OmaR - Desafio A3Data

https://github.com/SlackPen/DesafioA3Data.git (https://github.com/SlackPen/DesafioA3Data.git)

- OCORRÊNCIA.csv - Informações sobre as ocorrências.
- OCORRÊNCIA_TIPO.csv - Informações sobre o tipo de ocorrência.
- AERONAVE.csv - Informações sobre as aeronaves envolvidas nas ocorrências.
- FATOR_CONTRIBUINTE.csv - Informações sobre os fatores contribuinte das ocorrências que tiveram investigações finalizadas.
- RECOMENDAÇÃO.csv - Informações sobre as recomendações de segurança geradas nas ocorrências.

In [106]:
```
import os, sys, glob
import warnings
warnings.filterwarnings('ignore')
```

In [103]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as srn
```

In [3]:
```
os.getcwd()
```

Out[3]:
```
'E:\\OMAR\\Desafio\\A3DATA'
```

In [4]:
```
arqs = { n:nome for n,nome in enumerate(glob.glob('*.csv'))}
arqs
```

Out[4]:
```
{0: 'aeronave.csv',
 1: 'fator_contribuinte.csv',
 2: 'ocorrencia.csv',
 3: 'ocorrencia_tipo.csv',
 4: 'recomendacao.csv'}
```

In [5]:
```
dfAeronave = pd.read_csv(arqs[0], sep=';')
dfFator    = pd.read_csv(arqs[1], sep=';')
dfOcorr    = pd.read_csv(arqs[2], sep=';')
dfTipoOcor = pd.read_csv(arqs[3], sep=';')
dfRecomend = pd.read_csv(arqs[4], sep=';')
```

A idéia foi criar um Modelo Entidade Relacionamento para facilitar a visualização quanto a cada tipo de dado e os relacionamentos entre os mesmos

Usei mysql pelo recurso de banco de dados relacional e pela ferramenta de modelagem visual embutida no kit de ferramentas open source do mesmo.

In [40]:
```
import mysql.connector

cnx = mysql.connector.connect(user='root', password='omar',
                              host='127.0.0.1',
                              database='myomardb')
```

Abaixo foi gerado um script para cada arquivo e o mesmo processado no mysql

In [25]:
```
sSql = pd.io.sql.get_schema(dfRecomend, 'tbRecomendacoes')
print(sSql.replace('"', ''))
```

```
CREATE TABLE tbRecomendacoes (
codigo_ocorrencia4 INTEGER,
  recomendacao_numero TEXT,
  recomendacao_dia_assinatura TEXT,
  recomendacao_dia_encaminhamento TEXT,
  recomendacao_dia_feedback TEXT,
  recomendacao_conteudo TEXT,
