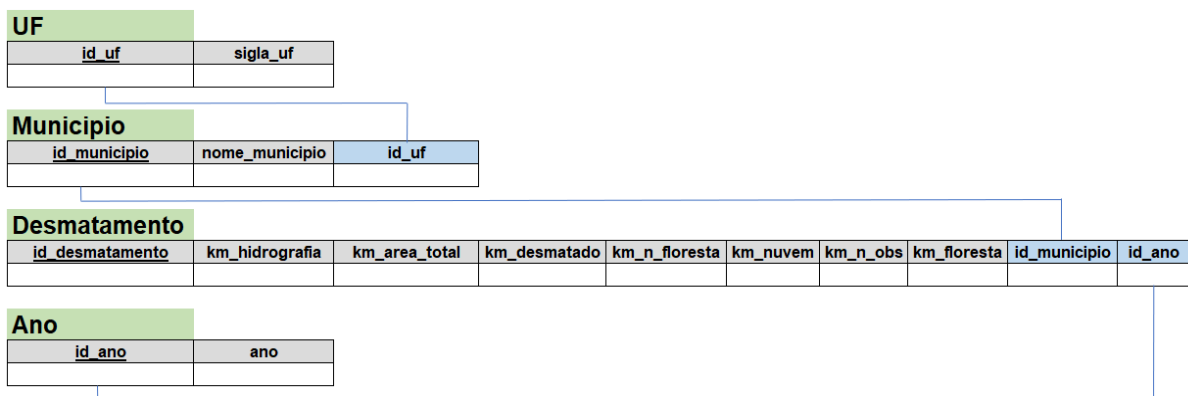
As variáveis utilizadas foram: Município (nome do município), Codlbg (código do município no Instituto Brasileiro de Geografia e Estatística), Estado (sigla da UF do Estado), AreaKm2 (área total, em km²), Desmatado (área desmatada, em km²), Floresta (área de floresta, em km²), Nuvem (área que no momento da captura das imagens do satélite estava com nuvem e não foi possível verificar o tipo de área real, em km²), NaoObservado (área que não foi capturada nas imagens de satélites, em km²), NaoFloresta (área que não tem florestas, em km²), Hidrografia (área com hidrografia, em km²) e Ano (ano da medição).

Para a criação do Banco de Dados (BD), foi necessário a elaboração de um Modelo Entidade Relacionamento (MER) e de um Modelo Relacional (MR) com a adição de novos campos:

- Modelo Entidade Relacionamento



- Modelo Relacional



Após a imputação dos dados no BD, foi feita a análise de dados com a utilização de Python para entender as características da evolução do desmatamento na Amazônia Legal.

- Descreve dos dados

Estatísticas	id_desmatamento	ano	km_hidrografia	km_area_total	km_desmatado	km_n_floresta	km_nuvem	km_n_obs	km_floresta
Contagem	15.960	15.960	15.960	15.960	15.960	15.960	15.960	15.960	15.960
Média	7.980	2.010	149	6.668	949	1.265	318	100	3.971
Desvio Padrão	4.607	6	430	13.858	1.360	2.402	1.983	576	11.930
Valor Mínimo	1	2.000	0	64	0	0	0	0	0
Primeiro Quartil	3.991	2.005	0	892	120	5	0	0	18
Segundo Quartil (Mediana)	7.980	2.010	4	2.380	450	196	0	0	222
Terceiro Quartil	11.970	2.015	65	6.422	1.250	1.437	5	1	2.296
Valor Máximo	15.960	2.020	4.500	159.540	19.886	19.781	68.395	11.321	152.014

