

HELLEM CRISTINA TEIXEIRA RODRIGUES  
MILENA MARÍLIA NOGUEIRA DE ANDRADE

**DETECÇÃO E  
MAPEAMENTO  
AUTOMÁTICOS DE ÁREAS  
AFETADAS PELO FOGO NO  
ESTADO DO PARÁ: SCRIPT  
DO APLICATIVO DO  
GOOGLE EARTH ENGINE**

Belém-Pará  
2022



UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE GEOCIÊNCIAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM GESTÃO DE RISCOS  
E DESASTRES NA AMAZÔNIA

Produto Técnico vinculado a Dissertação apresentada ao Programa de Pós-Graduação em Gestão de Riscos e Desastres na Amazônia, do Instituto de Geociências da Universidade Federal do Pará, em cumprimento às exigências para obtenção do título de Mestre em Gestão de Riscos e Desastres Naturais na Amazônia.



**DETECÇÃO E MAPEAMENTO AUTOMÁTICOS  
DE ÁREAS AFETADAS PELO FOGO NO  
ESTADO DO PARÁ**

**SCRIPT DO APLICATIVO DO GOOGLE EARTH ENGINE**

Hellem Cristina Teixeira Rodrigues  
Milena Marília Nogueira de Andrade

PROGRAMA DE PÓS-GRADUAÇÃO EM GESTÃO DE  
RISCOS E DESASTRE NA AMAZÔNIA  
BELÉM – PARÁ  
2022

## CONSIDERAÇÕES INICIAIS

- Esse Script acompanha metodologias recomendadas e segue fielmente classificações pré-estabelecidas por autores consolidados, mas cabe destacar que, é possível fazer adaptações e inclusão de novas funções, por isso, o script foi disponibilizado na sua integralidade;
- Cada função está composta de comentário autoexplicativo, visando auxiliar o usuário no entendimento da mesma, assim como, possui referências a trabalhos utilizados, que foram as bases conceituais e podem ser consultadas em caso de dúvidas também;
- Por se tratar de um aplicativo atemporal, a precisão dos dados mapeados pode variar de acordo com período e região, cada novo mapeamento exige nova avaliação de acurácia, sendo recomendado que os valores de referencias sejam coletados por especialista com uso de imagem de melhor resolução espacial, se possível, uma vez que, os dados oficiais de referencias do Brasil (a saber: BDQueimadas) possuem em média erro de 1.000 metros no mapeamento;
- Caso tenha interesse em se aprofundar em aplicativos em nuvem, o Sr. Samapriya Roy desenvolveu um aplicativo chamado “**ee-appshot**” que realiza o download de todos os aplicativos já criados na plataforma do GEE, possibilitando que o usuário estude a partir de implementações já aplicadas e testadas, disponível em: <https://github.com/samapriya?tab=repositories>;
- Espero que este trabalho contribua para o seu desenvolvimento e que isso ajude a torna o mundo um lugar melhor.

```
//HELLEM CRISTINA TEIXEIRA RODRIGUES©
// Detecção adaptada do workshop Serviço florestal dos EUA
//Adaptado do workshop "Gerenciamento de Incêndios Florestais por meio do Uso da
Plataforma Google Earth Engine (GEE)"
//https://www.youtube.com/watch?v=xvP6MzS1E3I&t=341s - 12/03/2021 com 1:56:01 horas
//Instituto de Altos Estudios Espaciales Mario Gulich (CONAE/UNC) – Argentina
// Tabela de classificação para interpretar a Severidade da queimada - United States
Geological Survey (USGS)
```

```
////////////////////////////////////
//          1) Importa as Layers de Interesse          //
////////////////////////////////////
// import tools
var tools = require('users/fitoprincipe/geetools:tools');
var cities = ee.FeatureCollection("users/hellemcristinat/Municipios_PA");
var city = ee.FeatureCollection("users/hellemcristinat/Municipios_PA");
var mask = ee.Image("users/hellemcristinat/Mapbiomas_mascara_11_12_23_33");
var HRSL = ee.ImageCollection("projects/sat-io/open-datasets/hrsl/hrslpop");
var palettes = require('users/gena/packages:palettes');
var style = require('users/gena/packages:style');
var logo = ee.Image("users/hellemcristinat/Brasao_UPA");

////////////////////////////////////
//  2) Começa a configuração da aparência do mapa e do App de Layers  //
////////////////////////////////////
//          SEVERIDAD DE INCENDIO - CASO PARÁ
Map.setOptions('TERRAIN');//Database:HYBRID, ROADMAP, SATELLITE, TERRAIN
//Área de estudo sem preenchimento
var cityoutline_obj = ui.data.ActiveList();
var cityoutline = ee.Image().byte().paint({
  featureCollection: cities,
  color: '000000',
  width: 1
});
cityoutline_obj.add(cityoutline);
Map.addLayer(cityoutline, {palette: 'black'}, 'Municípios');

////////////////////////////////////
//  3) Configuração painéis e widgets para exibição  //
////////////////////////////////////

var panel = ui.Panel({style: {width:'25.000%'}});

var Logo = ui.Thumbnail({
  image: logo,
  params: {
    bands:['b1','b2','b3'],
    min:0,
```

```

    max:255,
    position: 'top-center'},
    style: {
        width:'63px',
        height:'83px'
    }
});

```

```
// Centralizando a Logo
```

```

var Info = ui.Panel({
    style: {
        margin:'2px 2px 2px 150px'
    },
    layout: ui.Panel.Layout.flow('horizontal')
});
Info.add(Logo);
panel.add(Info);

```

```
// Definindo Titulo e subtítulo
```

```

panel.add(ui.Label({
    value: 'Detecção e Mapeamento de Queimadas e Incêndios Florestais - V.1.0',
    style:{
        fontWeight: 'bold',
        fontSize: '24px',
        margin: '10px 2px 2px 10px',
        textAlign:'center'
    }
}));

```

```

var subtitle = ('Mestrado Profissional em Gestão de Riscos e Desastres Naturais na Amazônia
- Universidade Federal do Pará (UFPA)');