  recomendacao_status TEXT,
  recomendacao_destinatario_sigla TEXT,
  recomendacao_destinatario TEXT
)
```

Abaixo comando para criar os relacionamentos nas tabelas conforme observei a disposição dos dados nos arquivos

In [ ]:
```
alter table tbtipoocorrencias  add foreign key fkTipoOcorrencia
(codigo_ocorrencia1) references tbocorrencias(codigo_ocorrencia1)
alter table tbaeronave         add foreign key fkAeronaveOcorrencia
(codigo_ocorrencia2) references tbocorrencias(codigo_ocorrencia2)
alter table tbfator            add foreign key fkFatorOcorrencia
(codigo_ocorrencia3) references tbocorrencias(codigo_ocorrencia3)
alter table tbrecomendacoes    add foreign key fkRecomendacoesOcorre
ncias (codigo_ocorrencia4) references tbocorrencias(codigo_ocorrenci
a4)
```

Feita a conexão efetua-se a exportação para o mysql

In [60]:
```
# import the module
from sqlalchemy import create_engine

# create sqlalchemy engine
engine = create_engine("mysql+pymysql://{user}:{pw}@localhost/{db}"
                       .format(user="root",
                               pw="omar",
                               db="myomardb"))
```

In [62]:
```
dfOcorr.to_sql('tbocorrencias', con = engine, if_exists = 'append',
chunksize = 1000, index=False)
dfAeronave.to_sql('tbaeronave', con = engine, if_exists = 'append',
chunksize = 1000, index=False)
dfFator.to_sql('tbfator', con = engine, if_exists = 'append', chunks
ize = 1000, index=False)
dfTipoOcor.to_sql('tbtipoocorrencias', con = engine, if_exists = 'ap
pend', chunksize = 1000, index=False)
dfRecomend.to_sql('tbrecomendacoes', con = engine, if_exists = 'appe
nd', chunksize = 1000, index=False)
```

Modelo Entidade Relacionamento

In [119]:
```
from IPython.display import Image
Image(filename='mer.png')
```

Out[119]:

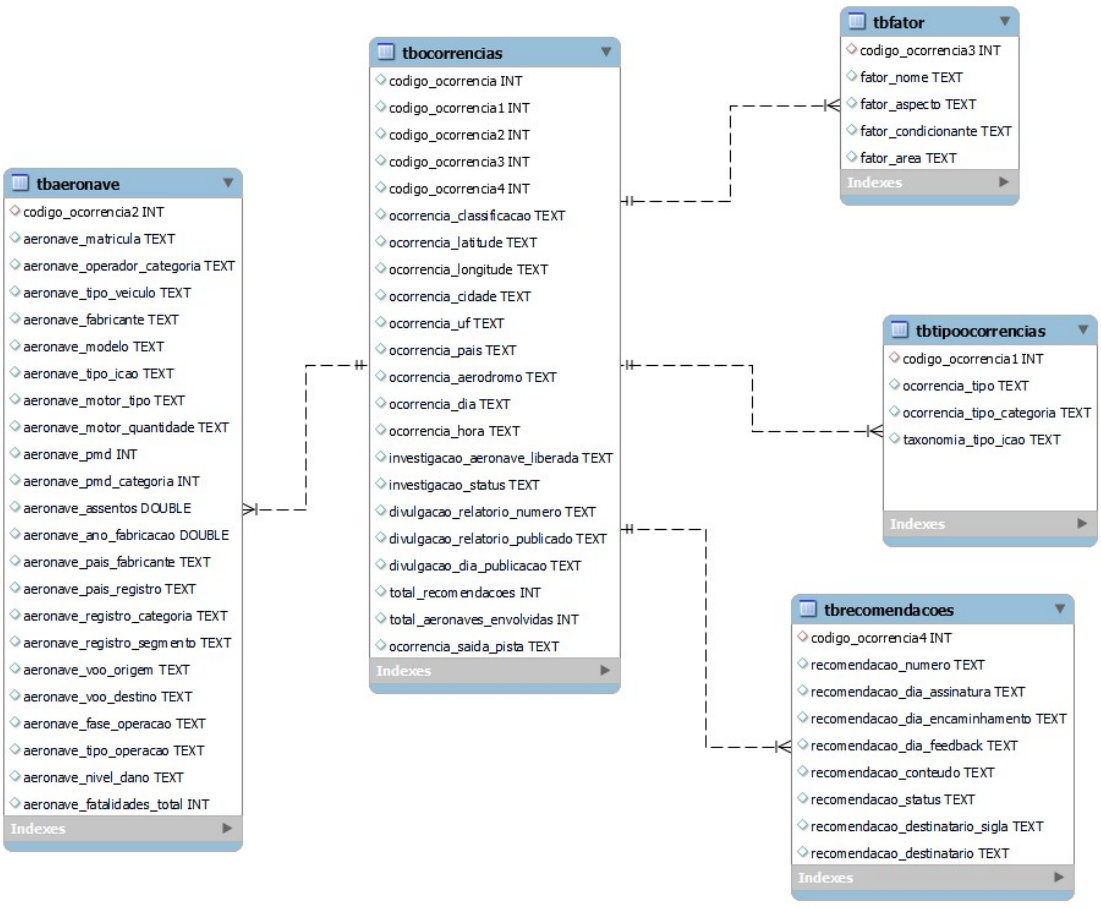


Tabela Ocorrências - Visão Geral

In [180]:
```
dfOcorr.shape
```

Out[180]: (5167, 22)

In [118]:
```
dfOcorr.head(5)
```

Out[118]:

	codigo_ocorrencia	codigo_ocorrencia1	codigo_ocorrencia2	codigo_ocorrencia3	codigo_
0	52242	52242	52242	52242	
1	45331	45331	45331	45331	
2	45333	45333	45333	45333	
3	45401	45401	45401	45401	