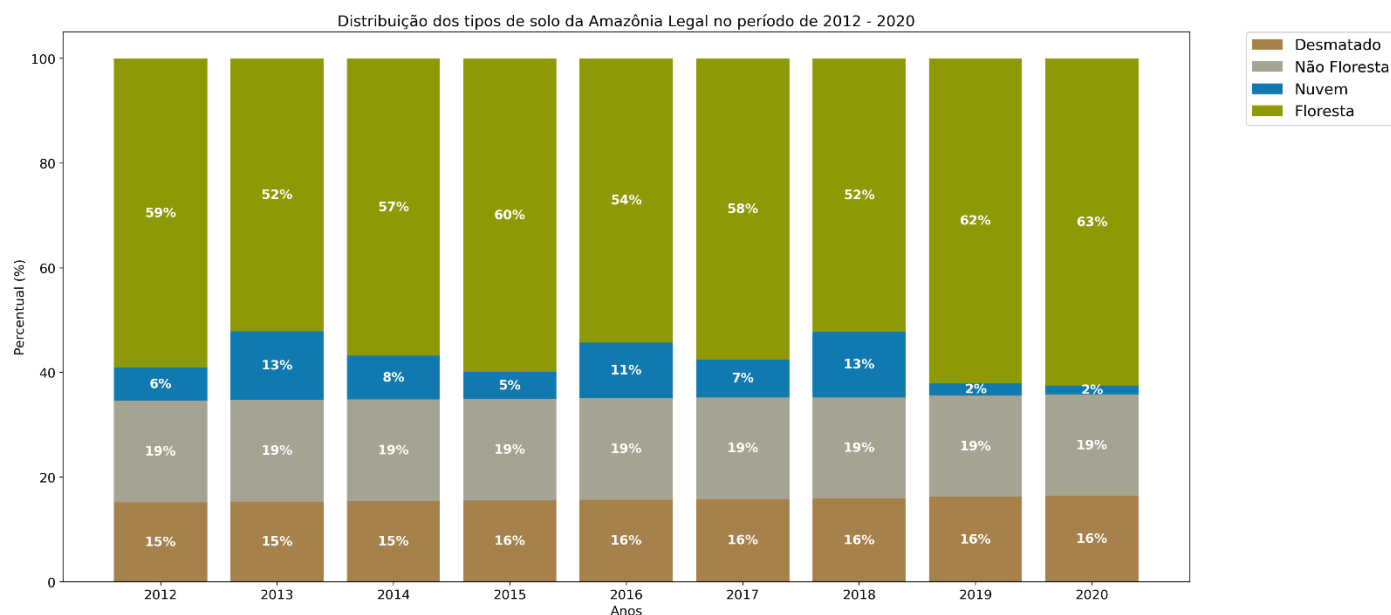
Observando as principais variáveis, temos na base 15.960 registros de informações sobre os municípios dos Estados da Amazônia Legal (Acre, Amapá, Amazonas, Maranhão, Mato Grosso, Pará, Roraima, Rondônia, Tocantins). O maior município tem uma área total de quase 160mil km², enquanto o menor tem uma área de 64 km². Os municípios tem em média 3.971 km² de floresta e 949 km² desmatados.

- Gráfico 1

O primeiro gráfico criado mostra a distribuição dos tipos de solo da Amazônia Legal no período de 2012 até 2020. Para todas as análises, foi utilizado um período mais recente devido a uma maior consistência nos dados. Em períodos mais antigos, foram notados erros nas medições, como por exemplo, a soma de todas as áreas do município ser maior do que a área total dele. Isto não ocorreu para as informações mais recentes. Além disto, só foram utilizadas as informações de áreas Desmatadas, Não Floresta, Floresta e de Nuvens, pois estas são mais significantes para o entendimento sobre a evolução do desmatamento.