```

```
// Adicionando descrição
```

```

panel.add(ui.Label({
    value: subtitle,
    style: {
        fontWeight: 'bold',
        fontSize: '16px',
        margin:'2px 10px 10px 10px',
        textAlign:'justify'
    }
}));

```

```
var subtitle1 = ('Autora: Hellem Cristina Teixeira Rodrigues');
```

```
// Adicionando Autoria
```

```

panel.add(ui.Label({
    value: subtitle1,
    style: {
        fontWeight: 'bold',
        fontSize: '16px',
        margin:'2px 10px 10px 10px',
        textAlign:'justify'
    }
}));

```

```

));

var subtitle2 = ('Co-autora: Milena Marília Nogueira de Andrade');
// Adicionando Co-autoria
panel.add(ui.Label({
  value: subtitle2,
  style: {
    fontWeight: 'bold',
    fontSize: '16px',
    margin: '2px 10px 10px 10px',
    textAlign: 'justify'}
}));

//*****
//                               SELECIONE UMA PLATAFORMA DE SATÉLITE

// Você pode selecionar imagens de dois sensores de satélite disponíveis.
// Considere os detalhes de cada missão abaixo para escolher os dados certos para
// suas necessidades.

// LANDSAT 8                               | SENTINEL-2 (A & B)
//-----
// Lançado:      11/02/2013                | 23/06/2015 & 07/03/2017
// Revisita:     16 dias                    | 5 dias (desde 2017)
// Resolução:    30 metros                 | 10 metros
// Vantagens:    Largas series temporal    | 9 series de alto detalhes espacial
// Exporta arquivos pequenos              | Maior probabilidade de imagens sem nuvens

var DataSelectorLabel=ui.Label('1: Escolha o satélite :',{
  fontWeight: 'bold',
  margin: '2px 10px 10px 10px'
});

// ----- datasetSelector -----
var Data = {
  'Sentinel 2': {
    dataset: "COPERNICUS/S2",
    layer_name_prefix: "SENTINEL 2"},
  'Landsat 8': {
    dataset: "LANDSAT/LC08/C02/T1_L2",
    layer_name_prefix: "LANDSAT 8"},
};
var selectedDataset_obj= ui.data.ActiveList();
var datasetselector_obj = ui.data.ActiveList();
var datasetSelector = ui.Select({
  items:Object.keys(Data),
  placeholder:"Selecione o satélite ...",
  onChange: function(selected){
    var selectedDataset = (Data[selected]['dataset']); //ImCol //satelite
    print(selectedDataset);
  }
});

```

```

        selectedDataset_obj.add(selectedDataset);
    }
});
datasetselector_obj.add(datasetSelector);

////////////////////////////////////
//4) Escolha da área de estudo - municípios do estado do Pará //
////////////////////////////////////

// Dropdowns vazios
var citiesDD = ui.Select([], 'Carregando..');
var cityDD = ui.Select([], 'Carregando..');
// Carregando Estado
var citiesNames = cities.aggregate_array('nm_municip');

function getcity (cite){
// Dado um estado obtém todos os municípios
var feat = ee.Feature(cities.filterMetadata('nm_municip', 'equals', cite).first());

var citiefp = ee.String(feat.get('nm_municip'));

var filteredcity = city.filterMetadata('nm_municip', 'equals', citiefp);
var filteredcityNames = filteredcity.aggregate_array('nm_municip');
return ee.List(filteredcityNames);
}

citiesNames.evaluate(function (cities){
    citiesDD.items().reset(cities);
    citiesDD.setPlaceholder('Selecione o município');
    citiesDD.onChange(function(cite){
// depois de selecionar um estado (onChange) obtém todos os condados e preenche a lista
suspensa
        cityDD.setPlaceholder('Carregando..');
        var city = getcity(cite);
        city.evaluate(function(cityNames){
            cityDD.items().reset(cityNames);
            cityDD.setPlaceholder("Zoom no município..");

        });
    });
});

var area_escolhido_obj = ui.data.ActiveList();
var city_name_obj = ui.data.ActiveList();
cityDD.onChange(function(name){
    var Escolhido = ee.Feature(city.filterMetadata('nm_municip', 'equals', name).first());
    Map.addLayer(Escholhido, { }, name);
    Map.centerObject(Escholhido, 8); //Escolhido == Area

```



```

city_name_obj.add(name);
area_escolhido_obj.add(Escolhido);

//RunSelectedImageCollection();
});

var labelInteresse = ui.Label('2: Escolha o município de interesse :', {
  fontWeight: 'bold',
  margin:'2px 10px 10px 10px'
});

var labelInteresseZoom = ui.Label('3: Zoom no município escolhido :', {
  fontWeight: 'bold',
  margin:'2px 10px 10px 10px'
});

//Adicionando zoom para a área de estudo - Municípios do estado do Pará
var Zoom = ui.Button({
  style:{position: 'top-center',
  border: '1px solid #000000',
  width: '350px',
  fontSize: '20px'},
  label: 'Zoom no Pará',
  onClick: function(){Map.setCenter(-52.734, -3.569,6);}
});

////////////////////////////////////
//      5) Determinação do período de análise          //
////////////////////////////////////
// Defina as datas de início e término de um período ANTES e DEPOIS do incêndio.
Certifique-se de
// que o espaço temporal é grande o suficiente para adquirir uma imagem Sentinel-2 (revisita
= 5 dias)
// ou imagem Landsat 8 (revisita = 16 dias).

var labelPeriodo = ui.Label('4: Selecione o período de análise :', {
  fontWeight: 'bold',
  margin:'2px 10px 10px 10px'
});

//Data inicial antes do fogo
var labelPeriodoT0 = ui.Label('4.1 Data inicial antes do fogo');

var prefire_start = ee.Date('2014-01-01')
var prefire_end = ee.Date(new Date().getTime())

// Agora defina os mesmos parâmetros para APÓS o incêndio.
var postfire_start = ee.Date('2014-01-01')