4	45407	45407	45407	45407	

5 rows × 22 columns

Relatório das ocorrências que foram ou não divulgados

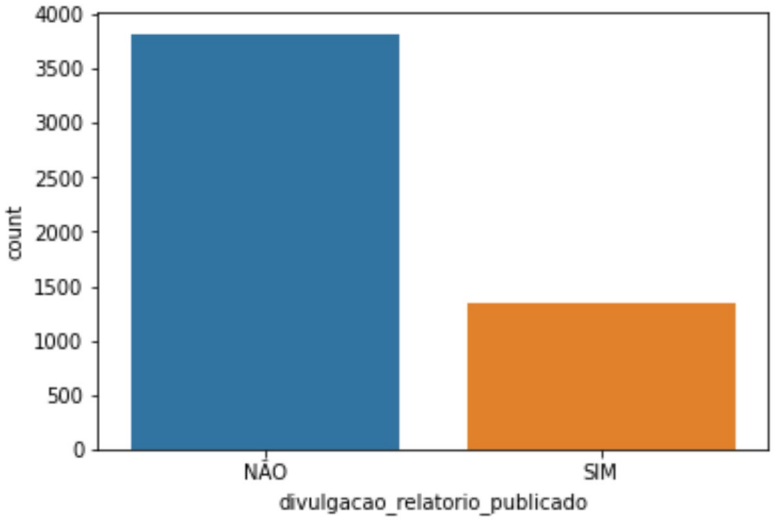
In [294]:
```
dfOcorr.groupby(['divulgacao_relatorio_publicado']) \
.agg(Percentual=('divulgacao_relatorio_publicado', 'count')) \
.groupby(level=0).apply(lambda x:  round(x / x.count() / dfOcorr.sha
pe[0] * 100, 2)  )
```

Out[294]:

	Percentual
divulgacao_relatorio_publicado	
NÃO	73.89
SIM	26.11

In [117]:
```
srn.countplot(dfOcorr['divulgacao_relatorio_publicado'])
```

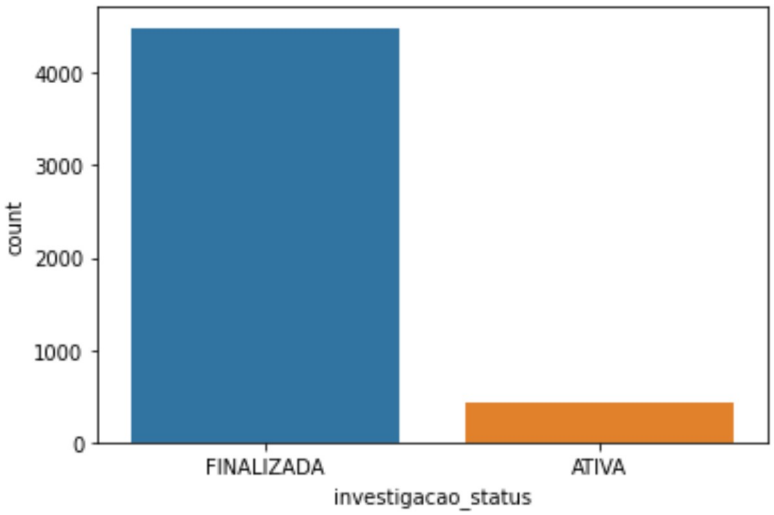
Out[117]:
```
<AxesSubplot:xlabel='divulgacao_relatorio_publicado', ylabel='coun
t'>
```



Status das Investigações sobre as ocorrências

In [116]:
```
srn.countplot(dfOcorr['investigacao_status'])
```

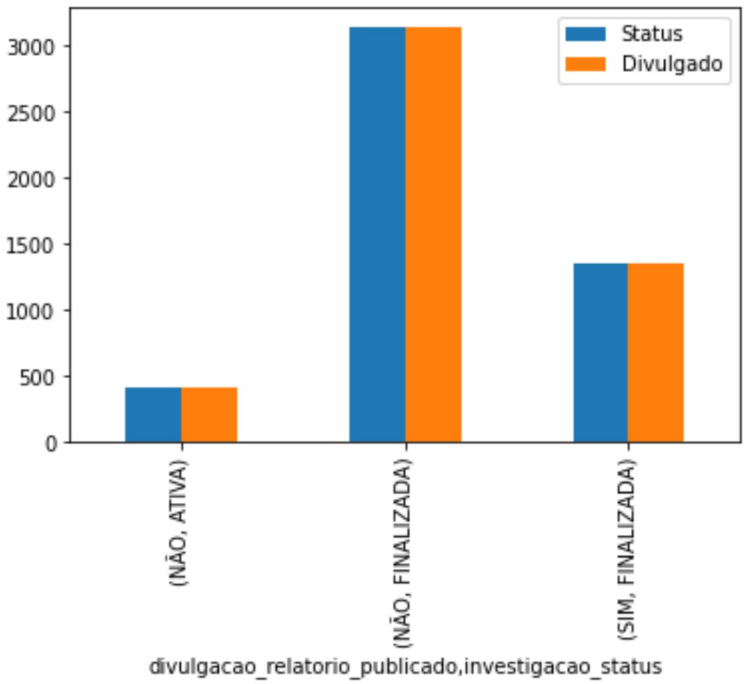
Out[116]:
```
<AxesSubplot:xlabel='investigacao_status', ylabel='count'>
```



Status das investigações em ocorrências com relatórios divulgados ou não

In [115]:
```
dfOcorr.groupby(['divulgacao_relatorio_publicado', 'investigacao_status']).agg(Status=('investigacao_status', 'count'), Divulgado=('divulgacao_relatorio_publicado', 'count')).plot.bar()
```

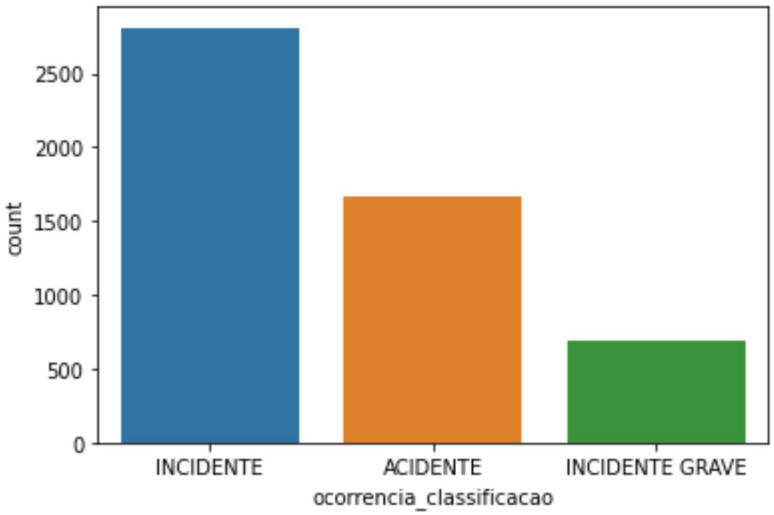
Out[115]:
```
<AxesSubplot:xlabel='divulgacao_relatorio_publicado,investigacao_status'>
```



Quantificação dos tipos de ocorrências

In [254]:
```
srn.countplot(dfOcorr['ocorrencia_classificacao'])
```

Out[254]:
```
<AxesSubplot:xlabel='ocorrencia_classificacao', ylabel='count'>
```



Ocorrências por Estado

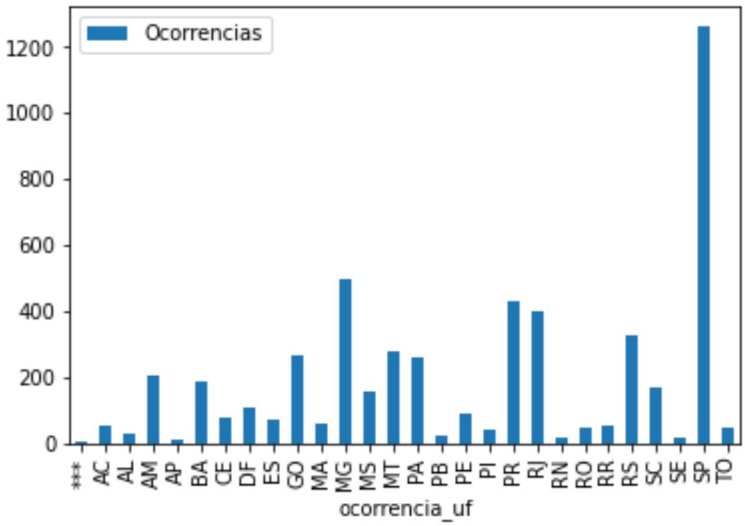
In [293]:
```
dfOcorr.groupby(['ocorrencia_uf']) \
.agg(Percentual=('ocorrencia_uf', 'count')) \
.groupby(level=0).apply(lambda x:  round(x / x.count() / dfOcorr.sha
pe[0] * 100, 2)  )
```

Out[293]:

ocorrencia_uf	Percentual
***	0.04
AC	0.95
AL	0.54
AM	4.01
AP	0.23
BA	3.54
CE	1.51
DF	2.05
ES	1.41
GO	5.19
MA	1.14
MG	9.66
MS	3.06
MT	5.34
PA	5.05
PB	0.43
PE	1.66
PI	0.74
PR	8.34
RJ	7.78
RN	0.33
RO	0.83
RR	1.05
RS	6.33
SC	3.23
SE	0.27
SP	24.40
TO	0.89

In [101]:
```
dfOcorr.groupby(['ocorrencia_uf']).agg(Ocorrencias=('ocorrencia_uf',
'count')).plot.bar()
```