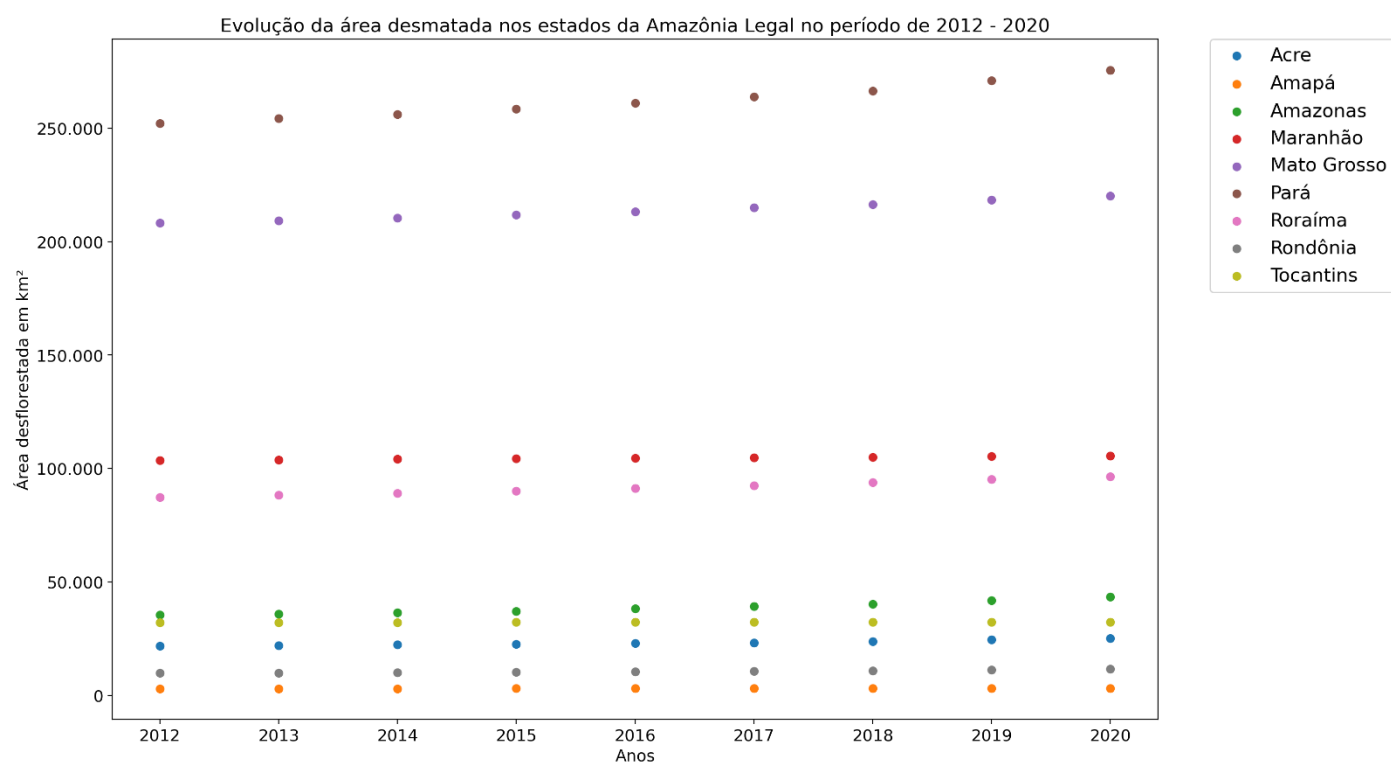
Observando o gráfico, é notado que em percentual, o tamanho da área desmatada cresceu pouco nos últimos anos. O que chama mais atenção é a variação no tamanho da área de Floresta e da área de Nuvens. Isto ocorre devido ao tipo de obtenção destes dados, que são colhidos via

imagens de satélites. Se no momento da captura da imagem existe um acúmulo de nuvens no local, não é possível saber se aquela área é de floresta, de desmatamento ou de não floresta.