```

```

var postfire_end = ee.Date(new Date().getTime())
var dates_s_prefire_start = ee.Dictionary({start: prefire_start,end: prefire_end});
dates_s_prefire_start.evaluate(
  function(dates_c){

    var start_value = dates_c.start.value;
    var end_value = dates_c.end.value;

    var textBoxT0 = ui.DateSlider({
      start: start_value,
      end: end_value,
      onChange: renderDateRangePrefireStart,
      value: end_value, // Checkar
      'style': {
        'width': '350px',
      },
    });

    panel.add(labelPeriodoT0);
    panel.add(textBoxT0);

  });

var prefire_start_obj= ui.data.ActiveList();
function renderDateRangePrefireStart(dateRange) {
//print("4.1 Data inicial antes do fogo: ",dateRange);
  prefire_start_obj.add(dateRange);
}

//Data final antes do fogo
var labelPeriodoT1 = ui.Label('4.2 Data final antes do fogo ');

var dates_s_prefire_end = ee.Dictionary({start: prefire_start,end: prefire_end});
dates_s_prefire_end.evaluate(
  function(dates_c){

    var start_value = dates_c.start.value;
    var end_value = dates_c.end.value;

    var textBoxT1 = ui.DateSlider({
      start: start_value,
      end: end_value,
      onChange: renderDateRangePrefireEnd,
      value: end_value, // Checkar
      'style': {
        'width': '350px',

```

```

    },
  });

  panel.add(labelPeriodoT1);
  panel.add(textBoxT1);

});

var prefire_end_obj= ui.data.ActiveList();
function renderDateRangePrefireEnd(dateRange) {
  //print("4.2 Data final antes do fogo: ",dateRange);
  prefire_end_obj.add(dateRange);
}

//Data inicial depois do fogo
var labelPeriodoT2 = ui.Label('4.3 Data inicial depois do fogo');
var dates_s_posfire_start = ee.Dictionary({start: postfire_start,end: postfire_end});
dates_s_posfire_start.evaluate(
  function(dates_c){

    var start_value = dates_c.start.value;
    var end_value = dates_c.end.value;

    var textBoxT2 = ui.DateSlider({
      start: start_value,
      end: end_value,
      onChange: renderDateRangePosfireStart,
      value: end_value, // Checkar
      'style': {
        'width': '350px',
      },
    });

    panel.add(labelPeriodoT2);
    panel.add(textBoxT2);

  });

var posfire_start_obj= ui.data.ActiveList();
function renderDateRangePosfireStart(dateRange) {
  //print("4.3 Data inicial depois do fogo: ",dateRange);
  posfire_start_obj.add(dateRange);
}

//Data final depois do fogo
var labelPeriodoT3 = ui.Label('4.4 Data final depois do fogo');
```

```

var dates_s_posfire_end = ee.Dictionary({ start: postfire_start,end: postfire_end});
var btn_exec_obj = ui.data.ActiveList();
var btn_lim_obj = ui.data.ActiveList();
var btn_exp_dnbr_obj = ui.data.ActiveList();
var btn_exp_shp_obj = ui.data.ActiveList();
dates_s_posfire_end.evaluate(
  function(dates_c){

    var start_value = dates_c.start.value;
    var end_value = dates_c.end.value;

    var textBoxT3 = ui.DateSlider({
      start: start_value,
      end: end_value,
      onChange: renderDateRangePosfireEnd,
      value: end_value, // Checkar
      'style': {
        'width': '350px',
      },
    });

    panel.add(labelPeriodoT3);
    panel.add(textBoxT3);

    ////////////////////////////////////////////////////
    //          6) Botões de execução          //
    ////////////////////////////////////////////////////
    //Botão dNBR
    var btn_executar = ui.Button({
      style:{position: 'top-center',
      border: '1px solid #000000',
      width: '350px',
      fontSize: '20px'},
      label: 'Executar',
      onClick: function(){

        RunSelectedImageCollection();
      }
    }).setDisabled(true);
    panel.add(btn_executar);
    btn_exec_obj.add(btn_executar);

    //Botão Limpar
    var btn_limpar = ui.Button({
      style:{position: 'top-center',
      border: '1px solid #000000',
      width: '350px',
      fontSize: '20px'},

```

```

label: 'Limpar',
onClick: function(){

    LimparSelectedImageCollection();
}
}).setDisabled(true);
panel.add(btn_limpar);
btn_lim_obj.add(btn_limpar);

//Botão para exportar o dNBR
var btn_exportar_drive_dnbr = ui.Button({
    style:{ position: 'top-center',
border: '1px solid #000000',
width: '350px',
fontSize: '20px'},
    label: 'Exportar dNBR',
    onClick: function(){
        Export_dNBR_ToDrive();
    }
}).setDisabled(true);
panel.add(btn_exportar_drive_dnbr);
btn_exp_dnbr_obj.add(btn_exportar_drive_dnbr);

//Botão para exportar o mapeamento da queimada
var btn_exportar_drive_shp = ui.Button({
    style:{ position: 'top-center',
border: '1px solid #000000',
width: '350px',
fontSize: '20px'},
    label: 'Exportar KML',
    onClick: function(){
        Export_SHP_ToDrive();
    }
}).setDisabled(true);
panel.add(btn_exportar_drive_shp);
btn_exp_shp_obj.add(btn_exportar_drive_shp);

////////////////////////////////////
//          7) Mais informações          //
////////////////////////////////////
//Adicionando URL da dissertação
var URL = ui.Label({
    value: "Publicação PPGGRD",
    targetUrl:'https://ppggrd.propesp.ufpa.br/index.php/br/',
    style: {margin:'0px 10px 0px 2px' }});

//Adicionando o e-mail para contato
var e_mail = ui.Label({
    value: 'Hellem Rodrigues',
    targetUrl:'mailto:hellem.cristinat@gmail.com',