Out[101]: <AxesSubplot:xlabel='ocorrencia_uf'>



In [ ]:

In [ ]:

Tabela Tipos de ocorrências - Visão Geral

In [181]:
```
dfTipoOcor.shape
```

Out[181]: (5347, 4)

In [123]:
```
dfTipoOcor.head(5)
```

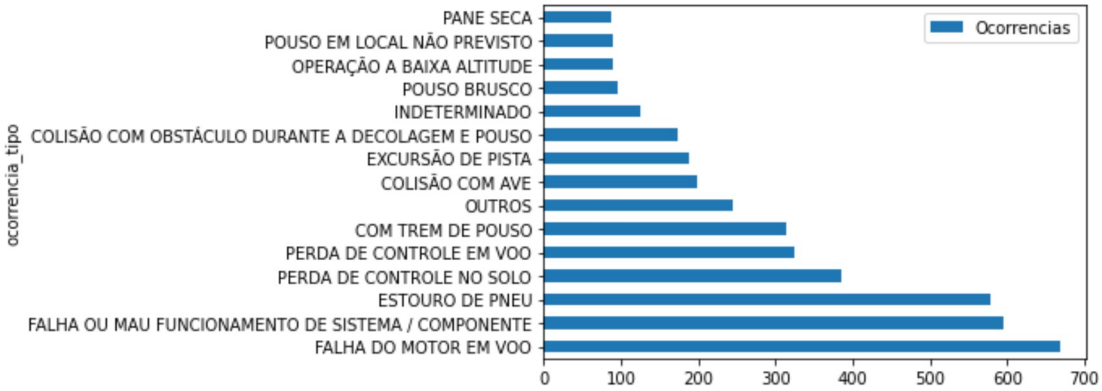
Out[123]:

	codigo_ocorrencia1	ocorrencia_tipo	ocorrencia_tipo_categoria	taxonomia_tipo_icao
0	45331	COM PESSOAL EM VOO	OUTROS | COM PESSOAL EM VOO	OTHR
1	45332	PERDA DE CONTROLE NO SOLO	PERDA DE CONTROLE NO SOLO	LOC-G
2	45333	FALHA DO MOTOR EM VOO	FALHA OU MAU FUNCIONAMENTO DO MOTOR | FALHA DO...	SCF-PP
3	45334	ESTOURO DE PNEU	FALHA OU MAU FUNCIONAMENTO DE SISTEMA / COMPON...	SCF-NP
4	45390	OPERAÇÃO A BAIXA ALTITUDE	OPERAÇÃO A BAIXA ALTITUDE	LALT

Tipos de problemas ocorridos (15+)

In [158]:
```
dfTipoOcor.groupby(['ocorrencia_tipo']).agg(Ocorrencias=('ocorrencia_tipo', 'count')) \
    .sort_values(by=['Ocorrencias'], axis=0, ascending=False)[0:15].plot.barh()
```

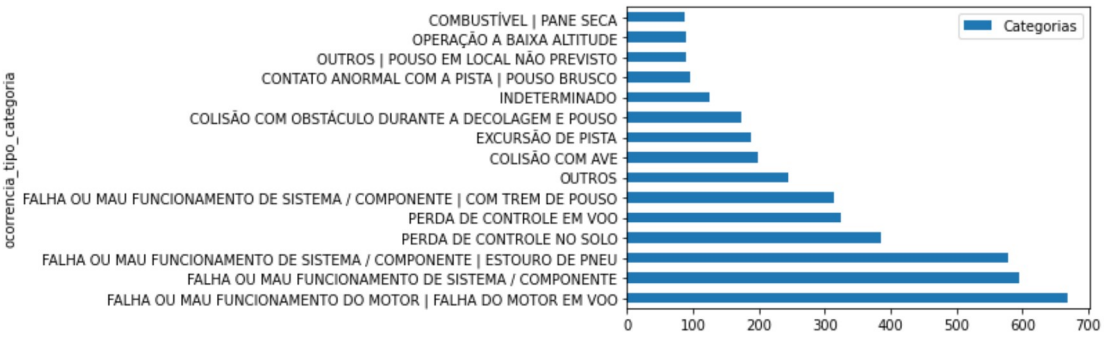
Out[158]: <AxesSubplot:ylabel='ocorrencia_tipo'>



Categorias (15+)

In [162]:
```
dfTipoOcor.groupby(['ocorrencia_tipo_categoria']).agg(Categorias=('ocorrencia_tipo_categoria', 'count')) \
    .sort_values(by=['Categorias'], axis=0, ascending=False)[0:15].plot.barh()
```

Out[162]: <AxesSubplot:ylabel='ocorrencia_tipo_categoria'>



In [ ]:

In [ ]:

Tabela Aeronaves - Visão Geral

In [184]:
```
dfAeronave.shape
```

Out[184]: (5235, 23)

In [120]:
```
dfAeronave.head(5)
```

Out[120]:

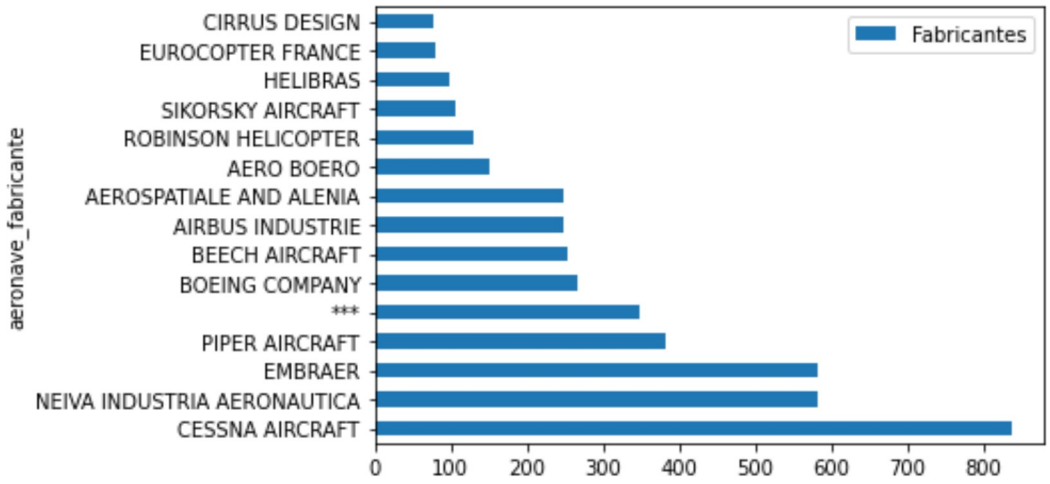
	codigo_ocorrencia2	aeronave_matricula	aeronave_operador_categoria	aeronave_tipo_vei
0	45331	PRTKB	***	AV
1	45332	PTUDD	***	AV
2	45333	PTGOO	***	AV
3	45334	PRMHX	REGULAR	AV
4	45390	PTUEW	***	AV

5 rows × 23 columns

Fabricante Aeronave (15+)

In [187]:
```
dfAeronave.groupby(['aeronave_fabricante']).agg(Fabricantes=('aeronave_fabricante', 'count')) \
    .sort_values(by=['Fabricantes'], axis=0, ascending=False)[0:15].plot.barh()
```

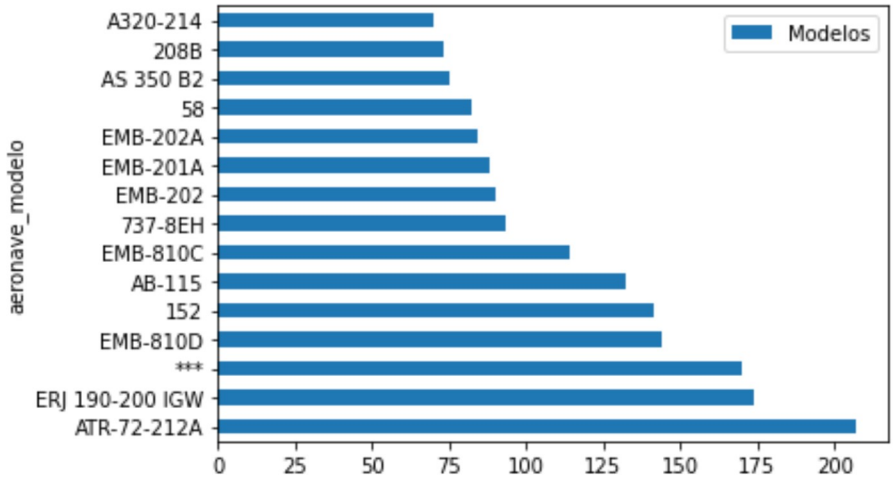
Out[187]: <AxesSubplot:ylabel='aeronave_fabricante'>



Modelos de Aeronaves com maior ocorrência de problemas (15+)

In [192]:
```