- Gráfico 2

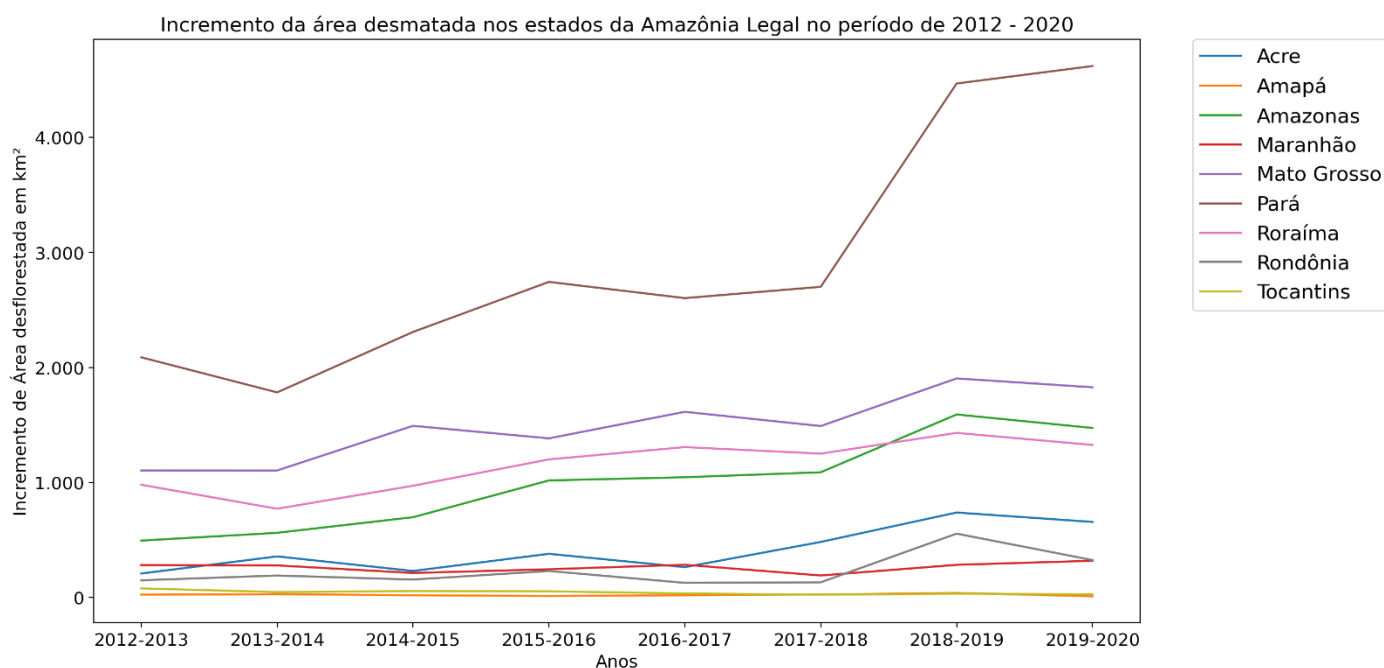
O segundo gráfico mostra a evolução da área desmatada nos estados da Amazônia Legal no período de 2012 até 2020. Nele podemos ver que os estados do Pará e do Mato Grosso possuem a maior área desflorestada / desmatada e esta segue em crescimento ao longo dos anos, principalmente no ano de 2020.



- Gráfico 3

O terceiro gráfico mostra o incremento da área desmatada nos Estados da Amazônia Legal no período de 2012 até 2020. Este indicador mostra a variação incremental da área desmatada em km² em relação a cada ano. Com este cálculo, é possível verificar quais estados estão tendo maiores taxas de crescimento nos indicadores de desflorestamento.