```

```

    style: {margin:'0px 10px 0px 2px' }}));

    var Info = ui.Panel({style:{margin:'10px 2px 10px 10px'}, layout:
ui.Panel.Layout.flow('horizontal')});
    Info.add(ui.Label({value: 'Mais informações:',style: {fontWeight: 'bold',margin:'0px 0px
0px 2px'}}));
    Info.add(URL);
    Info.add(ui.Label({value: 'Contato:',style: {fontWeight: 'bold',margin:'0px 2px 0px
10px'}}));
    Info.add(e_mail);

    panel.add(Info);
    panel.add(ui.Label({value: "Fontes de dados:",
        style:{margin:'0px 2px 2px 10px', fontSize: '14px',fontWeight: 'bold'}}));
    panel.add(ui.Label({value: "Limites Municipais: IBGE (2015) & Máscara: Projeto
TerraClass (2014)",
        style:{margin:'2px 2px 2px 10px', fontSize: '12px'}}));

});

var posfire_end_obj= ui.data.ActiveList();
function renderDateRangePosfireEnd(dateRange) {
//print("4.4 Data final depois do fogo: ",dateRange);
posfire_end_obj.add(dateRange);

var btn_exec = btn_exec_obj.get(0);
btn_exec.setDisabled(false);
}

function LimparSelectedImageCollection(){
var btn_exec = btn_exec_obj.get(0);
btn_exec.setDisabled(true);

var btn_limp = btn_lim_obj.get(0);
btn_limp.setDisabled(true);

var btn_exp_dnbr_ = btn_exp_dnbr_obj.get(0);
btn_exp_dnbr_.setDisabled(true);

var btn_exp_shp_ = btn_exp_shp_obj.get(0);
btn_exp_shp_.setDisabled(true);

removeLayer('Mapeamento');
removeLayer('smooth');
removeLayer('Threshold');
removeLayer('dnBR classificado');
removeLayer('dnBR em escala de cinza');
removeLayer('Pos-tfire NBR');
removeLayer('Pre-fire NBR');

```

```

removeLayer('Img pos-incêndio: msk');
removeLayer('Img pre-incêndio: msk');
removeLayer('Img pos-incêndio');
removeLayer('Img pre-incêndio');
removeLayer('Área de estudo');
var city_name_ =city_name_obj.get(0);
removeLayer(city_name_);
city_name_obj.reset();

var legend_ = legend_obj.get(0);
Map.remove(legend_);
legend_obj.reset();

//selectedDataset_obj.reset();

}

function removeWidgets(name) {
  var widgets = Map.widgets();
  var names = [];
  widgets.forEach(function(wid){
    var wid_name = wid.getName();
    names.push(wid_name);
  });
  var index = names.indexOf(name);
  if (index > -1){
    var widget = widgets.get(index);
    Map.remove(widget);
  }
  else{
    print('widget '+name+' not found');
  }
}

function removeLayer(name) {
  var layers = Map.layers();
  // list of layers names
  var names = [];
  layers.forEach(function(lay) {
    var lay_name = lay.getName();
    names.push(lay_name);
  });
  // get index
  var index = names.indexOf(name);
  if (index > -1) {
  // if name in names
    var layer = layers.get(index);
    Map.remove(layer);
  } else {
    print('Layer '+name+' not found');
  }
}

```

```

    }
}

```

```

var dNBR_obj= ui.data.ActiveList();
var fc_obj = ui.data.ActiveList();
var legend_obj = ui.data.ActiveList();
function RunSelectedImageCollection(){

```

```

//=====
=====

```

```

//Localização

```

```

    var area_escolhida = area_escolhido_obj.get(0);
    var area_escolhida_geometry = area_escolhida.geometry();

```

```

//Condições de funcionalidades

```

```

    var btn_exec = btn_exec_obj.get(0);
    btn_exec.setDisabled(true);

```

```

    var btn_limp = btn_lim_obj.get(0);
    btn_limp.setDisabled(false);

```

```

    var btn_exp_dnbr_ = btn_exp_dnbr_obj.get(0);
    btn_exp_dnbr_.setDisabled(false);

```

```

    var btn_exp_shp_ = btn_exp_shp_obj.get(0);
    btn_exp_shp_.setDisabled(false);

```

```

    var prefire_start_date = prefire_start_obj.get(0);
    var prefire_end_date = prefire_end_obj.get(0);
    var prefire_start_full_date = prefire_start_date.start().format("y-MM-d");
    var prefire_end_full_date = prefire_end_date.start().format("y-MM-d");

```

```

//Adicionando ao console

```

```

    print("prefire_start: ",prefire_start_full_date);
    print("prefire_end: ",prefire_end_full_date);

```

```

    var posfire_start_date = posfire_start_obj.get(0);
    var posfire_end_date = posfire_end_obj.get(0);
    var posfire_start_full_date = posfire_start_date.start().format("y-MM-d");
    var posfire_end_full_date = posfire_end_date.start().format("y-MM-d");

```

```

//Adicionando ao console

```

```

    print("posfire_start: ",posfire_start_full_date);
    print("posfire_end: ",posfire_end_full_date);

```

```

//===== Selecciona imagens do Satélite por Tempo e Área

```

```

// Nas linhas a seguir, as imagens serão agrupadas em como uma ImageCollection, atendendo
// aos criterios de localização já definidos, período de análise e aplicado a função .sorte
// para cobertura de nuvens, sendo uma função que apresenta em ordem crescente as imagens