dfAeronave.groupby(['aeronave_modelo']).agg(Modelos=('aeronave_modelo', 'count')) \
    .sort_values(by=['Modelos'], axis=0, ascending=False)[0:15].plot.barh()
```

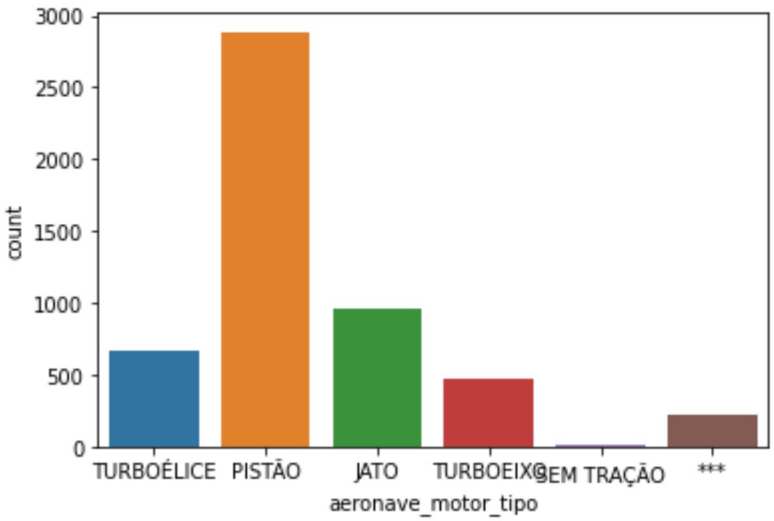
Out[192]: <AxesSubplot:ylabel='aeronave_modelo'>



Tipo de motor das aeronaves

In [190]:
```
srn.countplot(dfAeronave['aeronave_motor_tipo'])
```

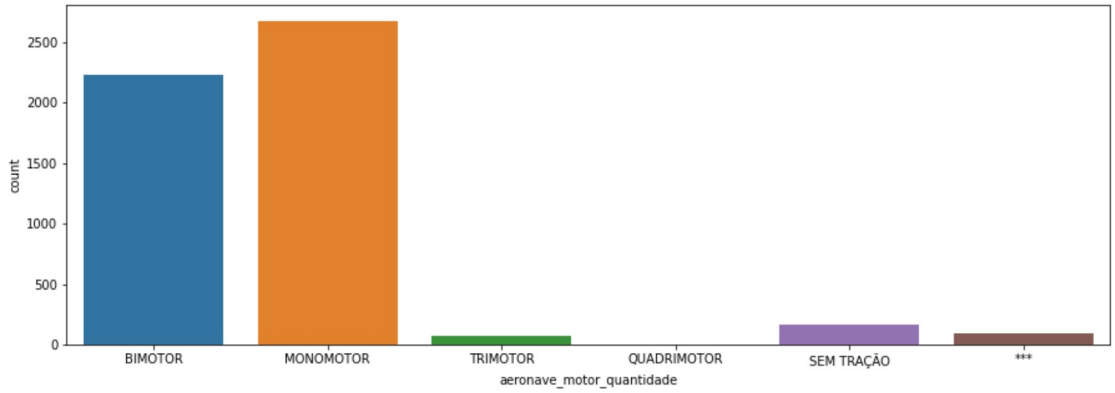
Out[190]: <AxesSubplot:xlabel='aeronave_motor_tipo', ylabel='count'>



Qtde. de Motor me aeronaves das ocorrências

In [198]:
```
plt.figure(figsize=(15,5))
srn.countplot(dfAeronave['aeronave_motor_quantidade'])
```

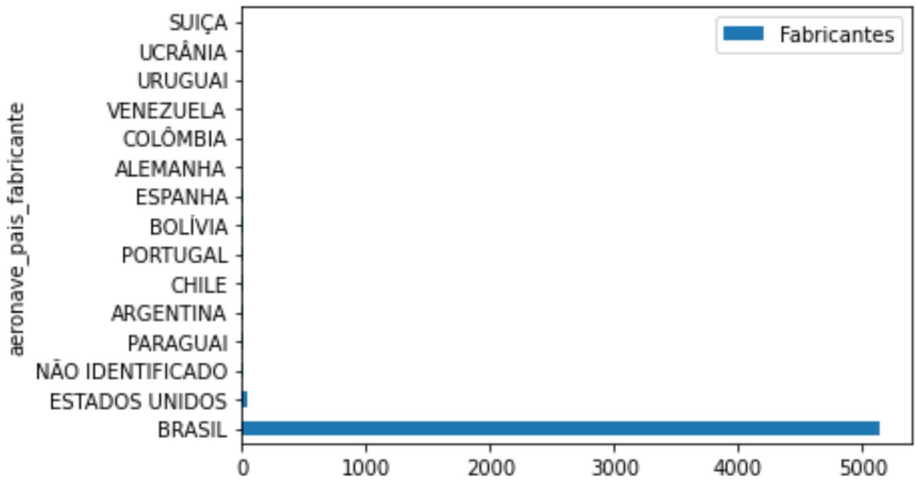
Out[198]: <AxesSubplot:xlabel='aeronave_motor_quantidade', ylabel='count'>



Fabricante da aeronave com problema (15+)

In [209]:
```
dfAeronave.groupby(['aeronave_pais_fabricante']).agg(Fabricantes=('aeronave_pais_fabricante', 'count')) \
    .sort_values(by=['Fabricantes'], axis=0, ascending=False)[0:15].plot.barh()
```

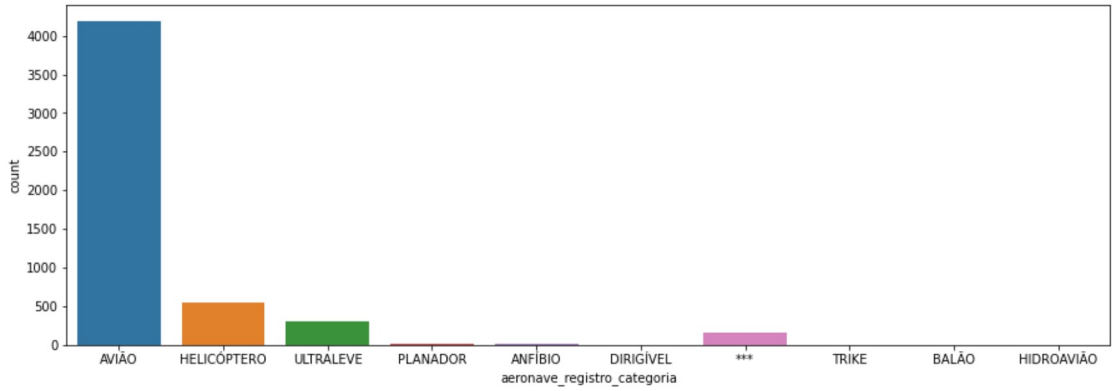
Out[209]: <AxesSubplot:ylabel='aeronave_pais_fabricante'>



Categorias de registro das aeronaves

In [210]:
```
plt.figure(figsize=(15,5))
srn.countplot(dfAeronave['aeronave_registro_categoria'])
```

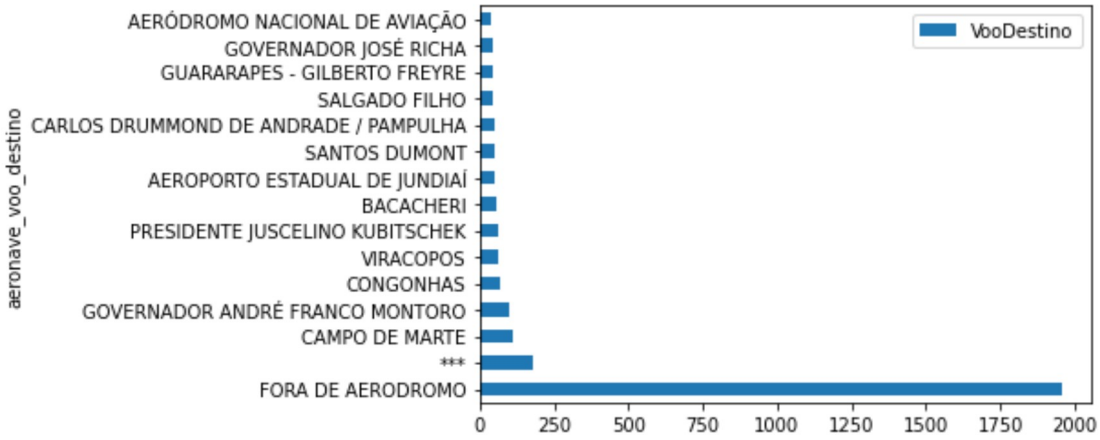
Out[210]: <AxesSubplot:xlabel='aeronave_registro_categoria', ylabel='count'>



Destino de voo

In [211]:
```
dfAeronave.groupby(['aeronave_voo_destino']).agg(VooDestino=('aeronave_voo_destino', 'count')) \
    .sort_values(by=['VooDestino'], axis=0, ascending=False)[0:15].plot.barh()
```

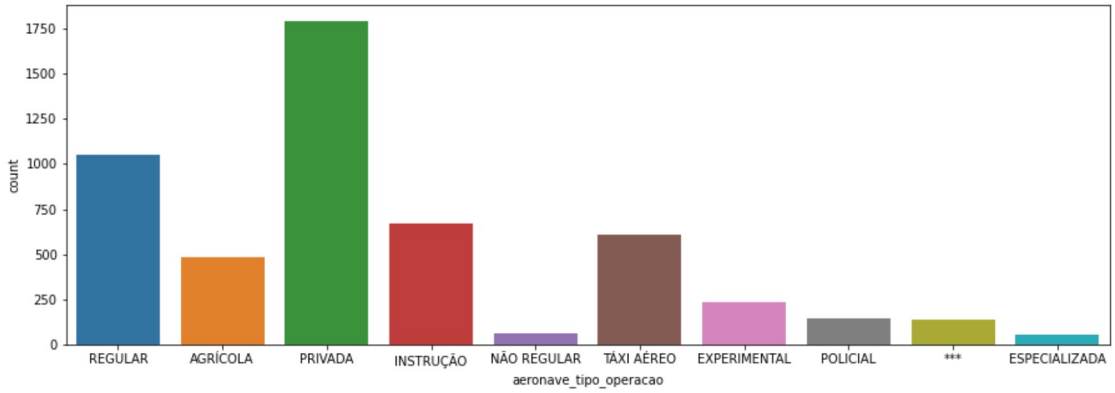
Out[211]: <AxesSubplot:ylabel='aeronave_voo_destino'>



Tipos de operações em que se encontrava a aeronave quando se deu a ocorrência

In [206]:
```
plt.figure(figsize=(15,5))
srn.countplot(dfAeronave['aeronave_tipo_operacao'])
```

Out[206]: <AxesSubplot:xlabel='aeronave_tipo_operacao', ylabel='count'>



Nível de dano ocorrido nas aeronaves

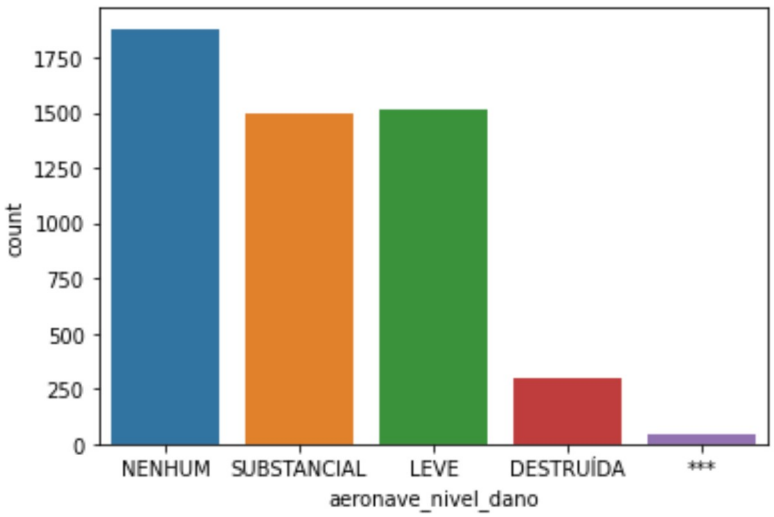
In [302]:
```
dfAeronave.groupby(['aeronave_nivel_dano']) \
.agg(Percentual=('aeronave_nivel_dano', 'count')) \
.groupby(level=0).apply(lambda x:  round(x / x.count() / dfAeronave.