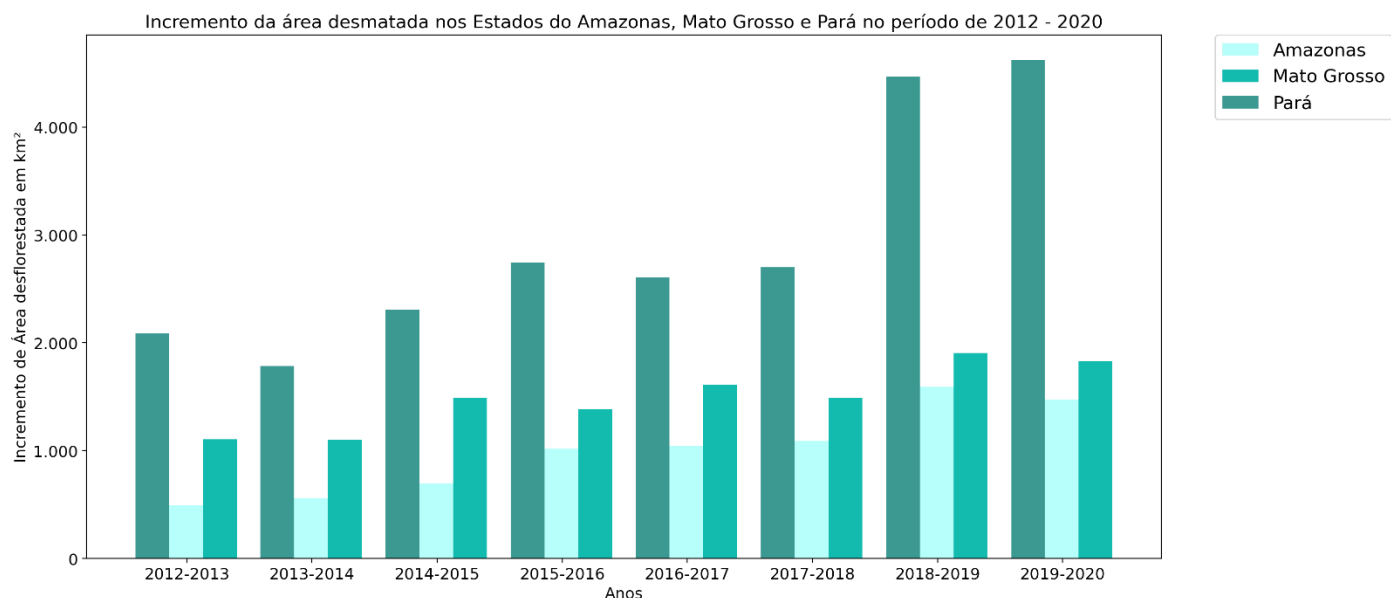
Vemos que o estado do Pará sofreu um crescimento expressivo comparando a área desmatada do ano de 2018 com o ano de 2019. A extensão praticamente dobrou. Além disso, os estados de Mato Grosso e do Amazonas também merecem uma observação mais detalhada nos seus índices.



- Gráfico 4

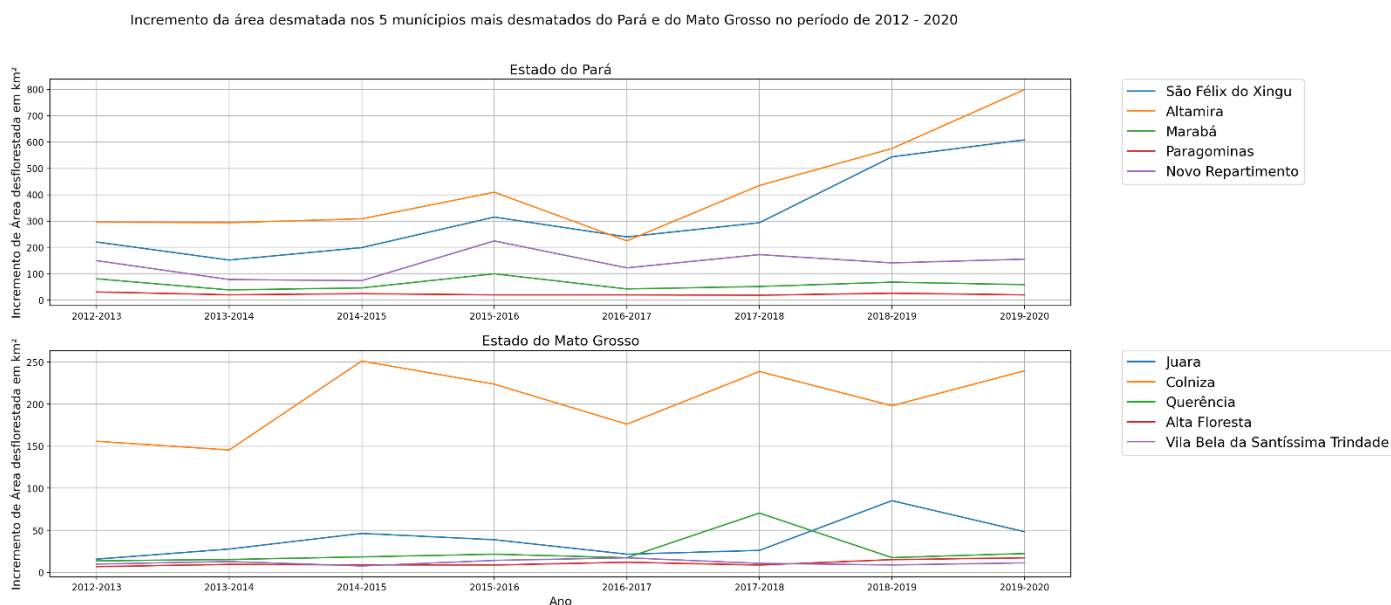
O quarto gráfico mostra o incremento da área desmatada nos estados do Amazonas, Mato Grosso e Pará no período de 2012 até 2020. Estes três estados foram os que mais tiveram incremento na área desmatada no ano de 2020.

Podemos observar que o incremento anual no desflorestamento no estado do Amazonas está crescendo num ritmo bem acelerado e o mesmo está chegando perto dos índices do estado do Mato Grosso.



- Gráfico 5 e 6

Os gráficos 5 e 6 mostram o incremento no período de 2012 até 2020 da área desmatada nos cinco municípios dos estados do Pará e do Mato Grosso que tem a maior área total desflorestada. No estado do Pará, os municípios que mais estão sofrendo com os desflorestadores são: Altamira e São Félix do Xingu. Estes tiveram um crescimento significativo nos últimos anos. Já no estado do Mato Grosso, um município domina o ranking de incremento de área desmatada: Colniza, chegando a ter um aumento de quase 250 km² no ano de 2020.



3. Preparação Pedagógica

PLANO DE AULA

IDENTIFICAÇÃO

Disciplina: Análise de dados com Python

Professor: Bruna Mattioli de Oliveira e Gabriel Andrade Varga

TEMA DA AULA

Matplotlib - Gráficos de barras e subplots. Tema: Desmatamento.

PRÉ-REQUISITOS

Biblioteca Panda e comandos básicos da linguagem Python.

OBJETIVOS

Habilitar os alunos a criar gráficos no Matplotlib e realizar análises baseadas nesses gráficos.

CONTEÚDO PROGRAMÁTICO - APRESENTAÇÃO

Importar a biblioteca Matplotlib

Inserir os dados já tratados nos diferentes tipos de gráficos utilizando o Matplotlib

Explorar as principais funções da biblioteca

Apresentar o site da documentação

DESENVOLVIMENTO DO TEMA

Na aula será utilizado o Dataset de Desmatamento disponibilizado pelo professor, e a aula apresentará como deve ser feita a inserção dos dados já tratados anteriormente em gráficos pela biblioteca Matplotlib.

A aula será desenvolvida a partir da utilização de exemplos e o professor auxiliará os alunos para que estes criem seus próprios gráficos.

RECURSOS DIDÁTICOS

Projektor, computador, software Anaconda.

AVALIAÇÃO

Ao final da aula os alunos devem entregar 2 gráficos e a análise de dois cenários. A entrega valerá 1 ponto na média do aluno e avaliará a quantidade de elementos inseridos e como o gráfico contribuiu para a análise.

BIBLIOGRAFIA

HUNTER, J. D. "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007. Disponível em: <<https://matplotlib.org/>>. Acesso em 16/07/2021.

NELLI, Fabio. "Python Data Analytics", 2018. Second Edition.

YIM, Aldrin, CHUNG, Claire, YU, Allen. "Matplotlib for python developers". 2018. Second Edition.

PLANO DE ENSINO

CURSO	Especialização em Ciência de Dados
--------------	------------------------------------

DISCIPLINA/UNIDADE CURRICULAR	CÓDIGO	PERÍODO	CARGA HORÁRIA (qtde. aulas)
Análise de dados com Python	2	Matutino	20

PRÉ-REQUISITO	Lógica de Programação
----------------------	-----------------------

OBJETIVOS

Ensino da linguagem Python, formas de utilizar, principais comandos e particularidades. Habilitar os alunos a realizarem análises utilizando o Python e suas principais bibliotecas, Pandas e Matplotlib. Demonstrar uma análise de dados com dados reais desde a inserção dos dados no Python, o tratamento destes e análise.

EMENTA

Instalação Software Anaconda. Principais comandos Python. Tipos de dados do Python. Estruturas de agrupamento de dados. Tratamento de dados com Pandas. Gráfico de Barras com Matplotlib. Gráfico de Linha com Matplotlib. Gráfico de subplots com Matplotlib. Análise de Dados com Python.

CONTEÚDO PROGRAMÁTICO

ITEM	EMENTA	CONTEÚDO
1	Instalação Software Anaconda. Principais comandos Python. Tipos de dados do Python.	- Formas de utilizar o Python: Software Anaconda, Criação do Ambiente, Jupyter notebook. - Comandos: Mean, sum, print, input, for, while, if, else, etc. - Tipos de dados. Tipagem dos dados Int, Str, Datetime, etc.

		- Estruturas de agrupamento de dados. Listas, dicionários, container, etc.
2	Tratamento de dados com Pandas.	- Dataframes: Comandos head, tail, describe, etc. Adicionar colunas, Cálculos acumulados, Agrupamento de dados.
3	Gráfico de Barras com Matplotlib. Gráfico de Linha com Matplotlib. Gráfico de subplots com Matplotlib.	- Gráficos: Eixo X, Eixo Y, Subplot, Legendas, Rótulos, Edição dos eixos. - Gráfico de barras: Barras agrupadas, Barras não agrupadas. - Gráfico de linhas: Gráfico de 1 linhas Gráfico de duas ou mais linhas - Subplots: Dividindo o eixo X, Dividindo o eixo Y.
4	Ferramentas de análise de dados.	Tipos de gráficos para cada análise, Formas de análise, Introdução ao Storytelling.
5	Análise de dados de um caso real.	Análise do dataset real com tema Desmatamento.

--

PROFESSOR	TURMA
Bruna Mattioli de Oliveira e Gabriel Andrade Varga	5

PROGRAMAÇÃO E CONTEÚDOS DAS AULAS (PREVISÃO)		
Dia/Mês ou Semana	Conteúdo das Aulas	Número de Aulas
02/08	Como instalar o Software Anaconda e montar o ambiente para o Python.	2
04/08	Principais comandos e tipos de dados do Python.	2
09/08	Principais bibliotecas do Python.	2
11/08	Tratamento de dados com Pandas. Tema: Desmatamento.	2
16/08	Matplotlib - Gráficos de linhas. Tema: Desmatamento.	2
18/08	Matplotlib - Gráficos de barras e subplots. Tema: Desmatamento.	2
23/08	Análise dos dados de uma base real. Tema: Desmatamento. Parte 1	2
25/08	Análise dos dados de uma base real. Tema: Desmatamento. Parte 2	2
30/08	Avaliação	2
01/09	Exame	2

PROCEDIMENTOS DE ENSINO
AULAS TEÓRICAS
<p>Aulas expositivas dialogadas para apresentação e discussão dos conceitos com a utilização de exemplos.</p> <p>As aulas expositivas serão apoiadas por recursos didáticos e utilização de bases de dados reais para análise.</p>
AULAS PRÁTICAS
<p>Todas as aulas terão uma parte prática e deverão ser realizadas com a utilização de um computador (podendo ser os computadores do laboratório) para que o aluno possa acompanhar o professor na execução dos exemplos.</p>
PROCEDIMENTOS DE AVALIAÇÃO
<p>Média = trabalho*0,4 + avaliação*0,5 + entregas * 0,1</p> <p>Caso a média do aluno fique acima de 3 e menor que 6, o aluno poderá realizar o exame.</p> <p>A soma da média anterior com o exame será dividida por 2 para chegar a média final.</p> <p>Para aprovação, o aluno deve ter média final de ao menos 6 pontos e frequência de 70% nas aulas.</p>

REFERÊNCIAS
Referências Básicas: NELLI, Fabio. "Python Data Analytics", 2018. Second Edition. YIM, Aldrin, CHUNG, Claire, YU, Allen. "Matplotlib for python developers". 2018. Second Edition.
Referências Complementares: HUNTER, J. D. "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007. Disponível em: < https://matplotlib.org/ >. Acesso em 16/07/2021.

ORIENTAÇÕES GERAIS
Em caso de não comparecimento na avaliação, a segunda chamada será aplicada no mesmo dia do exame, mediante encaminhamento da coordenação.

Assinatura do Professor

Assinatura do Coordenador do Curso