```



```

// com menor percentual de nuvens
var selectedDataset = selectedDataset_obj.get(0);
print("selectedDataset: ",selectedDataset);

var imagery = ee.ImageCollection(selectedDataset);

if (selectedDataset == 'COPERNICUS/S2' ) {

    var prefireSelectDataset = ee.ImageCollection(imagery
//Filtra por Tempo
    .filterDate(prefire_start_full_date,prefire_end_full_date)
//Filtra por Área
    .filterBounds(area_escolhida_geometry)
//Implementação para imagem mais livre de nuvem
    .sort('CLOUDY_PIXEL_PERCENTAGE',false));
//mostrar no console a coleção de imagens que atendem ao criterio
    print("Coleção de imagens pré-incêndio: ", prefireSelectDataset );

    var postfireSelectDataset = ee.ImageCollection(imagery
// Filtrar por tempo
    .filterDate(posfire_start_full_date, posfire_end_full_date)
// Filtrar por área
    .filterBounds(area_escolhida_geometry)
//Implementação para imagem mais livre de nuvem
    .sort('CLOUDY_PIXEL_PERCENTAGE',false));
//mostrar no console a coleção de imagens que atendem ao criterio
    print("Coleção de imagens pós-incêndio: ", postfireSelectDataset);

} else {
    var prefireSelectDataset = ee.ImageCollection(imagery
//Filtra por Tempo
    .filterDate(prefire_start_full_date,prefire_end_full_date)
//Filtra por Área
    .filterBounds(area_escolhida_geometry)
//Implementação para imagem mais livre de nuvem
    .sort('CLOUD_COVER_LAND',false));
//mostrar no console a coleção de imagens que atendem ao criterio
    print("Coleção de imagens pré-incêndio: ", prefireSelectDataset);

    var postfireSelectDataset = ee.ImageCollection(imagery
// Filtrar por tempo
    .filterDate(posfire_start_full_date, posfire_end_full_date)
// Filtrar por área
    .filterBounds(area_escolhida_geometry)
//Implementação para imagem mais livre de nuvem
    .sort('CLOUD_COVER_LAND',false));
//mostrar no console a coleção de imagens que atendem ao criterio
    print("Coleção de imagens pós-incêndio: ", postfireSelectDataset);
}
}

```

```

// Aplicar máscara de nuvem específica da plataforma
if (selectedDataset == 'COPERNICUS/S2' ) {
    var prefire_CM_SelectDataSet = prefireSelectDataset.map(maskS2sr);
    var postfire_CM_SelectDataSet = postfireSelectDataset.map(maskS2sr);
} else {
    var prefire_CM_SelectDataSet = prefireSelectDataset.map(maskL8sr);
    var postfire_CM_SelectDataSet = postfireSelectDataset.map(maskL8sr);
}

//----- Mosaico e recorte de imagens para a área de estudo -----

// Isso é especialmente importante para o desenvolvimento do trabalho, visto que encontram-
se inumeras imagens e
// precisa-se transformar a coleção em uma única imagem, por isso, implementou-se a função
de mosaico,
//permitindo a aplicação de todos os processamentos.

    var pre_mos = prefireSelectDataset.mosaic().clip(area_escolhida_geometry);
    var post_mos = postfireSelectDataset.mosaic().clip(area_escolhida_geometry);

//Como o índice NBR é bem similar a outros índices, como o NDWI, a classificação de
severidade de queimadas
//também classifica a água, para evitar este equívoco, optou-se pela aplicação de máscara de
exclusão para
//este trabalho, sendo aplicado a classe 5.1 rio, lago e oceano do projeto Mapbiomas - coleção
6 - ano de 2020
// No entanto, observou-se que os campos alagados também apresentavam variação e por isso
confundiram na
//classificação, por isso, utilizou-se a inclusão das classes seguintes na máscara, como 4.1
praia, duna e areal
//e 2.1 Campo alagado e área pantanosa, 2.2 formação campestre
    var mask_MapBiomas_11_12_23_33 = mask.eq(0);

    var pre_cm_mos =
prefire_CM_SelectDataSet.mosaic().updateMask(mask_MapBiomas_11_12_23_33).clip(area
_escolhida_geometry);
    var post_cm_mos =
postfire_CM_SelectDataSet.mosaic().updateMask(mask_MapBiomas_11_12_23_33).clip(are
a_escolhida_geometry);

// Chamando as bandas para aplicação no índice
/*    var BandsName = {
        B1: 'B1', B2: 'B2', B3: 'B3',
        B4: 'B4', B5: 'B5', B6: 'B6',
        B7: 'B7', B8: 'B8', B8A: 'B8A',
        B10: 'B10', B11: 'B11', B12: 'B12'
    };
*/

```

```

//----- Calcular NBR para imagens pré e pós-incêndio -----
// Aplique o NBR à plataforma específica = (NIR-SWIR2) / (NIR+SWIR2)
if (selectedDataset == 'COPERNICUS/S2' ) {
  var preNBR = pre_cm_mos.normalizedDifference(['B8', 'B12']);
  var postNBR = post_cm_mos.normalizedDifference(['B8', 'B12']);
} else {
  var preNBR = pre_cm_mos.normalizedDifference(['SR_B5', 'SR_B7']);
  var postNBR = post_cm_mos.normalizedDifference(['SR_B5', 'SR_B7']);
}

// Adicione as imagens NBR ao console à direita
print("Índice de Área Queimada previa ao incêndio:", preNBR);
print("Índice de Área Queimada posterior ao incêndio:", postNBR);

//----- Calcule a diferença entre as imagens pré e pós-incêndio -----
// O resultado é chamado de diferença NBR o dNBR
var dNBR_unscaled = preNBR.subtract(postNBR);

// Dimensione o produto para os padrões USGS (FIREMON): Sistema de monitoramento e
inventário de efeitos do fogo
var dNBR = dNBR_unscaled.multiply(1000);

// Adicione a imagem dNBR ao console direito
print("Índice diferencial de área queimada: ", dNBR);

//----- Imagem em cores reais -----

// Aplicar parâmetros de exibição específicos da plataforma para imagens coloridas
// que realçam a área queimada.
if (selectedDataset == 'COPERNICUS/S2') {
  var vis = {bands: ['B12', 'B8', 'B2'], min: 800, max: 4500, gamma: 2.0};
} else {
  var vis = {bands: ['SR_B7', 'SR_B5', 'SR_B2'], min: 8000, max: 24000, gamma: 1.5};
}

// Adicione as imagens de cores reais ao mapa.
Map.addLayer(pre_mos, vis,'Img pre-incêndio',0);
Map.addLayer(post_mos, vis,'Img pos-incêndio',1);

// Adicione as imagens de cores reais ao mapa.
Map.addLayer(pre_cm_mos, vis,'Img pre-incêndio: msk',0);
Map.addLayer(post_cm_mos, vis,'Img pos-incêndio: msk',0);

//----- Produto da área queimada - escala de cinza -----

var grey = ['white', 'black'];

// o antes e depois do NBR separadamente.
Map.addLayer(preNBR, {min: -1, max: 1, palette: grey}, 'Pre-fire NBR',0);
Map.addLayer(postNBR, {min: -1, max: 1, palette: grey}, 'Pos-tfire NBR',0);