shape[0] * 100, 2)  )
```

Out[302]:

aeronave_nivel_dano	Percentual
***	0.90
DESTRUÍDA	5.73
LEVE	28.84
NENHUM	35.89
SUBSTANCIAL	28.63

In [205]:
```
srn.countplot(dfAeronave['aeronave_nivel_dano'])
```

Out[205]: <AxesSubplot:xlabel='aeronave_nivel_dano', ylabel='count'>



Fatalidades total

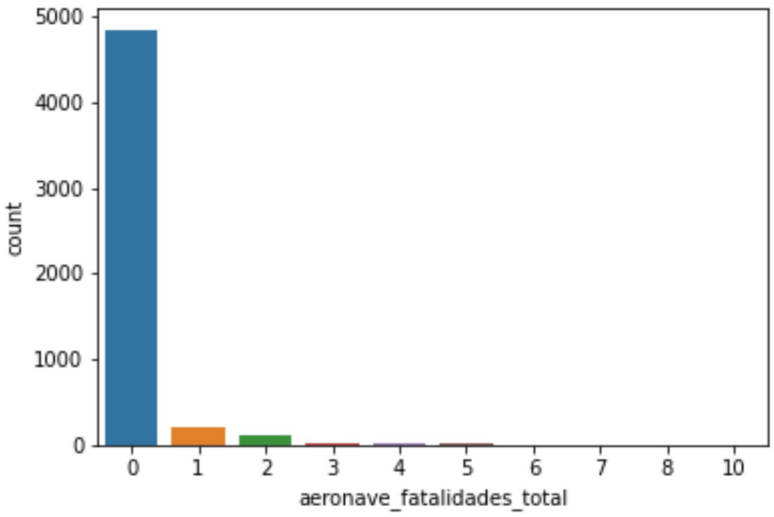
In [301]:
```
dfAeronave.groupby(['aeronave_fatalidades_total']) \
.agg(Percentual=('aeronave_fatalidades_total', 'count')) \
.groupby(level=0).apply(lambda x:  round(x / x.count() / dfAeronave.
shape[0] * 100, 2)  )
```

Out[301]:

aeronave_fatalidades_total	Percentual
0	92.55
1	3.88
2	2.04
3	0.48
4	0.46
5	0.36
6	0.10
7	0.06
8	0.06
10	0.02

In [212]:
```
srn.countplot(dfAeronave['aeronave_fatalidades_total'])
```

Out[212]: <AxesSubplot:xlabel='aeronave_fatalidades_total', ylabel='count'>



In [ ]:

In [ ]:

Tabela Fatores - Visão Geral

In [182]:
```
dfFator.shape
```

Out[182]:
```
(3464, 5)
```

In [70]:
```
dfFator.head(5)
```

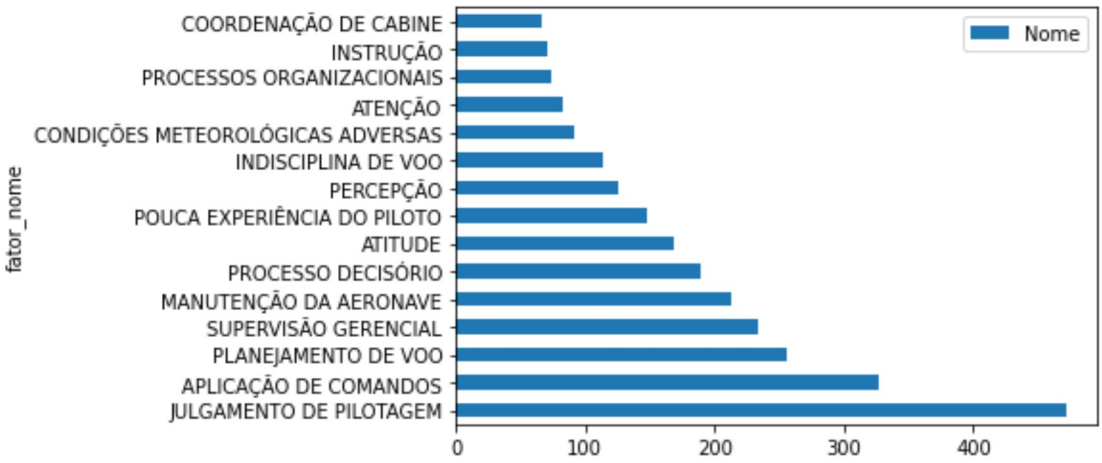
Out[70]:

	codigo_ocorrencia3	fator_nome	fator_aspecto	fator_condicionante	fator_a
0	45331	APLICAÇÃO DE COMANDOS	DESEMPENHO DO SER HUMANO	OPERAÇÃO DA AERONAVE	FAT OPERACION
1	45331	ATENÇÃO	ASPECTO PSICOLÓGICO	INDIVIDUAL	FAT HUMA
2	45331	CAPACITAÇÃO E TREINAMENTO	ASPECTO PSICOLÓGICO	ORGANIZACIONAL	FAT HUMA
3	45331	CLIMA ORGANIZACIONAL	ASPECTO PSICOLÓGICO	ORGANIZACIONAL	FAT HUMA
4	45331	COMUNICAÇÃO	ASPECTO PSICOLÓGICO	PSICOSSOCIAL	FAT HUMA

Nome do fator de geração de problema (15+)

In [173]:
```
dfFator.groupby(['fator_nome']).agg(Nome=('fator_nome', 'count')) \
    .sort_values(by=['Nome'], axis=0, ascending=False)[0:15].plot.barh()
```

Out[173]:
```
<AxesSubplot:ylabel='fator_nome'>
```



Aspecto

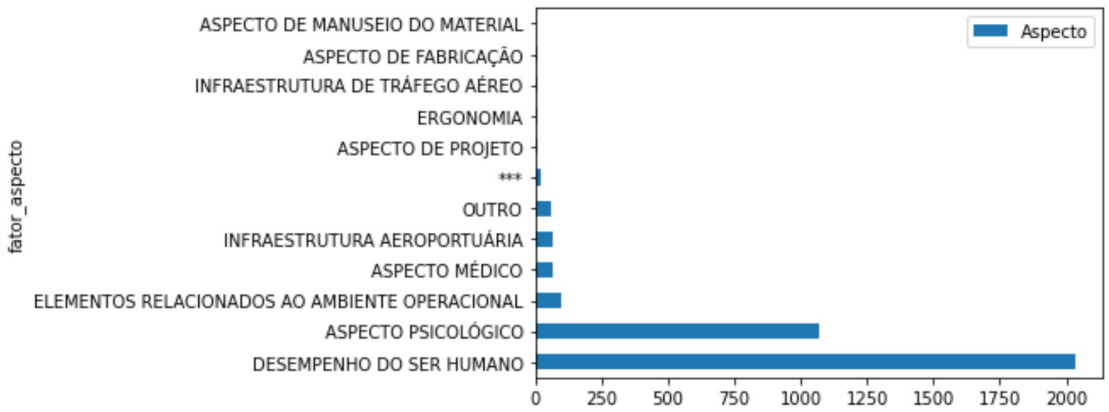
In [300]:
```
dfFator.groupby(['fator_aspecto']) \
.agg(Percentual=('fator_aspecto', 'count')) \
.groupby(level=0).apply(lambda x:  round(x / x.count() / dfFator.sha
pe[0] * 100, 2)  )
```

Out[300]:

fator_aspecto	Percentual
***	0.55
ASPECTO DE FABRICAÇÃO	0.23
ASPECTO DE MANUSEIO DO MATERIAL	0.14
ASPECTO DE PROJETO	0.29
ASPECTO MÉDICO	1.99
ASPECTO PSICOLÓGICO	30.95
DESEMPENHO DO SER HUMANO	58.83
ELEMENTOS RELACIONADOS AO AMBIENTE OPERACIONAL	2.77
ERGONOMIA	0.29
INFRAESTRUTURA AEROPORTUÁRIA	1.91
INFRAESTRUTURA DE TRÁFEGO AÉREO	0.26
OUTRO	1.79

In [171]:
```
dfFator.groupby(['fator_aspecto']).agg(Aspecto=('fator_aspecto', 'co
unt')) \
    .sort_values(by=['Aspecto'], axis=0, ascending=False).plot.barh
()
```

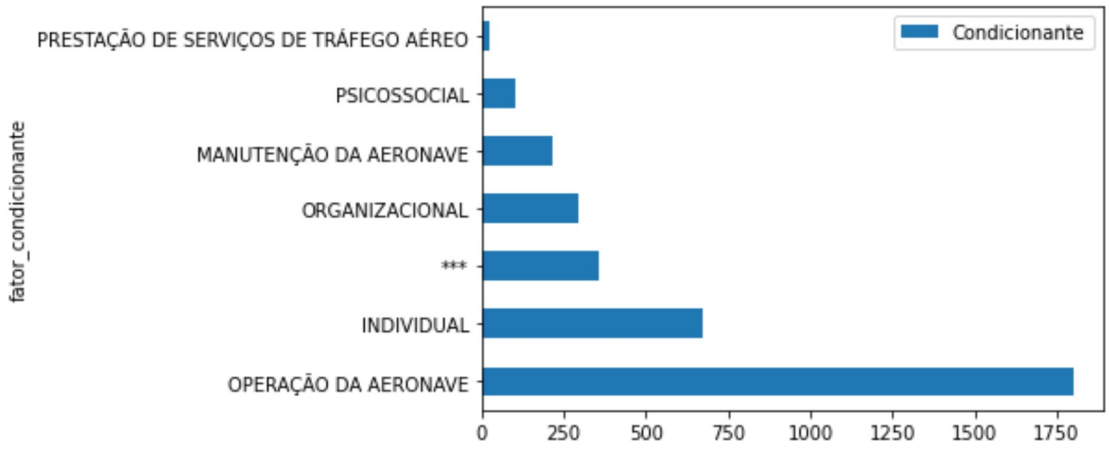
Out[171]: <AxesSubplot:ylabel='fator_aspecto'>



Fator que condicionou o problema

In [169]:
```
dfFator.groupby(['fator_condicionante']).agg(Condicionante=('fator_condicionante', 'count')) \
    .sort_values(by=['Condicionante'], axis=0, ascending=False).plot.barh()
```

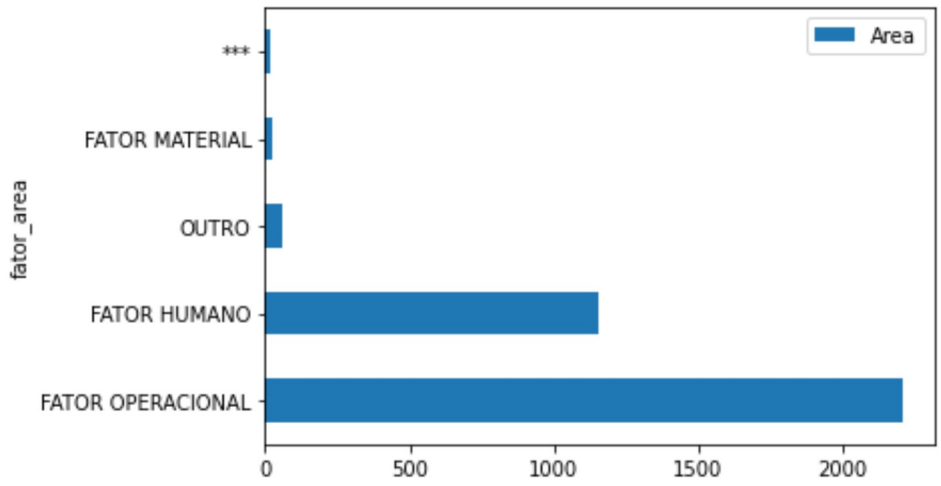
Out[169]: <AxesSubplot:ylabel='fator_condicionante'>



Area

In [170]:
```
dfFator.groupby(['fator_area']).agg(Area=('fator_area', 'count')) \
    .sort_values(by=['Area'], axis=0, ascending=False).plot.barh()
```

Out[170]: <AxesSubplot:ylabel='fator_area'>



In [ ]:

In [ ]:

Tabela Recomendações - Visão Geral

In [183]:
```
dfRecomend.shape
```

Out[183]: (1197, 9)

In [72]:
```
dfRecomend.head(5)
```

Out[72]:

	codigo_ocorrencia4	recomendacao_numero	recomendacao_dia_assinatura	recomendaca
0	45331	A-582/CENIPA/2014 - 01	2016-07-29	
1	45331	A-582/CENIPA/2014 - 02	2016-07-29	
2	45331	A-582/CENIPA/2014 - 03	2016-07-29	
3	45392	A-032/CENIPA/2014 - RSV 001	2014-04-07	
4	45392	A-032/CENIPA/2014 - RSV 002	2014-04-07	

Recomendações mais publicadas (10+)

In [231]:
```
dfRecomend.groupby(['recomendacao_conteudo']).agg(Texto=('recomendacao_conteudo', 'count')) \
    .sort_values(by=['Texto'], axis=0, ascending=False)[0:10]
```

Out[231]:

recomendacao_conteudo	Texto
Divulgar o conteúdo do presente relatório durante a realização de seminários, palestras e atividades afins voltadas aos proprietários, operadores e exploradores de aeronaves de asas rotativas.	5
Divulgar o conteúdo do presente relatório durante a realização de seminários, palestras e atividades afins voltadas aos proprietários, operadores e exploradores de aeronaves agrícolas.	4
Divulgar o conteúdo do presente relatório durante a realização de seminários, palestras e atividades afins voltadas aos proprietários, operadores e exploradores de aeronaves.	2
Divulgar o conteúdo deste Relatório Final a todos os Destacamentos de Controle de Espaço Aéreo.	2
ATUAR JUNTO AO AEROCLUBE DE CANELA, A FIM DE QUE AQUELE OPERADOR, POR OCASIãO DOS TREINAMENTOS OFERECIDOS A SEUS TRIPULANTES, ENFATIZE AS TéCNICAS DE ARREMETIDA E OS FATORES QUE LEVAM à EXECUçãO DESSE PROCEDIMENTO, SOBRETUDO QUANDO FOR CONSTATADO PELA TRIPULAçãO QUE A AERONAVE ESTá EM UMA SITUAçãO NãO ESTABILIZADA.	2
ATUAR JUNTO A FOLIARAVIAçãO AGRíCOLA LTDA., A FIM DE ENFATIZAR àQUELE OPERADOR A IMPORTâNCIA DE SE OBSERVAR O CONSTANTE NO ART. 88-N DO CóDIGO BRASILEIRO DE AERONáUTICA QUE, SALVO NOS CASOS EXCEPCIONAIS ESTABELECIDOS, PROíBE A REMOçãO DE AERONAVES ACIDENTADAS DO LOCAL DA OCORRêNCIA SEM A DEVIDA AUTORIZAçãO DA AUTORIDADE DE INVESTIGAçãO SIPAER.	2
Tendo em vista as condições latentes listadas no conteúdo deste relatório, intensificar as ações de fiscalização no operador da aeronave acidentada.	2
Atuar junto à EJ Escola de Aeronáutica Ltda. ME e ao Aeroclube de Itápolis, a fim de que, conjuntamente, estas instituições realizem uma análise de risco sobre a realização de voos de instrução com aproximações de 180 e 360 graus concomitante a tráfegos executando circuitos normais (perna base e reta final) e “IFR simulados”, de maneira a facilitar a identificação dos perigos e a implementação de medidas mitigadoras adequadas.	2
Dar ampla divulgação do presente relatório entre seus associados, a fim de alertar sobre o problema da montagem invertida do parafuso do amortecedor de vibrações laterais do trem auxiliar e sua respectiva porca, bem como alertar quanto ao perigo na utilização de mão de obra não homologada para serviços de manutenção.	2