```

```
Map.addLayer(dNBR, {min: -1000, max: 1000, palette: grey}, 'dNBR em escala de cinza',0);
```

```
//----- Produto de severidade de incêndio - Classificação -----
//(Key & Benson, 2006)
//https://www.researchgate.net/publication/241687027_Landscape_Assessment_Ground_measure_of_severity_the_Composite_Burn_Index_and_Remote_sensing_of_severity_the_Normalized_Burn_Ratio
```

```
// Defina um estilo SLD de intervalo discreto para aplicar à imagem (paleta de cores).
```

```
var sld_intervals =
  '<RasterSymbolizer>' +
  '<ColorMap type="intervals" extended="false" >' +
  '<ColorMapEntry color="#ffffff" quantity="-500" label="-500"/>' +
  '<ColorMapEntry color="#7a8737" quantity="-250" label="-250" />' +
  '<ColorMapEntry color="#acbe4d" quantity="-100" label="-100" />' +
  '<ColorMapEntry color="#0ae042" quantity="100" label="100" />' +
  '<ColorMapEntry color="#fff70b" quantity="270" label="270" />' +
  '<ColorMapEntry color="#ffaf38" quantity="440" label="440" />' +
  '<ColorMapEntry color="#ff641b" quantity="660" label="660" />' +
  '<ColorMapEntry color="#FF0000" quantity="1350" label="1350" />' +
  '</ColorMap>' +
  '</RasterSymbolizer>';
```

```
// Adicione a imagem ao mapa usando a rampa de cores como os intervalos definidos.
```

```
Map.addLayer(dNBR.sldStyle(sld_intervals), {}, 'dNBR classificado');
```

```
// Separe o resultado em 8 classes de gravidade de incêndio.
```

```
//Baseado no artigo Detecção de áreas queimadas e severidade a partir do índice espectral  $\Delta$ NBR(http://www.dsr.inpe.br/sbsr2015/files/p0104.pdf)
```

```
var thresholds = ee.Image([-500, -250, -100, 100, 270, 440, 660, 1350]);
```

```
var classified = dNBR.lt(thresholds).reduce('sum').toInt();
```

```
//=====
```

```
//                ADICIONE ESTATÍSTICAS DE ÁREA QUEIMADA
```

```
// conta o número de pixels em toda a camada.
```

```
var allpix = classified.updateMask(classified); // mascara toda a camada
var pixstats = allpix.reduceRegion({
  reducer: ee.Reducer.count(),           // conta pixels em uma única classe
  geometry: area_escolhida_geometry,
  scale: 500
});
var allpixels = ee.Number(pixstats.get('sum'));
```

```
// cria uma lista vazia para armazenar os valores na área
```

```
var arealist = [];
```

```

// cria uma função para derivar o escopo de uma classe de gravidade de incêndio
// os argumentos são o número da classe e o nome da classe
var areacount = function(cnr, name) {
  var singleMask = classified.updateMask(classified.eq(cnr)); // mascara uma única
  classe
  var stats = singleMask.reduceRegion({
    reducer: ee.Reducer.count(), // conta pixels em uma única classe
    geometry: area_escolhida_geometry,
    scale: 500
  });

  var pix = ee.Number(stats.get('sum'));
  var hect = pix.multiply(100).divide(10000); //pixel Sentinel 10m x 10m = 100m2//
  Landsat 30m x 30m = 900 m2
  var perc = pix.divide(allpixels).multiply(10000).round().divide(100); // obtém a% área
  por classe e arredonda para 2 casas decimais
  arealist.push({Class: name, Pixels: pix, Hectares: hect, Percentage: perc});
};

// classes de gravidade em ordem crescente
var names2 = ['NA', 'Alta Severidade', 'Moderada/alta Severidade',
  'Moderada/baixa Severidade', 'Baixa Severidade', 'Não queimado', 'Baixa regeneração',
  'Alta regeneração'];

// executa a função para cada classe
for (var i = 0; i < 8; i++) {
  areacount(i, names2[i]);
}

//*****
//
// ADICIONE UMA LEGENDA

// define a posição da caixa de legenda.
var legend = ui.Panel({
  style: {
    position: 'bottom-left',
    padding: '8px 15px'
  }});

// Crie um título de legenda.
var legendTitle = ui.Label({
  value: 'Classes de dNBR',
  style: {fontWeight: 'bold',
    fontSize: '18px',
    margin: '0 0 4px 0',
    padding: '0'
  }});
// Adicione o título à caixa.
legend.add(legendTitle);

```

```

// Crie e estilize 1 linha da legenda.
    var makeRow = function(color, name) {

// Crie o rótulo que é realmente a caixa de cores.
    var colorBox = ui.Label({
        style: {
            backgroundColor: '#' + color,
// Use (preenchimento) para preencher e fornecer a altura e largura da caixa.
            padding: '8px',
            margin: '0 0 4px 0'
        });

// Crie uma etiqueta preenchida com o texto descritivo.
    var description = ui.Label({
        value: name,
        style: {margin: '0 0 4px 6px'}
    });

// retorna o painel
    return ui.Panel({
        widgets: [colorBox, description],
        layout: ui.Panel.Layout.Flow('horizontal')
    });

// Paleta de cores
    var palette =['7a8737', 'acbe4d', '0ae042', 'fff70b', 'ffaf38', 'ff641b', 'FF0000', 'ffffff'];

// Nome da legenda
    var names = ['Alta regeneração','Baixa regeneração', 'Não queimado', 'Baixa
Severidade','Moderada-Baixa Severidade', 'Moderada-Alta Severidade', 'Alta Severidade',
'NA'];

// Adicionar cor e nomes
    for (var i = 0; i < 8; i++) {
        legend.add(makeRow(palette[i], names[i]));
    }
// Adicione a legenda ao mapa (a legenda também pode ser impressa no console)
    Map.add(legend);
    legend_obj.add(legend);

    print('Área queimada por classe de Severidade', arealist, '--> Agora Clique na lista de
objetos para ver as classes individuais');

    dNBR_obj.add(dNBR);

//=====
//          SELECIONANDO SOMENTE DE BAIXA GRAVIDADE A ALTA PARA
EXPORTAR O SHP

```

//Sadeck (2020)

//Adaptado:

<https://www.youtube.com/watch?v=gA3TGd5PaEs&list=PLNFvG6bTA4NReWtgC93Mh9T w1RNG4EBMP&index=40&t=557s>

```

var boxcar = ee.Kernel.square({
  radius: 5, units: 'pixels', normalize: true
});

//var dNBR_ = dNBR_obj.get(0);

var smooth = dNBR.convolve(boxcar);

var Fire = smooth.gt(100).and(smooth.lt(1350));

var vec = Fire.eq(1);
var masked = vec.updateMask(vec);
var fc = masked.reduceToVectors({
  scale: 30,
  bestEffort: true,
  maxPixels:40e13,
  geometry: area_escolhida_geometry,
  // tileScale: 10,
});

fc_obj.add(fc);

Map.addLayer(Fire.clip(area_escolhida_geometry), {}, 'Threshold',0)
Map.addLayer(smooth.clip(area_escolhida_geometry), {}, 'smooth',0)
Map.addLayer(fc, {color: 'red'}, 'Mapeamento',0)
}

function Export_dNBR_ToDrive(){
//=====
//          PREPARE O ARQUIVO PARA EXPORTAR
var area_escolhida = area_escolhido_obj.get(0);
var area_escolhida_geometry = area_escolhida.geometry();

var dNBR_ = dNBR_obj.get(0);

var id = dNBR_.id().getInfo();

Export.image.toDrive(
  {image: dNBR_, scale: 30, description: 'dNBR',folder: 'Mapeamento de queimadas',
  fileNamePrefix: 'dNBR',
  region: area_escolhida_geometry, maxPixels: 1e10});

var btn_exp_dnbr_ = btn_exp_dnbr_obj.get(0);
btn_exp_dnbr_.setDisabled(true);

```

```

}

function Export_SHP_ToDrive(){
//=====
//          EXPORTANDO O ARQUIVO VETORIAL
var fc_ = fc_obj.get(0);
Export.table.toDrive({
  collection:fc_,
  description: 'Areas_mapeadas',
  folder: 'Mapeamento de queimadas',
  fileFormat: 'KML',
  // maxPixels: 1000000000.0
})

var btn_exp_shp_ = btn_exp_shp_obj.get(0);
btn_exp_shp_.setDisabled(true);

}

//=====
// Applies scaling factors USGS Landsat 8 Level 2, Collection 2, Tier 1
function applyScaleFactors(image) {
  var opticalBands = image.select('SR_B.').multiply(0.0000275).add(-0.2);
  var thermalBands = image.select('ST_B.*').multiply(0.00341802).add(149.0);
  return image.addBands(opticalBands, null, true)
    .addBands(thermalBands, null, true);
}
function maskL8sr(image) {
// Os bits 3 e 5 são nuvens e sombras de nuvens, respectivamente.
var cloudShadowBitMask = 1 << 3;
var cloudsBitMask = 1 << 5;
var snowBitMask = 1 << 4;
// Obtenha a banda de controle de qualidade de pixels
var qa = image.select('QA_PIXEL');
// Todos os sinalizadores devem ser definidos como zero, indicando condições claras ou
// Livre de nuvem
// Plataforma L8
var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0)
  .and(qa.bitwiseAnd(cloudsBitMask).eq(0))
  .and(qa.bitwiseAnd(snowBitMask).eq(0));
// Retorna a imagem mascarada e dimensionada para a refletância TOA, sem as bandas de
QA.
return image.updateMask(mask)
  .select("SR_B.")
  .copyProperties(image, ["system:time_start"]);
}

function maskS2sr(image) {

```



```

// Obtenha a banda de QA do controle de qualidade de pixel.
var qa = image.select('QA60');
// Os bits 10 e 11 são nuvens e nuvens cirros, respectivamente.
var cloudBitMask = ee.Number(2).pow(10).int();
var cirrusBitMask = ee.Number(2).pow(11).int();
// Todos os sinalizadores devem ser definidos como zero, indicando condições limpas ou
// livre de nuvem.
var mask = qa.bitwiseAnd(cloudBitMask).eq(0)
    .and(qa.bitwiseAnd(cirrusBitMask).eq(0));
// Retorna a imagem mascarada e dimensionada para a refletância TOA, sem as bandas de
// QA.
return image.updateMask(mask)
    .copyProperties(image, ["system:time_start"]);
}

```

```

panel.add(DataSelectorLabel);
panel.add(datasetSelector);
panel.add(labelInteresse);
panel.add(citiesDD);
panel.add(labelInteresseZoom);
panel.add(cityDD);
panel.add(Zoom);
panel.add(labelPeriodo);

```

```

ui.root.insert(1,panel);

```

```

//Em parceria com o Centro de Rede Internacional de Informações de Ciências da Terra
// (CIESIN) da Universidade
// de Columbia, o Facebook usou técnicas de visão computacional de última geração para
// identificar edifícios
//a partir de serviços de mapeamento acessíveis ao público para criar os conjuntos de dados
// populacionais
//mais precisos do mundo (https://samapriya.github.io/awesome-gee-community-datasets/projects/hrsl/)
var image=HRSL.median();
var rgbVis = {palette: palettes.colorbrewer.Red5[9]};
Map.addLayer(image.clip(cities),rgbVis,'HRSL',0);

```

## APÊNDICE B – QML STYLES KEY&BENSON (2006)

```

<!DOCTYPE qgis PUBLIC 'http://mrcc.com/qgis.dtd' 'SYSTEM'>
<qgis minScale="1e+08" styleCategories="AllStyleCategories" maxScale="0"
version="3.16.7-Hannover" hasScaleBasedVisibilityFlag="0">
  <flags>
    <Identifiable>1</Identifiable>
    <Removable>1</Removable>
    <Searchable>1</Searchable>
  </flags>
  <temporal mode="0" enabled="0" fetchMode="0">
    <fixedRange>
      <start></start>
      <end></end>
    </fixedRange>
  </temporal>
  <customproperties>
    <property key="WMSBackgroundLayer" value="false"/>
    <property key="WMSPublishDataSourceUrl" value="false"/>
    <property key="embeddedWidgets/count" value="0"/>
    <property key="identify/format" value="Value"/>
  </customproperties>
  <pipe>
    <provider>
      <resampling zoomedOutResamplingMethod="nearestNeighbour" enabled="false"
maxOversampling="2" zoomedInResamplingMethod="nearestNeighbour"/>
    </provider>
    <rasterrenderer alphaBand="-1" nodataColor="" type="singlebandpseudocolor"
opacity="1" classificationMax="1350" band="1" classificationMin="-250">
      <rasterTransparency/>
      <minMaxOrigin>
        <limits>None</limits>
        <extent>WholeRaster</extent>
        <statAccuracy>Estimated</statAccuracy>
        <cumulativeCutLower>0.02</cumulativeCutLower>
        <cumulativeCutUpper>0.98</cumulativeCutUpper>
        <stdDevFactor>2</stdDevFactor>
      </minMaxOrigin>
      <rastershader>
        <colorrampshader labelPrecision="0" colorRampType="INTERPOLATED"
minimumValue="-250" maximumValue="1350" clip="0" classificationMode="2">
          <colorramp name="[source]" type="gradient">
            <prop v="215,25,28,255" k="color1"/>
            <prop v="43,131,186,255" k="color2"/>
            <prop v="0" k="discrete"/>
            <prop v="gradient" k="rampType"/>
            <prop v="0.25;253,174,97,255:0.5;255,255,191,255:0.75;171,221,164,255"
k="stops"/>
          </colorramp>
          <item label="Alta regeneração" color="#7a8737" alpha="255" value="-250"/>

```

```
<item label="Baixa regeneração" color="#acbe4d" alpha="255" value="-100"/>
<item label="Não queimado" color="#0ae042" alpha="255" value="100"/>
<item label="Baixa severidade" color="#fff70b" alpha="255" value="270"/>
<item label="Moderada-baixa severidade" color="#ffaf38" alpha="255" value="440"/>
<item label="Moderada-alta severidade" color="#ff641b" alpha="255" value="660"/>
<item label="Alta severidade" color="#ff0000" alpha="255" value="1350"/>
</colorrampshader>
</rastershader>
</rasterrenderer>
<brightnesscontrast brightness="0" gamma="1" contrast="0"/>
<huesaturation grayscaleMode="0" colorizeStrength="100" saturation="0"
colorizeBlue="128" colorizeRed="255" colorizeGreen="128" colorizeOn="0"/>
<rasterresampler maxOversampling="2"/>
<resamplingStage>resamplingFilter</resamplingStage>
</pipe>
<blendMode>0</blendMode>
</qgis>
```

## APÊNDICE C – QML STYLES KEY&amp;BENSON (2006) PALETIZADO

```

<!DOCTYPE qgis PUBLIC 'http://mrcc.com/qgis.dtd' 'SYSTEM'>
<qgis version="3.16.7-Hannover" hasScaleBasedVisibilityFlag="0"
styleCategories="AllStyleCategories" minScale="1e+08" maxScale="0">
  <flags>
    <Identifiable>1</Identifiable>
    <Removable>1</Removable>
    <Searchable>1</Searchable>
  </flags>
  <temporal mode="0" fetchMode="0" enabled="0">
    <fixedRange>
      <start></start>
      <end></end>
    </fixedRange>
  </temporal>
  <customproperties>
    <property value="false" key="WMSBackgroundLayer"/>
    <property value="false" key="WMSPublishDataSourceUrl"/>
    <property value="0" key="embeddedWidgets/count"/>
    <property value="Value" key="identify/format"/>
  </customproperties>
  <pipe>
    <provider>
      <resampling zoomedInResamplingMethod="nearestNeighbour" maxOversampling="2"
enabled="false" zoomedOutResamplingMethod="nearestNeighbour"/>
    </provider>
    <rasterrenderer alphaBand="-1" opacity="1" band="1" classificationMax="7"
classificationMin="1" type="singlebandpseudocolor" nodataColor="">
      <rasterTransparency/>
      <minMaxOrigin>
        <limits>MinMax</limits>
        <extent>WholeRaster</extent>
        <statAccuracy>Estimated</statAccuracy>
        <cumulativeCutLower>0.02</cumulativeCutLower>
        <cumulativeCutUpper>0.98</cumulativeCutUpper>
        <stdDevFactor>2</stdDevFactor>
      </minMaxOrigin>
      <rastershader>
        <colorrampshader maximumValue="7" minimumValue="1" clip="0" labelPrecision="0"
colorRampType="INTERPOLATED" classificationMode="2">
          <colorramp name="[source]" type="gradient">
            <prop k="color1" v="43,131,186,255"/>
            <prop k="color2" v="215,25,28,255"/>
            <prop k="discrete" v="0"/>
            <prop k="rampType" v="gradient"/>
            <prop k="stops"
v="0.25;171,221,164,255:0.5;255,255,191,255:0.75;253,174,97,255"/>
          </colorramp>
          <item value="1" alpha="255" label="1" color="#7a8737"/>

```

```
<item value="2" alpha="255" label="2" color="#acbe4d"/>
<item value="3" alpha="255" label="3" color="#0ae042"/>
<item value="4" alpha="255" label="4" color="#fff70b"/>
<item value="5" alpha="255" label="5" color="#ffaf38"/>
<item value="6" alpha="255" label="6" color="#ff641b"/>
<item value="7" alpha="255" label="7" color="#e50000"/>
</colorrampshader>
</rastershader>
</rasterrender>
<brightnesscontrast brightness="0" gamma="1" contrast="0"/>
<huesaturation colorizeRed="255" saturation="0" colorizeGreen="128"
colorizeBlue="128" colorizeOn="0" colorizeStrength="100" grayscaleMode="0"/>
<rasterresampler maxOversampling="2"/>
<resamplingStage>resamplingFilter</resamplingStage>
</pipe>
<blendMode>0</blendMode>
</qgis>
```