Aperfeiçoar de maneira padronizada e sistêmica o briefing diário de alerta ao pessoal de serviço, realizado pelos Supervisores de Equipe, sobre todas as informações que possam ser utilizadas pelos controladores de tráfego aéreo, principalmente nos aeródromos que estejam passando por processo de modificação de infraestrutura aeroportuária.	2

Status de recomendações

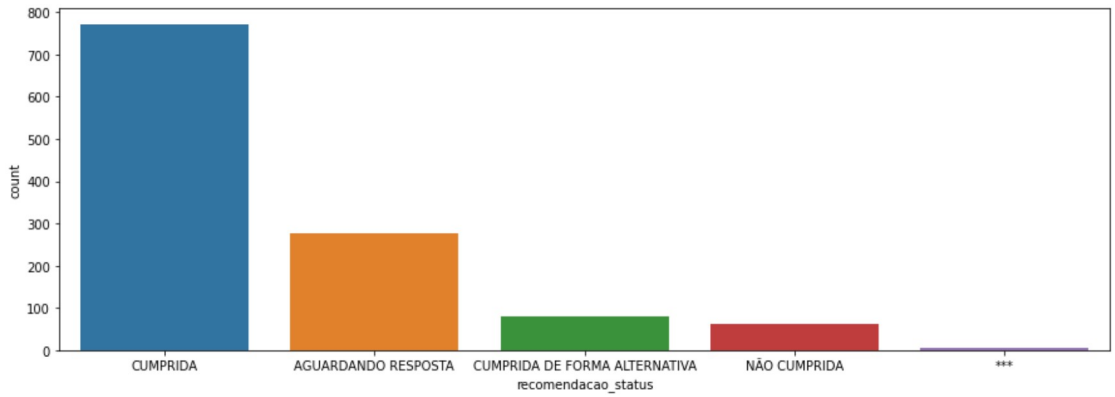
In [299]:
```
dfRecomend.groupby(['recomendacao_status']) \
.agg(Percentual=('recomendacao_status', 'count')) \
.groupby(level=0).apply(lambda x:  round(x / x.count() / dfRecomend.
shape[0] * 100, 2)  )
```

Out[299]:

recomendacao_status	Percentual
***	0.58
AGUARDANDO RESPOSTA	23.22
CUMPRIDA	64.41
CUMPRIDA DE FORMA ALTERNATIVA	6.68
NÃO CUMPRIDA	5.10

In [224]:
```
plt.figure(figsize=(15,5))
srn.countplot(dfRecomend['recomendacao_status'])
```

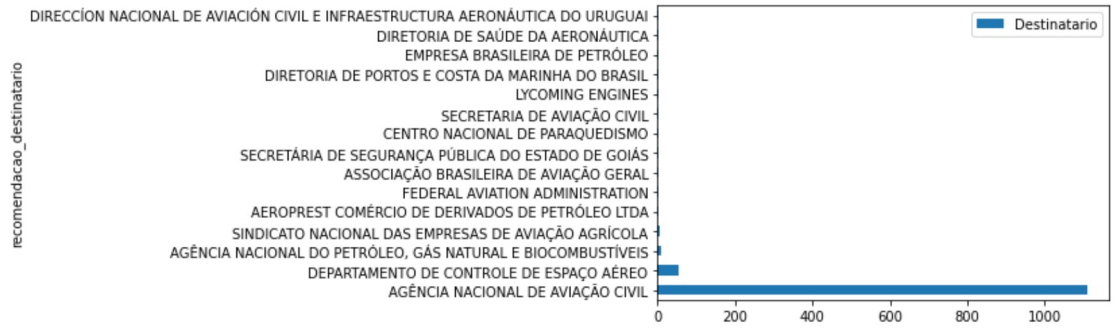
Out[224]: <AxesSubplot:xlabel='recomendacao_status', ylabel='count'>



Destinatário das recomendações

In [232]:
```
dfRecomend.groupby(['recomendacao_destinatario']).agg(Destinatario=
('recomendacao_destinatario', 'count')) \
    .sort_values(by=['Destinatario'], axis=0, ascending=False)[0:1
5].plot.barh()
```

Out[232]: <AxesSubplot:ylabel='recomendacao_destinatario'>



Conclusões & Insigths

Resumo geral da base

- O modelo apresenta um conjunto de 5 tabelas cujo assunto principal é o registro de ocorrências de diversos problemas que acontecem em aeronaves no Brasil
- Neste modelo a tabela de ocorrências é independente sendo referenciada pelas demais podendo ter ou não mais detalhes do problema ocorrido. Os detalhes são:
  - Tipo de ocorrência
  - Especificação da aeronave
  - Fatores que contribuiram para a ocorrência
  - Recomendações que orientam atitudes preventivas para eliminação de problemas
- Cada uma das tabelas com os detalhes fazem referência a tabela principal através de uma coluna com o código correspondente.

Observações

- Mais de 70% das ocorrências são finalizadas sem terem seus relatórios divulgados
- São Paulo concentra a maioria das ocorrências perfazendo um total de 25% dos incidentes
- Falha do motor e do sistemae problemas com pneus estão no topo das ocorrências
- Cessna é o avião que mais gera problemas seguido pela EMBRAER e NEIVA
- Mais de 80% das ocorrências estão fora do Aerodromo sugerindo assim uma abordagem de monitoramento do voo que seja mais eficiente e que possibilite uma intervenção mais ativa
- O modo de operação PRIVADO concentra a maioria das ocorrências
- Apenas 5% acabam destruídas normalmente gerando uma fatalidade
- 80% dos fatores contribuintes envolvem desempenho do ser humano e aspectos psicologicos, abrindo assim espaço para um acompanhamento social mais intensivo junto aos profissionais de pilotagem
- Apesar de 65% das recomendações serem cumpridas, acredito que esta abordagem pode ser melhorada com medidas mais restritivas nos aspectos que propiciam a geração das ocorrências. Passível de análise.

In [ ]: