Maestría en Ciencia de Datos

Calidad e Integración de Datos

Curso 2023

Proyecto Entregable

27/06/2023

Federico Carneiro

Diego Velasco

Introducción

El trabajo tiene como finalidad el análisis e implementación del modelo de Calidad aplicando todas la definiciones trabajadas en el curso. (ver letra_proyecto_parte1 y letra_proyecto_parte_2)
En el documento se pretende detallar todas las determinaciones que vamos a ir tomando de acuerdo a lo visto en clase junto con las definiciones de cada métrica y su aclaración de cuál es el objetivo de la misma, así como también la instancia de cada una sobre cada uno de los tipos de datos a los que las vamos a ir aplicando también dejando algunas puntualizaciones de cada caso.

Descripción general de la fuente de datos

El escenario a trabajar se enfoca en un conjunto de datos referente a viajeros que salen e ingresan al país e información de los operadores turísticos ubicados en el país.
Para eso se nos provee de tres sets de datos, dos en formato csv y uno en formato json.
Los csv corresponden a información de los turistas que salen del país y el segundo a información de los operadores turísticos ubicados en distintas partes del país.
En el data set de turistas que salen del país (emisivos.csv) se pueden observar registros que contienen datos anonimizados de personas que salen del país a través de los distintos puntos de migración ubicados en las distintas fronteras del país. La información es complementada con fecha de salida del país, fecha de retorno, nacionalidad, departamento de residencia, motivo del viaje, ocupación, nivel de estudio, medio de transporte, país de destino, tipo de estadía, gastos generales y algunos gastos diferenciados por su finalidad, como comida, paseos, etc.
Los datos de los operadores turísticos (operadores.csv) contiene información de los distintos operadores que prestan servicios en el país, como pueden ser, inmobiliarias, rentadoras de vehículos, alojamientos, centro turístico, etc. Todos los registros con información de ubicación, como ser departamento, localidad, sitio web, teléfono de contacto, correo electrónico e incluso ubicación geográfica compuesta por latitud y longitud.
El tercer set de datos corresponde a los turistas que ingresan al país (receptivos.json), similar al primero, contiene información anonimizada de las personas que ingresan al país, país de origen, lugar de residencia, nacionalidad, motivo del viaje, nivel de estudio, ocupación, información del destino dentro del país, como ser localidad y departamento, incluso información de un segundo destino dentro del país, fecha de ingreso y fecha de egreso, lugar de ingreso, medio de transporte de ingreso, lugar de egreso y medio de transporte de egreso, información de gastos generales y algunos particulares como alojamiento, transporte, compras en general, cultura, etc.

Data Profiling

Archivo emisivos.csv

Un análisis primario del Archivo nos devuelve algunas mediciones que apuntan a la calidad de algunas columnas.

Medición	**Valor**
Cantidad de Registros (filas)	20602
Cantidad de Columnas	43
Cantidad de Columnas Enteras	14
Cantidad de Columnas Strings	14
Cantidad de Columnas Decimales	15
Registros nulos en “Lugar Salida”	1993
Registros nulos en “IdDeptoResidencia”	2014
Registros nulos en “Departamento”	2014
Registro nulos en “IdMotivo”	610
Registro nulos en “Motivo”	610
Registro “Sin Datos” o “Desconocido / Sin Datos” en “Ocupacion”	230
Registro “Sin Datos” en “Estudio”	31
Registro “Sin Datos” en “Destino”	20
Registro “Sin Datos” en “Alojamiento”	19
Registro “Sin Datos” en “Trasporte Local”	5174
Registros en cero en “IdTranspLocal”	5174
Registros en cero en “GastoTotal”	221
Registros en cero en “GastoAlojamiento”	8705
Registros en cero en “GastoAlimentacion”	945
Registros en cero en “GastoTransporteInternac”	10014
Registros en cero en “GastoTransporteLocal”	4914
Registros en cero en “GastoCultural”	11596
Registros en cero en “GastoTours”	19608

Registros en cero en “GastoCompras”	5355
Registros en cero en “GastoResto”	8459

El Archivo contiene 20602 registros, 43 columnas, en las columnas podemos identificar algunas que tienen algún porcentaje de los registros nulos. También podemos visualizar que algunas columnas pueden ser relacionadas entre sí con la finalidad de detectar el grado de integridad de esos datos.

Para el caso de las columnas **“IdLugarSalida”** y **“Lugar Salida”** realizamos una primera visualización buscando alguna inconsistencia entre ellas y pudimos detectar que para un mismo identificador existen dos valores descriptivos posibles.

IdLugarSalida	Cant_Lugares	Cant_Nulos	Cant_Lugares_Distintos
1	3355	340	2
2	943	76	2
3	5159	484	2
4	994	114	2
5	172	13	2
7	3642	346	2
8	2071	208	2
9	2	0	1
10	2426	229	2
13	668	66	2
14	114	6	2
15	199	20	2
16	234	25	2
17	142	15	2
18	481	51	2

Analizando este aspecto más en detalle, se encuentra que cada valor de IdLugarSalida está asociado a un valor coherente (el cual se asume verdadero) y a valores faltantes. Esto puede verse a continuación.

IdLugarSalida	Lugar Salida
1	Colonia
1	NaN
2	Puerto de montevideo
2	NaN
3	Aeropuerto de Car...
3	NaN
4	Chuy
4	NaN
5	Carmelo
5	NaN
7	Fray Bentos
7	NaN
8	Paysandú
8	NaN
10	Salto
10	NaN
13	Rivera
13	NaN
14	Nueva Palmira
14	NaN

Para los casos de **“IdTranspSalidad”** y **“Transporte Internacional de Salida”**, **“IdNacionalidad”** y **“Pais”**, **“IdMotivo”** y **“Motivo”**, **“IdOcupacion”** y **“Ocupacion”**, **“IdNivelEstudio”** y **“Estudio”**, **“IdDestino”** y **“Destino”**, **“IdAlojamiento”** y **“Alojamiento”**, **“IdLugarIngreso”** y **“Lugar Ingreso”**, **“IdTranspIngreso”** y **“Transporte Internacional de Ingreso”**, **“IdTranspLocal”** y **“Trasporte Local”**, **“IdFecEntrada”** y **“FechaEntrada”,** cada identificador se corresponde con un solo campo descriptivo.

IdTranspSalidad	Cant_TranspSalida	Cant_Nulos	Cant_TranspSalida_Distintos
1	5162	0	1
2	8045	0	1
3	2796	0	1
4	21	0	1
5	4570	0	1
6	8	0	1

IdNacionalidad	Cant_Nacionalidad	Cant_Nulos	Cant_Nacionalidad_Distintos
1	20102	0	1
10	334	0	1
18	2	0	1
19	57	0	1
30	17	0	1
32	6	0	1
33	1	0	1
35	2	0	1
36	8	0	1
40	17	0	1
42	1	0	1
43	2	0	1
49	9	0	1
51	26	0	1
52	8	0	1
53	4	0	1
54	1	0	1
59	2	0	1
69	1	0	1
70	2	0	1

IdMotivo	Cant_Motivos	Cant_Nulos	Cant_Motivos_Distintos
1.0	9362	0	1
2.0	6905	0	1
3.0	2468	0	1
4.0	19	0	1
5.0	183	0	1
6.0	134	0	1
7.0	144	0	1
8.0	128	0	1
9.0	201	0	1
10.0	448	0	1
NaN	610	610	1

IdOcupacion	Cant_Ocupaciones	Cant_Nulos	Cant_Ocupaciones_Distintos
0	33	0	1
1	1824	0	1
2	1002	0	1
3	662	0	1
4	123	0	1
5	10	0	1
8	59	0	1
9	197	0	1
11	2849	0	1
12	360	0	1
13	3701	0	1
14	807	0	1
15	1449	0	1
16	6893	0	1
17	90	0	1
18	77	0	1
19	27	0	1
20	144	0	1
21	163	0	1
22	94	0	1
99	38	0	1

IdNivelEstudio	Cant_Estudio	Cant_Nulos	Cant_Estudio_Distintos
0	31	0	1
1	3	0	1
2	234	0	1
3	2661	0	1
4	7092	0	1
5	4802	0	1
6	5096	0	1
7	683	0	1

IdDestino	Cant_Destino	Cant_Nulos	Cant_Destino_Distintos
0	20	0	1
10	13166	0	1
20	4053	0	1
30	434	0	1
31	372	0	1
32	377	0	1
33	1200	0	1
40	851	0	1
50	24	0	1
60	71	0	1
61	29	0	1
62	2	0	1
70	3	0	1

IdAlojamiento	Cant_Alojamiento	Cant_Nulos	Cant_Alojamiento_Distintos
0	19	0	1
1	9625	0	1
2	490	0	1
3	127	0	1
4	91	0	1
5	1181	0	1
6	8283	0	1
7	13	0	1
8	2	0	1
9	21	0	1
10	20	0	1
11	425	0	1
12	305	0	1

IdLugarIngreso	Cant_Lugar_Ingreso	Cant_Nulos	Cant_Lugar_Ingreso_Distintos
1	3256	0	1
2	940	0	1
3	5168	0	1
4	986	0	1
5	157	0	1
7	3250	0	1
8	2649	0	1
10	2354	0	1
13	691	0	1
14	114	0	1
15	205	0	1
16	227	0	1
17	139	0	1
18	466	0	1

IdTranspIngreso	Cant_Transporte_Internacional_Ingreso	Cant_Nulos	Cant_Transporte_Internacional_Ingreso_Distintos
1	5169	0	1
2	7851	0	1
3	3087	0	1
4	24	0	1
5	4467	0	1
6	4	0	1

IdTranspLocal	Cant_Transporte_Local	Cant_Nulos	Cant_Transporte_Local_Distintos
0	5174	0	1
1	6538	0	1
2	389	0	1
3	4854	0	1
4	2667	0	1
5	95	0	1
6	885	0	1

Para el caso de “IdDeptoResidencia” y “Departamento”, si bien cada valor de la primera columna se corresponde con un único valor de la segunda columna, existen valores nulos de la primera que se corresponden con valores nulos de la segunda.

IdDeptoResidencia	Cant_Departamento	Cant_Nulos	Cant_Departamento_Distintos
NULL	0	2014	0
10.0	608	0	1
11.0	784	0	1
12.0	757	0	1
13.0	560	0	1
20.0	2278	0	1
21.0	8266	0	1
30.0	177	0	1
31.0	665	0	1
32.0	131	0	1
40.0	168	0	1
41.0	180	0	1
42.0	170	0	1
43.0	528	0	1
51.0	466	0	1
52.0	207	0	1
53.0	335	0	1
54.0	155	0	1
60.0	1213	0	1
61.0	940	0	1

Dado que existen 19 valores distintos de IdDeptoResidencia y que efectivamente solo existen 19 departamentos distintos, es sencillo evaluar si los departamentos están ingresados correctamente. Los departamentos distintos ingresados pueden verse a continuación.

Departamento
Montevideo
Maldonado
Canelones
Salto
Lavalleja
Saqn Jose
Rio Negro
NULL
Durazno
Cerro Largo
Colonia
Tacuarembo
Treinta y Tres
Rocha
Artigas
Paysandu
Rivera
Flores
Soriano
Florida

Es posible ver que, además de datos nulos, existe un error de sintaxis en el departamento de San José. Dado que este es el único error de exactitud en este atributo, el mismo será evaluado como exactitud sintáctica en la siguiente sección.

Para “IdFecSalida” y “FechaSalida” pasa algo similar a “Lugar Salida” e “IdLugarSalida”, para algunos casos para cada Id de Fecha tenemos dos valores descriptivos, es necesario revisar si es un tema de valores diferentes, valores sin ingresar o si se tratan de formatos distintos para que representen a una misma fecha.

IdFecSalida	Cant_FechaSalida	Cant_Nulos	Cant_FechaSalida_Distintos
13496	1	0	1
13501	2	0	1
13502	1	0	1
13503	1	0	1
13504	2	0	1
13505	2	0	1
13506	4	0	2
13507	7	0	1
13508	6	0	1
13509	9	0	1
13510	8	0	1
13511	18	0	2
13512	21	0	2
13513	39	0	2
13514	11	0	1
13515	9	0	2
13516	23	0	2
13517	20	0	2
13518	14	0	2
13519	24	0	2
13520	35	0	2

Nuevamente, analizando en mayor profundidad este aspecto, puede verse que la duplicidad de valores distintos de FechaSalida asociados a un mismo valor de IdFechaSalida se deben a incoherencias en el formato de dicha fecha. Esto puede verse a continuación.

IdFecSalida	FechaSalida
13506	12/23/2016
13506	2016-12-23
13511	2016-12-28
13511	12/28/2016
13512	2016-12-29
13512	12/29/2016
13513	2016-12-30
13513	12/30/2016
13515	2017-01-01
13515	01/01/2017
13516	2017-01-02
13516	01/02/2017
13517	2017-01-03
13517	01/03/2017
13518	2017-01-04
13518	01/04/2017
13519	2017-01-05
13519	01/05/2017
13520	2017-01-06
13520	01/06/2017

Para poder realizar una medición de la exactitud sintáctica de este atributo, se debe, en primer lugar, definir un formato de fecha correcto en base al cual se realizará la comparación. Para definir dicho formato, se evalúa en primer lugar cuál es el formato

predominante, para luego realizar la comparación contra dicho formato. Esto puede verse en la siguiente imagen.

cant_fechas_con_barra	cant_fechas_con_guion	cant_fecha_total
3098	17504	20602

Dado que, del total de registros, 3098 fechas son ingresadas en formato MM/DD/AAAA y los restantes 17504 son ingresados en formato AAAA-MM-DD, se toma el segundo como válido para definir la métrica de calidad. Dicha métrica es definida formalmente en la siguiente sección.

En esta línea, dado que se cuenta con las fechas de entrada y salida así como con la duración de la estadía de cada turista, es posible verificar la dependencia funcional entre estos tres atributos. Dado que las fechas son ingresadas como textos y que las fechas de salida tienen formatos distintos, se debe convertir todas a un formato de fecha común para luego operar con ellas. Procediendo de esta manera, se genera la siguiente tabla auxiliar:

index	FechaEntrada	FechaSalida	Fechas_dif	estadia
0	2017-01-10 00:00:00	2016-12-29 00:00:00	12	12
1	2017-01-09 00:00:00	2016-12-29 00:00:00	11	11
2	2017-01-27 00:00:00	2017-01-03 00:00:00	24	24
3	2017-01-21 00:00:00	2017-01-06 00:00:00	15	15
4	2017-01-25 00:00:00	2017-01-02 00:00:00	23	23
5	2017-01-09 00:00:00	2016-12-30 00:00:00	10	10
6	2017-01-03 00:00:00	2016-12-30 00:00:00	4	4
7	2017-01-11 00:00:00	2017-01-05 00:00:00	6	6
8	2017-01-09 00:00:00	2016-12-30 00:00:00	10	10
9	2017-01-07 00:00:00	2016-12-18 00:00:00	20	20
10	2017-01-16 00:00:00	2017-01-10 00:00:00	6	6
11	2017-01-25 00:00:00	2017-01-19 00:00:00	6	6
12	2017-01-09 00:00:00	2017-01-06 00:00:00	3	3
13	2017-01-07 00:00:00	2016-12-27 00:00:00	11	11

Luego, se cuentan los registros para los cuales la columna de estadía y de Fechas_dif no coinciden:

Cant_NO_COINCIDEN	Cant_COINCIDEN
211	20391

En base a esto, se define de manera formal una métrica de consistencia en la siguiente sección.

También resulta de interés investigar la precisión con la que se especifica el lugar de salida, de ingreso, el destino y la nacionalidad de los turistas. Los valores de estos atributos son los siguientes:

Lugar Salida
Chuy
NULL
Aeropuerto de Carrasco
Río Branco
Fray Bentos
Bella Unión
Colonia
Salto
Paysandú
Carmelo
Melo - Aceguá
Artigas
Rivera
Puerto de montevideo
Nueva Palmira
Otros

Destino
Brasil
Europa
Argentina
Chile
Resto Sud America
Centro y Norte America
Asia del Este y Pacifico
Oriente Medio
Paraguay
Africa
Sin Datos
Otros
Asia Meridional

Lugar Ingreso
Chuy
Río Branco
Aeropuerto de Carrasco
Fray Bentos
Bella Unión
Colonia
Salto
Paysandú
Carmelo
Melo - Aceguá
Artigas
Rivera
Puerto de montevideo
Nueva Palmira

Pais
Uruguay
Brasil
Argentina
Otro de Europa
España
EE.UU.
Italia
Paraguay
Chile
Ecuador
Venezuela
Bolivia
Otro de America
Peru
Gran Bretaña
Colombia
Mexico
Francia
Otro pais de Asia
Cuba

Es posible ver que el atributo `Lugar Salida` tiene valores nulos y valores clasificados como “otros”. Todos los valores únicos de esta columna deberían coincidir con los valores únicos de “Lugar Ingreso”, ya que, si bien un turista puede salir y entrar al país por lugares distintos, los lugares de ingreso y de salida son los mismos. Esto da lugar a una métrica asociada a ambas columnas, la cual es presentada en la sección siguiente.

Asimismo, las variables Destino y Pais tienen una precisión variable. Esta dimensión de calidad será evaluada en la sección siguiente.

En esta línea, los lugares de salida y de ingreso están relacionados con los atributos de “Transporte Internacional de Salida” y “Transporte Internacional de Ingreso” respectivamente. Particularmente, los registros cuyo valor de “Lugar Salida” o “Lugar Ingreso” son “Aeropuerto de Carrasco” o “Puerto de montevideo” deberían tener valores de transporte de salida y de ingreso “Aereo” y “Maritimo - Fluvial” respectivamente. Sin embargo, puede verse a continuación que esto no siempre ocurre:

Lugar Salida	Transporte Internacional de Salida
Puerto de montevideo	Aereo
Puerto de montevideo	Aereo
Puerto de montevideo	Aereo

Lugar Salida	Transporte Internacional de Salida
Aeropuerto de Carrasco	Terrestre Bus
Aeropuerto de Carrasco	Terrestre Auto
Aeropuerto de Carrasco	Terrestre Auto

Esto da lugar a una métrica de consistencia definida formalmente en la siguiente sección.

También existe una dependencia visible a comprobar para las columnas “GastoTotal” y “GastoAlojamiento”,”GastoAlimentacion”,”GastoTransporteInternac”,”GatoTransporteLocal”,” GastoCultural”,”GastoTours”,”GastoCompras”,”GastoResto” y verificar que la primera es la suma de todas las demás.

te	GastoTotal	GastoAlojamiento	GastoAlimentacion	GastoTransporteInternac	GatoTransporteLocal	GastoCultural	GastoTours	GastoCompras	GastoResto	
5	1507.12	981.39	350.5	0.0	175.24	0.0	0.0	0.0	0.0	8
5	3275.0	1100.0	880.0	0.0	385.0	300.0	235.0	210.0	165.0	9
2	4785.75	0.0	1129.75	214.0	836.85	523.03	256.29	2039.83	0.0	12
3	5883.0	2228.41	1782.73	117.0	779.94	445.68	0.0	311.98	334.26	12
5	3400.0	1800.0	933.12	0.0	408.24	152.14	0.0	106.5	0.0	8
5	3500.0	1400.0	1200.0	0.0	311.88	267.33	0.0	187.13	133.66	9
3	2836.16	0.0	473.17	283.9	120.0	84.12	0.0	262.87	1896.0	12

Dado que los gastos son ingresados con dos números decimales, se realiza una comparación con los valores redondeados al entero más cercano. Dicha comparación puede verse a continuación:

COINCIDEN	NO_COINCIDEN
9997	10605

Esto da lugar a una métrica de consistencia intra-relación definida formalmente en la siguiente sección

Respecto a los valores de latitud y longitud, resulta de interés evaluar si los mismos corresponden a valores posibles. En primer lugar, se detecta los valores máximos y mínimos de estas dos columnas

max(latitud)	min(latitud)	max(longitud)	min(longitud)
-30.7800110167659	-34.9718851931481	-53.2578316224576	-58.427848861416

Asimismo, se consulta la fuente de metadatos de la base de datos original, obtenida de la web del catálogo de Datos Abiertos de Uruguay, pero la misma no cuenta con explicación sobre a qué lugares corresponden las latitudes y longitudes indicadas en la base de datos. Sin embargo, dado que el rango de latitudes y longitudes en el cual se encuentra Uruguay es aproximadamente [-30.130153; -35.061084] y [-53.082044; -58.483502] respectivamente, y que es posible ver que los datos de latitud y longitud entran dentro de este rango, se deduce que estos atributos deben corresponder a la latitud y longitud del punto de salida o ingreso al país, o al departamento del encuestado. Resulta de interés, entonces, evaluar el

rango de latitud y longitud según estos atributos. A continuación se presentan estos rangos para los datos agrupados por lugar de salida, ingreso y departamento respectivamente.

Lugar Salida	Lugar Ingreso	max(latitud)	min(latitud)	max(longitud)	min(longitud)
NULL	Colonia	-30.7828042969024	-34.9401489777936	-53.375232682856	-58.4112013986486
Aeropuerto de Carrasco	Aeropuerto de Carrasco	-30.7800110167659	-34.9654588797914	-53.3674621580633	-58.4162439203056
Artigas	Rivera	-30.7820507977473	-34.8873961880012	-54.1653528019211	-58.0898611658771
Bella Unión	Bella Unión	-30.7874619615651	-34.871395796863	-55.4142607394362	-58.3129610713982
Carmelo	Carmelo	-33.2490559470216	-34.917569693407	-55.2720502464819	-58.4072776894985
Chuy	Chuy	-32.6967406621622	-34.9718851931481	-53.3893447072341	-58.3548798086728
Colonia	Colonia	-30.9056047711134	-34.9643591433398	-54.1580169229756	-58.4188827800768
Fray Bentos	Fray Bentos	-30.8892347478265	-34.9706320449695	-53.7835643950164	-58.427848861416
Melo - Aceguá	Rivera	-31.6971818074058	-34.9099749392289	-53.2578316224576	-57.0147457732781
Nueva Palmira	Nueva Palmira	-33.2448415206885	-34.8984666426871	-56.1891276347729	-58.4134553559783
Otros	Aeropuerto de Carrasco	-34.8501153458452	-34.8842878293783	-56.1242971403839	-56.2753566439657
Paysandú	Fray Bentos	-30.8748132783127	-34.9541723816559	-53.3732634408203	-58.4166858186629
Puerto de montevideo	Puerto de montevideo	-32.5736211377957	-34.9280114925982	-53.376416383807	-58.2598565529521
Rivera	Rivera	-30.873388448843	-34.9281850307158	-54.1498317784352	-58.160824270323
Río Branco	Río Branco	-31.9574858716784	-34.9245432403666	-53.2594856616288	-58.240580210748
Salto	Salto	-30.784652327494	-34.9322165408354	-53.4668387192126	-58.3296802447509

Lugar Salida	Lugar Ingreso	max(latitud)	min(latitud)	max(longitud)	min(longitud)
Aeropuerto de Carrasco	Aeropuerto de Carrasco	-30.7800110167659	-34.9654588797914	-53.3674621580633	-58.4162439203056
Artigas	Artigas	-30.7820507977473	-34.8873961880012	-54.1653528019211	-58.069057323044
Bella Unión	Bella Unión	-30.7874619615651	-34.871395796863	-54.9230902004299	-58.3129610713982
Carmelo	Carmelo	-33.3746118373503	-34.917569693407	-55.2720502464819	-58.4072776894985
Chuy	Chuy	-31.0979877097296	-34.9718851931481	-53.3893447072341	-58.3548798086728
Colonia	Colonia	-30.9056047711134	-34.9643591433398	-53.4541007730768	-58.4188827800768
Fray Bentos	Fray Bentos	-30.8892347478265	-34.9706320449695	-53.7835643950164	-58.427848861416
Melo - Aceguá	Melo - Aceguá	-31.6971818074058	-34.9099749392289	-53.2578316224576	-56.5247162535721
Nueva Palmira	Nueva Palmira	-33.2448415206885	-34.8984666426871	-56.1891276347729	-58.4134553559783
Paysandú	Paysandú	-30.7856250350781	-34.9541723816559	-53.3732634408203	-58.4136226752598
Puerto de montevideo	Puerto de montevideo	-32.5736211377957	-34.9280114925982	-53.376416383807	-58.2598565529521
Rivera	Rivera	-30.873388448843	-34.9281850307158	-54.1498317784352	-58.160824270323
NULL	Río Branco	-31.9574858716784	-34.9245432403666	-53.2594856616288	-58.2536678682981
Salto	Salto	-30.7828042969024	-34.9458495889541	-53.4668387192126	-58.3296802447509

Departamento	max(latitud)	min(latitud)	max(longitud)	min(longitud)
NULL	-30.7867953411933	-34.9597347518545	-53.2617062638109	-58.4172508604659
Artigas	-34.2862845970606	-34.8759676745883	-55.4003511411974	-56.4140203054701
Canelones	-34.2212449531458	-34.8770963124	-55.3980095280545	-56.4687826245541
Cerro Largo	-31.8744886912007	-32.8323339400207	-53.2578316224576	-55.1919404794281
Colonia	-33.8044915313828	-34.4732757208786	-57.0947539795062	-58.4188827800768
Durazno	-32.7207468232094	-33.4254089823821	-55.1345967725769	-57.0017676490295
Flores	-33.1365516694766	-33.9667482940994	-56.7445356073514	-57.1540506690572
Florida	-33.1073099854048	-34.4135749859081	-55.1324297934675	-56.413459405268
Lavalleja	-33.4483046826407	-34.602502965102	-54.5270827367428	-55.5083640409041
Maldonado	-34.1987408510311	-34.9718851931481	-54.5454165043224	-55.4244597748255
Montevideo	-34.7021832948793	-34.9281850307158	-56.0328299999999	-56.4072474231629
Paysandu	-31.581768355181	-32.5309369262311	-56.2399944857185	-58.160959386816
Rio Negro	-32.3577671069041	-33.2697889942656	-56.6967736617233	-58.3576621539532
Rivera	-30.873388448843	-31.8862706221374	-54.6857839125792	-55.9997649171326
Rocha	-33.2687432197576	-34.675667129598	-53.3893447072341	-54.5148626770602
Salto	-30.7800110167659	-31.7815100704664	-56.2190248982258	-58.0491006268428
Saqn Jose	-33.9929685976564	-34.7762297511383	-56.3577660183163	-57.14914734349
Soriano	-33.0574676422912	-33.8731825746505	-57.3045256204378	-58.427848861416
Tacuarembo	-31.5084950243635	-32.8180320545598	-54.8524543380037	-56.5371871769685
Treinta y Tres	-32.7613789631684	-33.2633062270424	-53.5179704002372	-55.1315326547

Es posible ver que los rangos agrupados según lugares de salida e ingreso al país son muy similares al rango total de Uruguay, de manera que los valores de latitud y longitud no deben corresponder a dichos lugares.
Sin embargo, al agrupar por departamento, se puede ver que cada departamento presenta rangos más acotados. Asimismo, es posible apreciar que los departamentos más al sur y norte presentan rangos de latitud cercanos al límite inferior y superior del rango de Uruguay respectivamente. Lo mismo ocurre con la longitud, dado que departamentos más al este y oeste presentan rangos cercanos al límite superior e inferior respectivamente. Igualmente, existen inconsistencias. A modo de ejemplo, Artigas presenta un rango similar de latitudes al de Montevideo, cuando los mismos deberían estar contra los extremos opuestos del rango de Uruguay. En virtud de esto, se evalúa la integridad de dominio de estos dos atributos en la sección siguiente.

Asimismo, se evalúa la presencia de duplicados en la tabla en cuestión. Dado que, a priori, no parece que exista una clave primaria para cada registro, se comienza buscando duplicados exactos, es decir, que exista el mismo registro dos veces de manera idéntica.

```
subset_columns = list(df.columns)
subset_columns.pop(subset_columns.index('FechaSalida')) # Se saca esta por tener distintos formatos para un mismo ID
subset_columns.pop(subset_columns.index('Lugar Salida')) # Se saca esta por tener distintos valores para un mismo ID

df[df.duplicated(subset=subset_columns, keep=False)].shape
```

✓ 0.1s

```
(0, 43)
```

Dado que no existen registros idénticos duplicados, se realiza la búsqueda según los atributos de gastos, ya que estos son los que tienen la mayor cantidad de valores únicos, además de que, dado que la precisión de los valores de gastos es de dos números decimales, resulta muy improbable que dos registros tengan valores idénticos para todos los gastos de manera simultánea. A continuación se muestran algunas filas de la tabla filtrada por las filas cuyos valores de gastos se repiten, dejando afuera las filas con todos los gastos nulos.

	IdLugarSalida	Lugar Salida	IdTranspSalidad	Transporte Internacional de Salida	FechaSalida	IdFecSalida	FechaEntrada	IdFecEntrada	IdNacionalidad	Pais
2685	2	Puerto de montevideo	5	Maritimo - Fluvial	2017-05-26	13660	2017-05-29	13663	1	Uruguay
2686	7	Fray Bentos	3	Terrestre Bus	2017-05-01	13635	2017-05-05	13639	1	Uruguay
13320	8	Paysandú	2	Terrestre Auto	2019-08-06	14462	2019-08-08	14464	1	Uruguay
13321	8	Paysandú	2	Terrestre Auto	08/06/2019	14462	2019-08-08	14464	10	Argentina
2120	7	Fray Bentos	3	Terrestre Bus	2017-04-13	13617	2017-04-17	13621	1	Uruguay
2121	7	NaN	3	Terrestre Bus	04/14/2017	13618	2017-04-17	13621	1	Uruguay

IdDeptoResidencia	Departamento	IdMotivo	Motivo	IdOcupacion	Ocupacion	IdNivelEstudio	Estudio	IdDestino	Destino
21.0	Montevideo	2.0	Visita familiares / amigos	16	Empl. Adm, Cajero, Vendedor	4	Secundaria completa	10	Argentina
54.0	Treinta y Tres	5.0	Tratamiento Salud	16	Empl. Adm, Cajero, Vendedor	4	Secundaria completa	10	Argentina
61.0	Paysandu	2.0	Visita familiares / amigos	13	Prof, Tecnico, Docente, Artista	6	Terciaria completa	10	Argentina
61.0	Paysandu	2.0	Visita familiares / amigos	16	Empl. Adm, Cajero, Vendedor	5	Terciaria incompleta	10	Argentina
21.0	Montevideo	1.0	Ocio, Recreo, Vacaciones	16	Empl. Adm, Cajero, Vendedor	6	Terciaria completa	10	Argentina
13.0	Soriano	2.0	Visita familiares / amigos	13	Prof, Tecnico, Docente, Artista	6	Terciaria completa	10	Argentina

	IdAlojamiento	Alojamiento	IdLugarIngreso	Lugar Ingreso	IdTranspIngreso	Transporte Internacional de Ingreso	IdTranspLocal	Trasporte Local	Estadia	Gente	GastoTotal
2685	6	Vivienda Familiares / Amigos	2	Puerto de montevideo	5	Maritimo - Fluvial	0	Sin Datos	3	2	190.96
2686	6	Vivienda Familiares / Amigos	7	Fray Bentos	3	Terrestre Bus	0	Sin Datos	4	1	190.96
13320	6	Vivienda Familiares / Amigos	8	Paysandú	2	Terrestre Auto	1	Auto propio	2	2	300.00
13321	6	Vivienda Familiares / Amigos	8	Paysandú	2	Terrestre Auto	1	Auto propio	2	2	300.00
2120	6	Vivienda Familiares / Amigos	7	Fray Bentos	3	Terrestre Bus	0	Sin Datos	4	2	200.00
2121	6	Vivienda Familiares / Amigos	7	Fray Bentos	3	Terrestre Bus	0	Sin Datos	3	2	200.00

GastoAlojamiento	GastoAlimentacion	GastoTransporteInternac	GatoTransporteLocal	GastoCultural	GastoTours	GastoCompras	GastoResto	Coef
0.0	66.54	0.0	0.00	0.0	0.0	0.0	124.41	161.49
0.0	66.54	0.0	0.00	0.0	0.0	0.0	124.41	81.96
0.0	67.44	0.0	108.79	0.0	0.0	0.0	123.77	211.47
0.0	67.44	0.0	108.79	0.0	0.0	0.0	123.77	107.31
0.0	68.57	0.0	31.43	0.0	0.0	0.0	100.00	107.10
0.0	68.57	0.0	31.43	0.0	0.0	0.0	100.00	107.10

	CoefTot	latitud	longitud
2685	322.98	-34.818097	-56.137911
2686	81.96	-33.233454	-54.390465
13320	422.95	-32.360346	-57.201348
13321	214.62	-31.932027	-57.889241
2120	214.21	-34.917569	-56.152041
2121	214.21	-33.864404	-57.371208

Si se analiza, por ejemplo, los registros 13320 y 13321, puede verse que todos los atributos coinciden, con excepción de “Pais”, “Ocupacion” y “Estudio”, “CoefTot”, “latitud” y “longitud”. Asimismo, la cantidad de gente total con la que viaja el encuestado es de 2. Esto sugiere que los registros puedan corresponder a dos personas que viajan juntas, siendo ambos encuestados. Si este fuera el caso, y los datos de gastos correspondieran al total de gastos del grupo del encuestado, se entiende que tales registros constituirían un duplicado, ya que se tendría dos veces valores de gastos asociados a un mismo grupo de gente.

Sin embargo, dado que no se cuenta con información sobre a qué corresponde específicamente estas columnas, no se puede asumir que este tipo de registros represente un duplicado. Por esto, y en virtud de la gran precisión que presentan los valores de “latitud” y “longitud” se considera que un registro será duplicado si se repiten los valores para todas las columnas exceptuando “Coef”, “CoefTot”, “latitud”, “longitud”, “FechaSalida” y “LugarSalida”. Cabe destacar que las últimas dos columnas no se toman en cuenta porque, como se vio anteriormente, presentan inexactitudes sintácticas e inconsistencias con la columna de Id respectivamente. En cambio, se consideran únicamente las columnas de “IdFecSalida” y “IdLugarSalida”. A continuación se presentan algunas filas que presentan duplicados bajo estas condiciones.

	IdLugarSalida	Lugar Salida	IdTranspSalidad	Transporte Internacional de Salida	FechaSalida	IdFecSalida	FechaEntrada	IdFecEntrada	IdNacionalidad	Pais
6365	7	Fray Bentos	3	Terrestre Bus	2017-11-21	13839	2017-11-27	13845	1	Uruguay
6366	7	Fray Bentos	3	Terrestre Bus	2017-11-21	13839	2017-11-27	13845	1	Uruguay
6358	7	Fray Bentos	3	Terrestre Bus	11/22/2017	13840	2017-11-27	13845	1	Uruguay
6362	7	Fray Bentos	3	Terrestre Bus	2017-11-22	13840	2017-11-27	13845	1	Uruguay
18809	7	Fray Bentos	3	Terrestre Bus	2022-11-30	15674	2022-12-03	15677	10	Argentina
19391	7	Fray Bentos	3	Terrestre Bus	2022-11-30	15674	2022-12-03	15677	10	Argentina

IdDeptoResidencia	Departamento	IdMotivo	Motivo	IdOcupacion	Ocupacion	IdNivelEstudio	Estudio	IdDestino	Destino
21.0	Montevideo	1.0	Ocio, Recreo, Vacaciones	16	Empl. Adm, Cajero, Vendedor	5	Terciaria incompleta	10	Argentina
21.0	Montevideo	1.0	Ocio, Recreo, Vacaciones	16	Empl. Adm, Cajero, Vendedor	5	Terciaria incompleta	10	Argentina
21.0	Montevideo	1.0	Ocio, Recreo, Vacaciones	16	Empl. Adm, Cajero, Vendedor	5	Terciaria incompleta	10	Argentina
21.0	Montevideo	1.0	Ocio, Recreo, Vacaciones	16	Empl. Adm, Cajero, Vendedor	5	Terciaria incompleta	10	Argentina
41.0	Flores	9.0	Estudios	13	Prof, Tecnico, Docente, Artista	6	Terciaria completa	10	Argentina
41.0	Flores	9.0	Estudios	13	Prof, Tecnico, Docente, Artista	6	Terciaria completa	10	Argentina

	IdAlojamiento	Alojamiento	IdLugarIngreso	Lugar Ingreso	IdTranspIngreso	Transporte Internacional de Ingreso	IdTranspLocal	Trasporte Local	Estadia	Gente	GastoTotal
6365	1	Hotel	7	Fray Bentos	3	Terrestre Bus	3	Taxi - Bus	6	2	1510.14
6366	1	Hotel	7	Fray Bentos	3	Terrestre Bus	3	Taxi - Bus	6	2	1510.14
6358	1	Hotel	7	Fray Bentos	3	Terrestre Bus	3	Taxi - Bus	5	2	1289.83
6362	1	Hotel	7	Fray Bentos	3	Terrestre Bus	3	Taxi - Bus	5	2	1289.83
18809	6	Vivienda Familiares / Amigos	1	Colonia	5	Maritimo - Fluvial	3	Taxi - Bus	3	1	50.00
19391	6	Vivienda Familiares / Amigos	1	Colonia	5	Maritimo - Fluvial	3	Taxi - Bus	3	1	50.00

GastoAlojamiento	GastoAlimentacion	GastoTransporteInternac	GatoTransporteLocal	GastoCultural	GastoTours	GastoCompras	GastoResto	Coef
521.09	285.71	35.13	125.50	0.0	0.0	188.32	389.52	65.32
521.09	285.71	35.13	125.50	0.0	0.0	188.32	389.52	65.32
434.24	238.09	35.13	104.58	0.0	0.0	188.32	324.60	65.32
434.24	238.09	35.13	104.58	0.0	0.0	188.32	324.60	65.32
0.00	20.00	120.00	30.00	0.0	0.0	0.00	0.00	535.66
0.00	20.00	120.00	30.00	0.0	0.0	0.00	0.00	79.35

	CoefTot	latitud	longitud
6365	130.64	-34.895308	-56.186726
6366	130.64	-34.844206	-56.169660
6358	130.64	-34.916655	-56.164114
6362	130.64	-34.875691	-56.104941
18809	535.66	-33.513556	-56.906042
19391	78.43	-33.525245	-56.908703

Como se desprende de las imágenes presentadas, estos registros presentan contradicciones en las columnas “Coef”, “CoefTot”, “latitud” y/o “longitud”. Existen 245 registros con estas características. A su vez, no existen datos duplicados ni nulos de estas columnas para los mencionados 245 registros, de manera que todos los registros corresponden a valores contradictorios. Esto es utilizado para definir formalmente la métrica de unicidad en la sección siguiente.

Por último, resta evaluar la densidad de la base de datos. Como se vio anteriormente, las columnas “Lugar Salida”, “IdDeptoResidencia”, “Departamento”, “IdMotivo” y “Motivo” cuentan con valores nulos. Asimismo, las columnas “Ocupacion”, “Estudio”, “Destino”, “Alojamiento” y “Trasporte Local” cuentan con registros “Sin Datos”, mientras que la

columna “Ocupacion” también cuenta con registros “Desconocido / Sin Datos”. En la sección siguiente se define una métrica a nivel de celda para cuantificar este aspecto.

Archivo operadores.csv

Un análisis primario del Archivo nos devuelve algunas mediciones que apuntan a la calidad de algunas columnas.

Medición	**Valor**
Cantidad de Registros (filas)	3288
Cantidad de Columnas	10
Cantidad de Columnas Enteras	0
Cantidad de Columnas Strings	8
Cantidad de Columnas Decimales	0
Cantidad de Columnas Coordenadas Geográficas	2

En la visualización primaria de los datos podemos ver que existen algunas columnas a las cuales podemos aplicar algunos métodos de validación de los datos ingresados, como ser las url de los sitios web o la estructura del correo electrónico, también podemos orientar algunas de las columnas a verificar la frescura o completitud de sus datos.
También por ejemplo existen dos columnas que contienen las coordenadas de la ubicación del operador, esta información podría validarse si los valores ingresados son correctos asumiendo que deben estar por lo menos en Uruguay y si fueran dentro de lo esperado poder utilizar algún servicio como Google Maps o la api de Geocoding de google que permite hacer peticiones de los datos de una determinada coordenada y de esa forma validar más certeramente la información.

Valores aproximados para el territorio Uruguayo:
Latitudes: entre -30.130153 y -35.061084
Longitud: entre -58.483502 y -53.082044

Analizando en primera instancia los valores de latitud y longitud de los datos, es posible ver que algunos de ellos están fuera de rango, ya que la latitud puede variar, teóricamente, entre -90 y 90, mientras que la longitud puede variar, teóricamente, entre -180 y 180.

Longitud	Latitud
-54	-34
-10000	-10000
-10000	-10000
-10000	-10000
-54	-13
-10000	-10000
-10000	-10000
-10000	-10000
-10000	-10000

Revisando un poco la unicidad de los registros tratando de determinar si existen registros duplicados en el set de datos vemos que si agrupamos todas las columnas del mismo tenemos 98 incidencias donde tenemos registros repetidos 2 o más veces.

Cant_Ocurrencias
98

En la imagen se puede visualizar un ejemplo de una serie de registros idénticos en todos sus valores.

TipoOperador	Operador	Departamento	Localidad	Direccion	Telefono	Web	EMail	Longitud	Latitud
Rentadora de autos	EUROPCAR	CANELONES	PASO CARRASCO	26 DE AGOSTO	610|094404570	S/D	silbus@vera.com.uy	-10000	-10000
Rentadora de autos	EUROPCAR	CANELONES	PASO CARRASCO	26 DE AGOSTO	610|094404570	S/D	silbus@vera.com.uy	-10000	-10000
Rentadora de autos	EUROPCAR	CANELONES	PASO CARRASCO	26 DE AGOSTO	610|094404570	S/D	silbus@vera.com.uy	-10000	-10000
Rentadora de autos	EUROPCAR	CANELONES	PASO CARRASCO	26 DE AGOSTO	610|094404570	S/D	silbus@vera.com.uy	-10000	-10000

En vista de que el archivo no nos brinda una clara clave que permita identificarla como única, es que estuvimos revisando posibles agrupaciones de datos.
La agrupación de todas las columnas nos devuelve que del total de 3288 registros en el archivo, hay 2770 que son únicos.

```
df = sc.sql("""
    select count(1) as Cant_Ocurrencias
    from (select TipoOperador,Operador,Departamento,Localidad,Direccion,Telefono,Web,EMail,Longitud,Latitud
    from df_operadores
    group by TipoOperador,Operador,Departamento,Localidad,Direccion,Telefono,Web,EMail,Longitud,Latitud
    having count(1) > 0)
""")
df.show(50)
```

```
+----------------+
|Cant_Ocurrencias|
+----------------+
|            2770|
+----------------+
```

Archivo receptivos.json

El conjunto de datos proporcionado cuenta con información recopilada por encuestas hechas a turistas que ingresan al país a través de los distintos puntos fronterizos. Tiene como finalidad registrar la información brindada por los turistas sobre el motivo de su viaje al país, cuál es el destino final dentro del país, alojamiento que utilizará en su estadía, medio en el que ingresa al país, así mismo el que utilizará para la salida, nivel de estudio del turista y gastos que realizó en su estadía, como ser en comida, cultura, alojamiento, etc.
Si bien el archivo puede considerarse bastante estructurado porque luego de su primer análisis vemos que es muy similar en su estructura al de “emisivos.csv”, los archivos .json suelen ser categorizados como datos desestructurados ya que su composición puede variar de un registro a otro, es decir lo que solemos ver como columnas, en este caso filas del archivo, puede no contenerse en todos los documentos (registros de una tabla relacional), entonces eso haría que su análisis sea más complejo y diferente en cuanto al modelo de calidad a aplicar.

Un análisis primario del Archivo nos devuelve algunas mediciones que apuntan a la calidad de algunas columnas.

Medición	**Valor**
Cantidad de Registros (filas)	47785
Cantidad de Columnas	48
Cantidad de Columnas Enteras	19
Cantidad de Columnas Strings	17
Cantidad de Columnas Decimales	10
Cantidad de Columnas Date	2

Revisando un poco el archivo podemos visualizar que existen muchos registros nulos o vacíos para toda la tupla.

Registros_nulos
15490

En las siguientes imágenes hicimos un análisis sobre cada columna para cuantificar además de los 15490 registros que ya sabemos que están nulos o vacíos a cuánto asciende si lo miramos una a una.

IdIngresos_nulos	Lugar_Ingreso_nulos	IdTranspIngreso_nulos	Transporte_Internacional_Ingreso_nulos	FechaIngreso_nulos
45286	45286	45286	45286	45286

IdFecIng_nulos	FechaEgreso_nulos	IdFecEgr_nulos	IdNacionalidad_nulos	Pais_nulos	IdResidencia_nulos	Residencia_nulos
45286	45286	45286	45286	45286	45286	45286

IdMotivo_nulos	Motivo_nulos	IdOcupacion_nulos	Ocupacion_nulos	IsEstudio_nulos	Estudio_nulos	IdDestinoLocalidad_nulos
45286	45286	45286	45286	18393	18393	15490

Localidad_nulos	IdDepartamentoDestino_nulos	Departamento_nulos	IdOtroDepartamento_nulos	Otro_Departamento_nulos
15490	15490	15490	15490	18393

IdOtraLocalidad_nulos	Otra_Localidad_nulos	IdAlojamiento_nulos	Alojamiento_nulos	IdTranspLocal_nulos	TransporteLocal_nulos
18393	18393	15490	15490	15490	15490

IdEgresos_nulos	Lugar_Egreso_nulos	IdTranspEgreso_nulos	Transporte_Internacional_Egreso_nulos	IdDestino_nulos
15490	15490	15490	15490	15490

Destino_nulos	Estadia_nulos	Gente_nulos	GastoTotal_nulos	GastoAlojamiento_nulos	GastoAlimentacion_nulos	GastoTransporte_nulos
15490	15490	15490	15490	15490	15490	15490

GastoCultural_nulos	GastoTours_nulos	GastoCompras_nulos	GastoOtros_nulos	Coef_nulos	CoefTot_nulos
15490	15490	15490	15490	15490	15490

Tenemos 16 columnas con más de un 94% de los valores en null, 5 columnas con 38,5% de sus valores en null y las restantes con un 32,4% de los valores en null.
También es posible ver que este archivo no presenta los problemas sintácticos en las columnas de fecha que presentaba el archivo de emisivos. Asimismo, se cuenta únicamente con datos de alojamiento, alimentación, transporte y cultural. Al no tener atributo de GastosOtros, como ocurría en el caso de emisivos, no es posible verificar la dependencia funcional entre los datos de gastos. Respecto a la dependencia funcional entre los atributos

de Id y el atributo al que hace referencia dicho Id, se verifica nuevamente que existen inconsistencias en algunos pares de atributos:

IdOtroDepartamento	Cant_Lugares	Cant_Nulos	Cant_Lugares_Distintos
	15490	0	1
0	1	0	1
1	2332	0	2
10	27	0	2
11	7	0	2
12	37	0	2
13	37	0	2
14	222	0	2
15	124	0	2
16	21	0	2
17	50	0	2
18	38	0	1
19	23	0	2
2	2866	0	2
3	1115	0	2
4	1023	0	2
5	738	0	2
6	575	0	2
7	301	0	2
8	98	0	2
80	21579	0	2
9	7	0	2
99	1074	0	2

IdOtraLocalidad	Cant_Lugares	Cant_Nulos	Cant_Lugares_Distintos
	18393	0	1
0	22582	0	5
1	3224	0	19
10	7	0	2
11	8	0	3
12	1	0	1
13	26	0	2
14	10	0	3
15	12	0	2
16	39	0	3
17	19	0	2
18	10	0	2
19	6	0	1
2	977	0	13
20	12	0	2
21	4	0	1
24	2	0	1
25	12	0	1
26	4	0	1
27	6	0	1
29	30	0	1
3	877	0	9
4	257	0	10
5	170	0	8
6	236	0	5
7	308	0	4
8	118	0	4
9	298	0	4
99	137	0	1

Especificación de modelo de calidad

Para la especificación del Modelo de Calidad elaboramos este documento con todo el detalle y justificaciones de cómo se van a ir midiendo los indicadores de calidad de los datos para cada uno de los archivos, también se elaboró la tabla con la especificación Organizada

epor Dimensiones (Tabla) donde se simplifica un poco todo el detalle de cada medición y su instancia con los métodos a aplicar.

Métricas Genéricas

Emisivos y operadores (modelo relacional)

1. EXACTITUD

Exactitud sintáctica

Se utiliza la siguiente métrica de manera genérica para evaluar la exactitud sintáctica.

Métrica	
Exact_sintactica_bool	
Descripción	Evalúa la correctitud sintáctica de una celda
Unidades	{0,1}
Granularidad	Celda

Métrica	
Exact_sintactica_dist	
Descripción	Evalúa la correctitud sintáctica de una celda en base a la mínima distancia de Levenshtein respecto a un referencial
Unidades	[0..1]
Granularidad	Celda

Exactitud semántica

Se plantea la siguiente medida de calidad genérica.

Métrica	
Exact_semantica_bool	
Descripción	Evalúa la correctitud semántica de una celda

Unidades	{0,1}
Granularidad	Celda

Precisión
Se utiliza la siguiente métrica de manera genérica para evaluar la precisión de una celda.

Métrica	
Exact_precision	
Descripción	Evalúa la precisión de una celda
Unidades	{0, 0.33, 0.66, 1}
Granularidad	Celda

2. COMPLETITUD

Cobertura

Métrica	
Completitud_cobertura	
Descripción	Mide si mi tabla cubre todos los registros que debería cubrir.
Unidades	[0..1]
Granularidad	Tabla

Densidad
Dado que existen valores nulos, así como columnas con valores registrados como “Sin Datos”, se define la siguiente métrica:

Métrica	
Completitud_densidad	
Descripción	Calcula el ratio de valores faltantes en una columna con respecto a la cantidad total de valores
Unidades	[0..1]

Granularidad	Columna

3. FRESCURA

Actualidad

Métrica	
ratio_actualidad	
Descripción	Evalúa la actualidad de una tabla
Unidades	[0..1]
Granularidad	Tabla

Oportunidad

Entendemos que los datos al estar en un portal abierto y público a internet siempre van a estar disponibles, salvo que se cayeran los servidores de https://catalogodatos.gub.uy/ lo cual conllevaría que no se puedan descargar. Dado este punto no consideramos ninguna métrica representativa para cada uno de los archivos utilizados.

4. CONSISTENCIA

Consistencia, integridad intra - relación.

Varias de las columnas en los distintos archivos corresponden a números de “ID” de valores de otra columna. A modo de ejemplo, existen columnas de “IdLugarSalida” y “Lugar Salida”, “FechaSalida” y “IdFecSalida”, etc. Las columnas de ID parecen ser una clave no declarada con información sobre la columna correspondiente, por lo que debería cumplirse una dependencia funcional entre ambas columnas que asegure que un único valor de Id esté asociado a un único valor de la columna correspondiente. Resulta de interés, entonces, generar un diccionario con los valores de lugares correspondientes a cada id y evaluar la consistencia entre ambas columnas para cada uno de estos pares de atributos.

Métrica	
Consistencia_Intra_Relacion	
Descripción	Evalúa la consistencia entre dos datos distintos

Unidades	{0,1}
Granularidad	Celda

Métrica	
Consistencia_Intra_Relacion_columna	
Descripción	Evalúa la consistencia entre dos columnas distintas
Unidades	{0,1}
Granularidad	Columna

Métrica	
Consistencia_Intra_Relacion_conj_celdas	
Descripción	Evalúa la consistencia entre un conjunto de celdas
Unidades	{0,1}
Granularidad	Conjunto de celdas

Consistencia, integridad de dominio

Como fue mencionado en la sección anterior, resulta de interés evaluar la integridad de dominio de los campos de latitud y longitud. Para ello se define la siguiente métrica:

Métrica	
Consistencia_Int_Dominio	
Descripción	Evalúa la integridad de dominio de un dato
Unidades	[0..1]
Granularidad	Celda

5. UNICIDAD

No duplicación

Métrica	
Unicidad_no_duplicacion_tupla	
Descripción	Evalúa si un registro es un duplicado exacto o no
Unidades	{0,1}
Granularidad	tupla

No contradicción

Métrica	
Unicidad_no_contradiccion_tupla	
Descripción	Evalua si un registro es un duplicado contradictorio o no
Unidades	{0,1}
Granularidad	tupla

Receptivos (datos no estructurados)

El archivo en cuestión consiste en un listado de documentos, cada uno de los cuales corresponde a una persona a la cual se le realizó la encuesta de turismo receptivo. Cada uno de esos elementos consiste en pares clave-valor, donde cada clave corresponde al nombre de un atributo y cada valor corresponde a la información de dicho atributo para el encuestado en cuestión.
Dado que la estructura del archivo que contiene a los datos en cuestión es distinta a la de los archivos anteriores, se definen los siguientes términos para poder definir las métricas a utilizar.

- **Archivo**: Archivo JSON que contiene los datos de los cuales se desea evaluar la calidad (receptivos.json).
- **Documento**: Cada juego de pares clave-valor dentro del archivo. Cada documento se corresponde con una entidad de la realidad (en este caso, un encuestado).
- **Atributo**: Cada clave de los pares clave-valor que integran un documento. Cada clave es una palabra o conjunto de palabras que indica el atributo medido para la entidad representada por el documento en el cual la clave está contenida.. Los atributos no necesariamente se repiten para todos los documentos.
- **Valor**: Cada valor de los pares clave-valor que integran un documento. Cada valor contiene la información correspondiente al atributo (clave) al que está ligado para la entidad representada por el documento en el cual está contenido.

A modo de ejemplo, se presenta una captura de una parte del archivo en cuestión en el cual se indica cada uno de los elementos definidos:

De esta manera, se consideran las siguientes granularidades para las métricas a utilizar:

- **Archivo**: Una métrica con esta granularidad toma un único valor para todo el archivo JSON.
- **Documento**: Una métrica con esta granularidad toma un único valor para cada documento.
- **Atributo**: Una métrica con esta granularidad toma un único valor para un atributo. Dicho valor comprende a todos los documentos para los cuales esté definido, o no, dicho atributo. Para instanciar una métrica con esta granularidad, se debe indicar el atributo para el cual se instancia.
- **Valor**: Una métrica con esta granularidad toma un único valor para un valor asociado a algún atributo dentro de un documento. Para instanciar una métrica con esta granularidad, se debe indicar el atributo cuyo valor será utilizado para realizar la medición.
- **Conjunto de valores**: Una métrica con esta granularidad toma un único valor para un conjunto de valores dentro de un documento. Para instanciar una métrica con esta granularidad, se debe indicar los atributos cuyos valores serán utilizados para realizar la medición.

1. EXACTITUD

Exactitud sintáctica

Se utiliza la siguiente métrica de manera genérica para evaluar la exactitud sintáctica.

Métrica	
Exact_sintactica_bool_json	
Descripción	Evalúa la correctitud sintáctica de un valor
Unidades	{0,1}
Granularidad	Valor

Exactitud semántica

Se plantea la siguiente medida de calidad genérica.

Métrica	
Exact_semantica_bool_json	
Descripción	Evalúa la correctitud semántica de un valor
Unidades	{0,1}
Granularidad	Valor

Precisión

Se utiliza la siguiente métrica genérica

Métrica	
Exact_precision_json	
Descripción	Evalúa la precisión de un valor
Unidades	{0, 0.25, 0.50, 0.75, 1}
Granularidad	Valor

2. COMPLETITUD

Cobertura

Métrica	
Completitud_cobertura_json	
Descripción	Mide si un archivo cubre todas las entidades que debería cubrir.
Unidades	[0..1]
Granularidad	Archivo

Densidad

Dado que existen valores nulos, así como atributos con valores registrados como “Sin Datos”, se define la siguiente métrica:

Métrica	
Completitud_densidad_json	
Descripción	Calcula el ratio de valores faltantes de algún atributo con respecto a la cantidad total de documentos. Se consideran todos los documentos, sin importar si el atributo en cuestión está o no definido para todos los documentos.
Unidades	[0..1]
Granularidad	Atributo

3. FRESCURA

Actualidad

Métrica	
ratio_actualidad_json	
Descripción	Evalúa la actualidad de un archivo
Unidades	[0..1]
Granularidad	Archivo

Oportunidad

Entendemos que los datos al estar en un portal abierto y público a internet siempre van a estar disponibles, salvo que se cayeran los servidores de https://catalogodatos.gub.uy/ lo cual conllevaría que no se puedan descargar. Dado este punto no se considera ninguna métrica representativa para el documento en cuestión.

4. CONSISTENCIA

Métrica	
Consistencia_Intra_Relacion_json	
Descripción	Evalúa la consistencia entre dos valores distintos
Unidades	{0,1}
Granularidad	Valor

Métrica	
Consistencia_Intra_Relacion_conj_valores_json	
Descripción	Evalúa la consistencia entre un conjunto de valores
Unidades	{0,1}
Granularidad	Conjunto de valores

5. UNICIDAD

No duplicación

Métrica	
Unicidad_no_duplicacion_documento_json	
Descripción	Evalúa si un registro es un duplicado exacto o no
Unidades	{0,1}
Granularidad	Documento

No contradicción

Métrica	
Unicidad_no_contradiccion_documento_json	
Descripción	Evalúa si un registro es un duplicado contradictorio o no
Unidades	{0,1}
Granularidad	Documento

Métricas instanciadas

Emisivos

1. EXACTITUD

Exactitud sintáctica - Métricas Instanciadas

Como se mencionó en la sección anterior, algunos valores de IdFechaSalida están asociados con más de una única fecha debido a incoherencias en el formato de ingreso de dicha fecha. En virtud de esto, se define la siguiente métrica instanciada.

Métrica instanciada	
Exact_sintactica_bool_FechaSalida	
Métrica	Exact_sintactica_bool
Datos	emisivos.'FechaSalida'
Método Medición	if check_date_format(FechaSalida) == True exactitud = 1 else: exactitud = 0

Nota: la función referida en el método de medición es la siguiente:

```
def check_date_format(string):
    pattern = r"\d{4}-\d{2}-\d{2}"
    match = re.fullmatch(pattern, string)
    if match:
        return True
    else:
```

```
        return False
```

Métrica instanciada	
Exact_sintactica_bool_Departamento	
Métrica	Exact_sintactica_bool
Datos	emisivos.Departamento
Método Medición	if Departamento not in list_departamentos exactitud = 0 else: exactitud = 1

Nota: list_departamentos corresponde a una lista exhaustiva de los departamentos del Uruguay. Dicha lista es indicada a continuación.
list_departamentos = ['Montevideo', 'Maldonado', 'Canelones', 'Salto', 'Lavalleja', 'San Jose', 'Rio Negro', nan, 'Durazno', 'Cerro Largo', 'Colonia', 'Tacuarembo', 'Treinta y Tres', 'Rocha', 'Artigas', 'Paysandu', 'Rivera', 'Flores', 'Soriano', 'Florida']

Exactitud semántica - Métricas Instanciadas

Asimismo, se plantean las siguientes métricas instanciadas

Métrica instanciada	
Exact_semántica_bool_`Lugar Salida`	
Métrica	Exact_semantica_bool
Datos	emisivos.`Lugar Salida`
Método Medición	if `Lugar Salida` not in list_lugares_salida exactitud = 0 else: exactitud = 1

Nota: list_lugares_salida corresponde a una lista de puntos de ingreso y egreso al país que se toma como referencial. Dicha lista es indicada a continuación.
list_lugares_salida = ['Chuy', 'Río Branco', 'Aeropuerto de Carrasco', 'Fray Bentos', 'Bella Unión', 'Colonia', 'Salto', 'Paysandú', 'Carmelo', 'Melo - Aceguá', 'Artigas', 'Rivera', 'Puerto de montevideo', 'Nueva Palmira']

Precisión - Métricas Instanciadas

La métrica se aplica a la columna de Destino y País de la siguiente manera:

Métrica instanciada	
Exact_precision_Destino	
Métrica	Exact_precision
Datos	emisivos.Destino

Método Medición	if Destino is pais:
	precision = 1.00
	elif Destino is region:
	precision = 0.66
	elif Destino is continente:
	precision = 0.33
	else:
	precision = 0

Métrica instanciada	
Exact_precision_Pais	
Métrica	Exact_precision
Datos	emisivos.Pais

Método Medición	if Pais is pais:
	exactitud = 1.00
	elif Pais is Gran Bretaña:
	exactitud = 0.66
	elif "Otro" in Pais:
	exactitud = 0.33
	else:
	exactitud = 0

Nota: la adjudicación de un valor de precisión menor para “Gran Bretaña” es realizada en base al entendido de que “Gran Bretaña” hace referencia a cualquiera de los países que forman la región (Inglaterra, Escocia y Gales).

2. COMPLETITUD

Cobertura - Métricas Instanciadas

Métrica instanciada	
Completitud_cobertura_emisivos	
Métrica	Completitud_cobertura
Datos	emisivos
Método Medición	cobertura = sum(emisivos[‘Gente’]) / L_ref_estim

Nota: sum(emisivos[‘gente’]) es la cantidad de turistas registrados por la encuesta, idealmente luego de identificar y filtrar los duplicados. Asimismo, L_ref_estim corresponde a una estimación de la cantidad real de turistas que viajaron en el período Dic/16 - Mayo/23. Esta cantidad es estimada en base al informe de turismo emisivo del Ministerio de Turismo del Uruguay, obtenido de la siguiente web: https://www.gub.uy/ministerio-turismo/datos-y-estadisticas/estadisticas/turismo-emisivo-2023. La cantidad de turistas por año se resumen en la siguiente tabla:

Dic 2016	47.624	No hay datos, se estima como dic-21
2017	1.788.792	
2018	1.788.792	No hay datos, se estima como 2017
2019	2.199.152	
2020	599.512	1er trimestre (hasta comienzo de pandemia)
2021	92.517	Noviembre y diciembre (“fin de pandemia”)
2022	2.383.901	
2023	1.141.301	1er trimestre
Tot	**10.041.591**	**Valor asignado para L_ref_estim**

Densidad - Métricas Instanciadas

Se instancia para todas las columnas con datos nulos o con celdas “Sin Datos”.

Métrica instanciada	
Completitud_densidad_Lugar Salida_emisivos	
Métrica	Completitud_densidad
Datos	emisivos.Lugar Salida
Método Medición	densidad = suma_nulos(df['Lugar Salida'])

Métrica instanciada	
Completitud_densidad_IdDeptoResidencia_emisivos	
Métrica	Completitud_densidad
Datos	emisivos.IdDeptoResidencia
Método Medición	densidad = suma_nulos(df['IdDeptoResidencia'])

Métrica instanciada	
Completitud_densidad_Departamento_emisivos	
Métrica	Completitud_densidad
Datos	emisivos.Departamento
Método Medición	densidad = suma_nulos(df['Departamento'])

Métrica instanciada	
Completitud_densidad_IdMotivo_emisivos	
Métrica	Completitud_densidad
Datos	emisivos.IdMotivo
Método Medición	densidad = suma_nulos(df['IdMotivo'])

Métrica instanciada

Completitud_densidad_Motivo_emisivos	
Métrica	Completitud_densidad
Datos	emisivos.Motivo
Método Medición	densidad = suma_nulos(df['Motivo'])

Métrica instanciada	
Completitud_densidad_Ocupacion_emisivos	
Métrica	Completitud_densidad
Datos	emisivos.Ocupacion
Método Medición	densidad = suma_nulos(df['Ocupacion'])

Métrica instanciada	
Completitud_densidad_Estudio_emisivos	
Métrica	Completitud_densidad
Datos	emisivos.Estudio
Método Medición	densidad = suma_nulos(df['Estudio'])

Métrica instanciada	
Completitud_densidad_Destino_emisivos	
Métrica	Completitud_densidad
Datos	emisivos.Destino
Método Medición	densidad = suma_nulos(df['Destino'])

Métrica instanciada	
Completitud_densidad_Alojamiento_emisivos	
Métrica	Completitud_densidad
Datos	emisivos.Alojamiento

Método Medición	densidad = suma_nulos(df['Alojamiento'])

Métrica instanciada	
Completitud_densidad_Trasporte Local_emisivos	
Métrica	Completitud_densidad
Datos	emisivos.Trasporte Local
Método Medición	densidad = suma_nulos(df['Trasporte Local'])

Nota: La métrica es definida en base a la función suma_nulos, la cual es presentada a continuación.

```
def suma_nulos(column):
    null_count = column.isnull().sum()
    sin_datos_count = (column == "Sin Datos").sum()
    desc_sin_datos_count = (column == "Desconocido / Sin Datos").sum()
    s_d_count = (column == "S/D").sum()

    cant_nulos = null_count + sin_datos_count + desc_sin_datos_count + s_d_count
    return round(cant_nulos/len(column),4)
```

3. FRESCURA

Actualidad - Métricas Instanciadas

Métrica instanciada	
ratio_actualidad_emisivos	
Métrica	ratio_actualidad
Datos	emisivos
Método Medición	frescura = 1 - (t2-t0) / Δt0

Nota:

- t0: fecha de actualización de tabla utilizada
- t2: fecha de realización de consulta
- Δt0: período de tiempo entre actualizaciones de tabla (inverso de frecuencia

de actualización).

Dado que se asume que hay turistas entrando y saliendo del país todos los días, la información de la tabla queda desactualizada respecto del mundo real en el momento en el cual se publica. Utilizando “días” como unidad de medida, dado que la actualización de los datos es mensual según el Catálogo Nacional de Datos Abiertos, se tiene que Δt0 = 30 días. Luego, si se realiza la consulta en el momento de actualización de los datos (t2 = t0), la métrica toma el valor de 1. A medida que la consulta se realiza días después de la última actualización, t2 aumenta y por tanto t2-t0 aumenta, lo que reduce la métrica de actualidad. De esta manera, si se realiza la consulta el último día antes de la actualización de la tabla (t2-t0 = Δt0), la métrica toma el valor de 0. Dado que la actualización es periódica, la métrica debe tomar el valor nulo si se realiza la consulta justo antes de la siguiente actualización de la tabla, ya que realizar la consulta un día después implicaría tener los datos lo más actualizados posible.

4. CONSISTENCIA

Consistencia, integridad intra - relación - Métricas Instanciadas

Varias de las columnas corresponden a números de “ID” de valores de otra columna. A modo de ejemplo, existen columnas de “IdLugarSalida” y “Lugar Salida”, “FechaSalida” y “IdFecSalida”, etc. Las columnas de ID parecen ser una clave no declarada con información sobre la columna correspondiente, por lo que debería cumplirse una dependencia funcional entre ambas columnas que asegure que un único valor de Id esté asociado a un único valor de la columna correspondiente. Resulta de interés, entonces, generar un diccionario con los valores de lugares correspondientes a cada id y evaluar la consistencia entre ambas columnas para cada uno de estos pares de atributos.

Métrica instanciada	
Consistencia_Intra_Relacion_Lugar.Salida_emisivos	
Métrica	Consistencia_Intra_Relacion
Datos	emisivos.'Lugar Salida'

Método Medición	if 'Lugar Salida' != dict_lugares_salida[IdLugarSalida]:
	consistencia = 0
	else:
	consistencia = 1

Nota: dict_lugares_salida corresponde a un diccionario en formato python con pares clave-valor cuyas claves corresponden a los valores de IdLugarSalida y los valores corresponden al Lugar de salida al cual hacen referencia.

Asimismo, en la sección anterior se planteó la dependencia funcional entre las columnas de “Lugar Ingreso” y “Lugar Salida”. Dicha dependencia funcional es evaluada en la siguiente métrica.

Métrica instanciada	
Consistencia-Intra-Relacion_columna-Lugar.Salida_Emisivos	
Métrica	Consistencia_Intra_Relacion_columna
Datos	emisivos.'Lugar Salida'
Método Medición	if set(df['Lugar Salida'].unique()) != set(df['Lugar Ingreso'].unique()) consistencia = 0 else: consistencia = 1

Nota: se evalúa la consistencia únicamente en la columna “Lugar Salida” porque se entiende que los datos incluidos en la columna de “Lugar Ingreso” son correctos. De esta manera, si hay inconsistencias entre ambas, corresponden a errores en la columna de “Lugar Salida”.

Métrica instanciada	
Consistencia_Intra_Relacion_Fechas_Estadia_Emisivos	
Métrica	Consistencia_Intra_Relacion_conj_celdas
Datos	emisivos.FechaEntrada, emisivos.FechaSalida, emisivos.Estadia
Método Medición	if FechaEntrada - FechaSalida != Estadia consistencia = 0 else: consistencia = 1

Nota: Está métrica debe ser aplicada una vez que se hayan corregido y estandarizado las columnas de FechaEntrada y FechaSalida.

Asimismo, se instancia la misma métrica para evaluar la consistencia entre las columnas de gastos del archivos.

Métrica instanciada	
Consistencia_Intra_Relacion_Gastos_Emisivos	
Métrica	Consistencia_Intra_Relacion_conj_celdas
Datos	emisivos.GastoAlojamiento, emisivos.GastoAlimentacion, emisivos.GastoCompras, emisivos.GastoCultural, emisivos.GastoResto, emisivos.GastoTours, emisivos.GastoTransporteInternac, emisivos.GatoTransporteLocal, emisivos.GastoTotal
Método Medición	if round(GastoAlojamiento + GastoAlimentacion + GastoCompras + GastoCultural + GastoResto + GastoTours + GastoTransporteInternac + GatoTransporteLocal,0) != round(GastoTotal,0) consistencia = 0 else: consistencia = 1

La misma métrica es también instanciada para evaluar la consistencia entre las columnas de “Lugar Salida” y “Transporte Internacional de Salida”, y entre las columnas de “Lugar Ingreso” y “Transporte Internacional de Ingreso”

Métrica instanciada	
Consistencia_Intra_Relacion_lugar_transporte_emisivos_salida	
Métrica	Consistencia_Intra_Relacion_conj_celdas
Datos	emisivos.`Lugar Salida`, emisivos.`Transporte Internacional de Salida`
Método Medición	if `Lugar Salida` == 'Aeropuerto de Carrasco' and `Transporte Internacional de Salida` != 'Aereo' consistencia = 0 elif `Lugar Salida` == 'Puerto de montevideo' and `Transporte Internacional de Salida` != 'Maritimo - Fluvial' consistencia = 0 else: consistencia = 1

Métrica instanciada
Consistencia_Intra_Relacion_lugar_transporte_emisivos_ingreso

Métrica	Consistencia_Intra_Relacion_conj_celdas
Datos	emisivos.`Lugar Ingreso`, emisivos.`Transporte Internacional de Ingreso`
Método Medición	if `Lugar Ingreso` == 'Aeropuerto de Carrasco' and `Transporte Internacional de Ingreso` != 'Aereo' consistencia = 0 elif `Lugar Ingreso` == 'Puerto de montevideo' and `Transporte Internacional de Ingreso` != 'Maritimo - Fluvial' consistencia = 0 else: consistencia = 1

Consistencia, integridad de dominio

Métrica instanciada	
Consistencia_Int_Dominio_Latitud_emisivos	
Métrica	Consistencia_Int_Dominio
Datos	emisivos.latitud
Método Medición	consistencia = 1 - dist_lat_norm (latitud, departamento)

Métrica instanciada	
Consistencia_Int_Dominio_Longitud_emisivos	
Métrica	Consistencia_Int_Dominio
Datos	emisivos.longitud
Método Medición	consistencia = 1 - dist_lon_norm (latitud, departamento)

Nota: Se definen funciones que calculan la distancia entre la latitud o longitud ingresada y el valor más cercano del rango de valores posibles para el departamento correspondiente. Dicha distancia es normalizada dividiéndola entre la mayor distancia posible. Estas funciones son presentadas a continuación.

```python
def dist_lat_norm (latitud, departamento):
    max_lat = dict_coord[departamento]['max_lat']
    min_lat = dict_coord[departamento]['min_lat']
    if latitud > max_lat:
        distancia_norm = abs(latitud - max_lat) / abs(max_lat - (90))
    elif latitud < min_lat:
        distancia_norm = abs(latitud - min_lat) / abs(min_lat - (-90))
    else:
        distancia_norm = 0

    return round(distancia_norm,4)
```

```python
def dist_lon_norm (longitud, departamento):
    max_lon = dict_coord[departamento]['max_lon']
    min_lon = dict_coord[departamento]['min_lon']
    if longitud > max_lon:
        distancia_norm = abs(longitud - max_lon) / abs(max_lon - (180))
    elif longitud < min_lon:
        distancia_norm = abs(longitud - min_lon) / abs(min_lon - (-180))
    else:
        distancia_norm = 0

    return round(distancia_norm,4)
```

A su vez, dict_coord corresponde a un diccionario en formato python con pares clave-valor cuyas claves corresponden a los departamentos y los valores corresponden a otro diccionario con claves “max_lat”, “min_lat”, “max_lon”, “min_lon”, y valores correspondientes a las latitudes y longitudes máximas y mínimas respectivamente para el departamento correspondiente. El mismo es presentado a continuación.

```python
dict_coord = {
    'Montevideo': {'max_lat': -34.7833, 'min_lat': -34.9300, 'max_lon': -56.0126, 'min_lon': -56.4000},
    'Maldonado': {'max_lat': -34.5666, 'min_lat': -34.9700, 'max_lon': -54.9127, 'min_lon': -55.4084},
    'Canelones': {'max_lat': -34.5340, 'min_lat': -34.9572, 'max_lon': -55.4167, 'min_lon': -56.4526},
    'Salto': {'max_lat': -31.2005, 'min_lat': -31.4503, 'max_lon': -57.6760, 'min_lon': -57.9723},
    'Lavalleja': {'max_lat': -33.1833, 'min_lat': -34.6729, 'max_lon': -54.7500, 'min_lon': -55.3333},
    'San José': {'max_lat': -33.0000, 'min_lat': -34.6311, 'max_lon': -56.1385, 'min_lon': -56.7500},
    'Río Negro': {'max_lat': -32.5000, 'min_lat': -33.8902, 'max_lon': -56.1667, 'min_lon': -58.0667},
    'Durazno': {'max_lat': -32.0675, 'min_lat': -33.6650, 'max_lon': -55.5169, 'min_lon': -57.4614},
    'Cerro Largo': {'max_lat': -31.9333, 'min_lat': -33.7691, 'max_lon': -53.9167, 'min_lon': -55.4750},
    'Colonia': {'max_lat': -33.1334, 'min_lat': -34.6827, 'max_lon': -56.1864, 'min_lon': -58.5143},
    'Tacuarembó': {'max_lat': -31.0833, 'min_lat': -32.6508, 'max_lon': -54.6000, 'min_lon': -56.6000},
    'Treinta y Tres': {'max_lat': -32.5333, 'min_lat': -34.4231, 'max_lon': -53.7500, 'min_lon': -55.3333},
    'Rocha': {'max_lat': -33.1000, 'min_lat': -34.7524, 'max_lon': -53.5833, 'min_lon': -54.7500},
    'Artigas': {'max_lat': -29.5833, 'min_lat': -30.7685, 'max_lon': -56.0000, 'min_lon': -57.8000},
    'Paysandú': {'max_lat': -31.1167, 'min_lat': -32.5227, 'max_lon': -57.0000, 'min_lon': -58.1000},
    'Rivera': {'max_lat': -30.8333, 'min_lat': -31.9759, 'max_lon': -55.4000, 'min_lon': -56.5000},
    'Flores': {'max_lat': -32.5167, 'min_lat': -33.6493, 'max_lon': -55.0833, 'min_lon': -56.3333},
    'Soriano': {'max_lat': -32.6333, 'min_lat': -33.8216, 'max_lon': -57.4167, 'min_lon': -58.5833},
    'Florida': {'max_lat': -33.1000, 'min_lat': -34.1117, 'max_lon': -55.5833, 'min_lon': -57.3333},
}
```

Se define de esta manera porque se entiende que un valor de latitud o longitud muy cercano al rango de valores posibles implica un dato de mayor calidad que un valor muy alejado.

5. UNICIDAD

No duplicación - Métricas Instanciadas

Métrica instanciada	
Unicidad_no_duplicacion_tupla_emisivos	
Métrica	Unicidad_no_duplicacion_tupla
Datos	emisivos.IdLugarSalida, emisivos.IdTranspSalidad, emisivos.Transporte Internacional de Salida, emisivos.IdFecSalida, emisivos.FechaEntrada, emisivos.IdFecEntrada, emisivos.IdNacionalidad emisivos.Pais, emisivos.IdDeptoResidencia, emisivos.Departamento, emisivos.IdMotivo emisivos.Motivo, emisivos.IdOcupacion, emisivos.Ocupacion emisivos.IdNivelEstudio, emisivos.Estudio, emisivos.IdDestino, emisivos.Destino, emisivos.IdAlojamiento, emisivos.Alojamiento emisivos.IdLugarIngreso, emisivos.Lugar Ingreso emisivos.IdTranspIngreso, emisivos.Transporte Internacional de Ingreso, emisivos.IdTranspLocal, emisivos.Trasporte Local emisivos.Estadia, emisivos.Gente, emisivos.GastoTotal, emisivos.GastoAlojamiento, emisivos.GastoAlimentacion emisivos.GastoTransporteInternac, emisivos.GatoTransporteLocal, emisivos.GastoCultural, emisivos.GastoTours, emisivos.GastoCompras, emisivos.GastoResto
Método Medición	unicidad = duplicados_row(indice)['no_duplicacion']

Métrica instanciada
Unicidad_no_contradiccion_tupla_emisivos

Métrica	Unicidad_no_contradiccion_tupla
Datos	emisivos.IdLugarSalida, emisivos.IdTranspSalidad, emisivos.Transporte Internacional de Salida, emisivos.IdFecSalida, emisivos.FechaEntrada, emisivos.IdFecEntrada, emisivos.IdNacionalidad emisivos.Pais, emisivos.IdDeptoResidencia, emisivos.Departamento, emisivos.IdMotivo emisivos.Motivo, emisivos.IdOcupacion, emisivos.Ocupacion emisivos.IdNivelEstudio, emisivos.Estudio, emisivos.IdDestino, emisivos.Destino, emisivos.IdAlojamiento, emisivos.Alojamiento emisivos.IdLugarIngreso, emisivos.Lugar Ingreso emisivos.IdTranspIngreso, emisivos.Transporte Internacional de Ingreso, emisivos.IdTranspLocal, emisivos.Trasporte Local emisivos.Estadia, emisivos.Gente, emisivos.GastoTotal, emisivos.GastoAlojamiento, emisivos.GastoAlimentacion emisivos.GastoTransporteInternac, emisivos.GatoTransporteLocal, emisivos.GastoCultural, emisivos.GastoTours, emisivos.GastoCompras, emisivos.GastoResto
Método Medición	unicidad = duplicados_row(indice)['no_contradiccion']

Nota: Se define la siguiente función que toma como argumento de entrada un dataframe de pandas generado a partir de la tabla en la cual se quiere buscar los duplicados y el índice de la fila que se quiere evaluar y retorna un diccionario de python con el valor de la métrica de “no duplicación” y de “no contradicción” de dicha fila. Esta función es presentada a continuación.

```python
def duplicados_row (df,row):
    subset_columns = list(df.columns)
    cols_not_to_consider = ['Coef', 'CoefTot', 'latitud', 'longitud', 'FechaSalida', 'Lugar Salida']

    # Genero listado de columnas a partir de las cuales se buscan duplicados
    for col in cols_not_to_consider:
        subset_columns.pop(subset_columns.index(col))

    # Genero dataframe con duplicados
    df_duplicates = df[df.duplicated(subset=subset_columns,keep=False)].sort_values(by = subset_columns)

    df_dup_row = df_duplicates[(df_duplicates[subset_columns] == df.loc[row][subset_columns]).all(axis=1)] # Dataframe con duplicados para cada registro duplicado

    if len(df_dup_row) == 0:
        no_duplicacion = 1
        no_contradiccion = 1
        return {'no_duplicacion': no_duplicacion, 'no_contradiccion': no_contradiccion}

    # Genero listas con los atributos que definiran si el duplicado es contradictorio o no
    for col in cols_not_to_consider:
        coef_list = [coef for coef in df_dup_row['Coef']]
        coef_tot_list = [coef_tot for coef_tot in df_dup_row['CoefTot']]
        lat_list = [lat for lat in df_dup_row['latitud']]
        lon_list = [lon for lon in df_dup_row['longitud']]
    check_list = [coef_list, coef_tot_list, lat_list, lon_list]

    to_remove = [None, 'Sin Datos', 'Desconocido / Sin Datos'] # valores a remover para chequear si los restantes son contradictorios

    # Chequeo si los duplicados son contradictorios
    for lista in check_list:
        contradiccion = []
        for value in to_remove:
            lista.remove(value) if value in lista else lista
        if len(lista) == len(set(lista)):
            contradiccion.append(True) # Existen valores distintos luego de sacar los nulos o faltantes
        else:
            contradiccion.append(False) # No existen valores distintos luego de sacar los nulos o faltantes

    if True in contradiccion:
        no_duplicacion = 1
        no_contradiccion = 0
    else:
        no_duplicacion = 0
        no_contradiccion = 1

    return {'no_duplicacion': no_duplicacion, 'no_contradiccion': no_contradiccion}
```

Operadores

1. EXACTITUD

Exactitud sintáctica - Métricas Instanciadas

Para el caso de la fuente de datos de operadores podemos aplicarle exactitud sintáctica a algunas de sus celdas para poder establecer la correctitud de la información.

Métrica instanciada	
Exact_sintactica_bool_telefono_ope	
Métrica	Exact_sintactica_bool
Datos	operadores.Telefono'
Método Medición	if val_tel(telefono) == True exactitud = 1 else: exactitud = 0

La función “val_tel” recibe como parámetro una cadena de texto y determina si es correcta la sintaxis, pudiendo evaluarse un número de teléfono fijo o un celular.

```python
def val_tel (telefono):
    if telefono[0] == '0':
        return True if len(telefono) == 9 else False
    elif telefono[0] == '2' or telefono[0] == '4':
        return True if len(telefono) == 8 else False
    else:
        return False
```

Métrica instanciada	
Exact_sintactica_bool_Sitio_web_ope	
Métrica	Exact_sintactica_bool
Datos	operadores.'Web'
Método Medición	if es_sitio_web(Web) == True exactitud = 1 else: exactitud = 0

La función “es_sitio_web” recibe como parámetro una cadena de texto y determina si es correcta la sintaxis, pudiendo evaluarse un sitio web, un sitio localhost, un sitio donde el dominio sea una ip, o incluso un repositorio ftp.

```python
import re

def es_sitio_web(cadena):
    # Patrón para verificar si la cadena coincide con una URL
    patron_url = re.compile(r'^(?:http|ftp|sftp)s?://|^(?:www)s?.'  # http:// o https:// o www.
                            r'(?:(?:[A-Z0-9](?:[A-Z0-9-]{0,61}[A-Z0-9])?\.)+(?:[A-Z]{2,6}\.?|[A-Z0-9-]{2,}\.?)|'  # Dominio
                            r'localhost|'  # "localhost"
                            r'\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})'  # Dirección IP
                            r'(?::\d+)?'  # Puerto opcional
                            r'(?:/?|[/?]\S+)$', re.IGNORECASE)

    if re.match(patron_url, cadena):
        return True
    else:
        return False
```

Métrica instanciada	
Exact_sintactica_bool_Email_ope	
Métrica	Exact_sintactica_bool

Datos	operadores.'EMail'
Método Medición	if es_correo_electronico(Email) == True exactitud = 1 else: exactitud = 0

La función “es_correo_electronico” recibe como parámetro una cadena de texto y determina si es correcta la sintaxis para un posible correo electrónico

```python
import re

def es_correo_electronico(cadena):
    # Patrón para verificar si la cadena coincide con una dirección de correo electrónico
    patron_correo = re.compile(r'^[\w\.-]+@[\w\.-]+\.\w+$')

    if re.match(patron_correo, cadena):
        return True
    else:
        return False
```

Se evalúa la exactitud sintáctica de la columna de “Direccion” utilizando la distancia de Levenshtein normalizada. Para ello, se utiliza un referencial de Catálogo Abierto de Datos obtenido de la siguiente web: https://catalogodatos.gub.uy/dataset/ide-calles-por-localidad-y-departamento, el cual cuenta con calles por localidad y departamento. Se implementa la siguiente función.

```python
import Levenshtein

def min_levenshtein_distance(string, referencial):
    # Se sacan los números de la columna de Dirección
    modified_string = ''.join(char for char in string if not char.isdigit())

    # Se calcula la mínima distancia de Levenshtein a algún valor del referencial
    # y se guarda el valor
    min_distance = float('inf')
    min_value = None

    for value in referencial:
        distance = Levenshtein.distance(modified_string, str(value))
        if distance < min_distance:
            min_distance = distance
            min_value = value

    return round(min_distance/(len(modified_string) + len(min_value)),4)
```

Métrica instanciada	
Exact_sintactica_dist_direccion	
Métrica	Exact_sintactica_dist

Datos	operadores.Direccion
Método Medición	exactitud = 1 - min_levenshtein_distance (Direccion, referencial)

Nota: La distancia es normalizada al dividirla entre la distancia máxima posible, la cual es igual a la longitud de la calle a chequear y la longitud de la calle más cercana del referencial.

Exactitud semántica - Métricas Instanciadas

Evaluar si el teléfono ingresado en el archivo de operadores es válido corroborando su existencia en el mundo real.

Métrica instanciada	
Exact_semantica_bool_Telefono_ope	
Métrica	Exact_semantica_bool
Datos	operadores.'Telefono'
Método Medición	Llamar al número ingresado al menos 3 veces. Si atienden y el número es correcto: exactitud = 1 Si no hay línea o el número está equivocado: exactitud = 0 Si hay línea pero nadie atiende: llamar nuevamente en el mismo día y 2 veces por día en los próximos 5 días. Si nadie atiende nunca: exactitud = 0 Si atienden pero el número está equivocado: exactitud = 0 Sino: exactitud = 1

Evaluar si el sitio web ingresado en el archivo de operadores es válido corroborando su existencia en el mundo real.

Métrica instanciada	
Exact_semantica_bool_Web_ope	
Métrica	Exact_semantica_bool

Datos	operadores.'Web'
Método Medición	if validar_website(Web) == True exactitud = 1 else: exactitud = 0

Se implementó una función que genera un request del sitio y evalúa su respuesta para determinar si el sitio es existente o no

```
def validar_website(url):
  if url.startswith("http"):
    url = url
  else:
    url = 'https://' + url

    try:
        response = requests.get(url)
        return response.status_code == 200
    except requests.exceptions.RequestException:
        return False
```

Métrica instanciada	
Exact_semantica_bool_Dir_ope	
Métrica	Exact_semantica_bool
Datos	operadores.'Direccion'
Método Medición	Buscar dirección en google maps. Si la dirección existe: exactitud = 1 else: exactitud = 0

Evaluar si el email ingresado en el archivo de operadores es válido corroborando su existencia en el mundo real.

Métrica instanciada	
Exact_semantica_bool_EMail_ope	
Métrica	Exact_semantica_bool

Datos	operadores.'EMail'
Método Medición	if validar_mail(EMail) == True exactitud = 1 else: exactitud = 0

Se implementó una función que envía un mail de prueba al email ingresado y devuelve True si el mail es enviado correctamente.

```python
def validar_mail(email):
    sender = "your_email@gmail.com"  # Colocar e-mail de remitente
    password = "your_password"  # Colocar contraseña de remitente

    # Crear mail
    message = MIMEText("Mail de prueba")
    message["Subject"] = "Mail de prueba"
    message["From"] = sender
    message["To"] = email

    try:
        # Conectarse a SMTP server
        smtp_server = smtplib.SMTP("smtp.gmail.com", 587)
        smtp_server.starttls()

        # Iniciar sesión a mail
        smtp_server.login(sender, password)

        # Mandar mail
        smtp_server.sendmail(sender, email, message.as_string())

        # Cerrar la conexión
        smtp_server.quit()

        print("Mail enviado con éxito.")
        return True
    except Exception as e:
        print(f"Error en envío de mail {e}")
        return False
```

Basándonos en el set de datos de Localidades de Uruguay descargado del catálogo de datos abiertos realizamos una comparativa de la exactitud semántica de los datos de Departamento y Localidades ingresados en operadores.
Se elaboran dos funciones que evalúan por separado los valores ingresados en Departamentos y Localidades

```
def val_depto(depto):
  for index, row in df_localidades.iterrows():
    if row['departamento'] == depto.upper():
      return True

  return False
```

```
def val_localidad(localidad):
  for index, row in df_localidades.iterrows():
    if row['localidad'] == localidad.upper():
      return True

  return False
```

Exact_semantica_bool_Depto_ope	
Métrica	Exact_semantica_bool
Datos	operadores.'Departamento'
Método Medición	if val_depto(Departamento) == True exactitud = 1 else: exactitud = 0

Exact_semantica_bool_Localidad_ope	
Métrica	Exact_semantica_bool
Datos	operadores.'Localidad'
Método Medición	if val_localidad(Localidad) == True exactitud = 1 else: exactitud = 0

Nota: En las métricas de consistencia se evaluarán la integridad entre las dos columnas con la finalidad de corroborar que la Localidad ingresada corresponde al Departamento ingresado

2. COMPLETITUD

Cobertura - Métricas Instanciadas

No se cuenta con un referencial adecuado para realizar la comparación, por lo que no se realiza la métrica

Densidad - Métricas Instanciadas

Métrica instanciada	
Completitud_densidad_Tipo_Operador_Ope	
Métrica	Completitud_densidad
Datos	operadores.TipoOperador
Método Medición	densidad = suma_nulos(df['TipoOperador'])

Métrica instanciada	
Completitud_densidad_Operador_Ope	
Métrica	Completitud_densidad
Datos	operadores.Operador
Método Medición	densidad = suma_nulos(df['Operador'])

Métrica instanciada	
Completitud_densidad_Departamento_Ope	
Métrica	Completitud_densidad
Datos	operadores.Departamento
Método Medición	densidad = suma_nulos(df[Departamento])

Métrica instanciada	
Completitud_densidad_Localidad_Ope	
Métrica	Completitud_densidad
Datos	operadores.Localidad

Método Medición	densidad = suma_nulos(df[Localidad])

Métrica instanciada	
Completitud_densidad_Direccion_Ope	
Métrica	Completitud_densidad
Datos	operadores.Direccion
Método Medición	densidad = suma_nulos(df[Direccion])

Métrica instanciada	
Completitud_densidad_Telefono_Ope	
Métrica	Completitud_densidad
Datos	operadores.Telefono
Método Medición	densidad = suma_nulos(df[Telefono])

Métrica instanciada	
Completitud_densidad_Web_Ope	
Métrica	Completitud_densidad
Datos	operadores.Web
Método Medición	densidad = suma_nulos(df[Web])

Métrica instanciada	
Completitud_densidad_EMail_Ope	
Métrica	Completitud_densidad
Datos	operadores.EMail

Método Medición	densidad = suma_nulos(df[EMail])

Métrica instanciada	
Completitud_densidad_Latitud_Ope	
Métrica	Completitud_densidad
Datos	operadores.Latitud
Método Medición	densidad = suma_nulos(df[Latitud])

Métrica instanciada	
Completitud_densidad_Longitud_Ope	
Métrica	Completitud_densidad
Datos	operadores.Longitud
Método Medición	densidad = suma_nulos(df[Longitud])

Nota: La métrica es definida en base a la función suma_nulos, la cual fue presentada anteriormente pero se repite a continuación.

```
def suma_nulos(column):
    null_count = column.isnull().sum()
    sin_datos_count = (column == "Sin Datos").sum()
    desc_sin_datos_count = (column == "Desconocido / Sin Datos").sum()
    s_d_count = (column == "S/D").sum()

    cant_nulos = null_count + sin_datos_count + desc_sin_datos_count + s_d_count
    return round(cant_nulos/len(column),4)
```

3. FRESCURA

Actualidad - Métricas Instanciadas

Dado que la frecuencia de actualización de la base de datos es diaria, y que los datos sobre operadores turísticos no deberían cambiar significativamente día a día, se entiende que los datos siempre estarán actualizados mientras se utilice la última tabla disponible.

4. CONSISTENCIA

Consistencia, integridad intra - relación - Métricas Instanciadas

Para los datos de Departamentos y Localidades del archivo de operadores es necesario también evaluar si la tupla de celdas es válida. Para poder corroborar la información utilizaremos el set de datos descargado del catálogo de datos abiertos donde tenemos todas las Localidades del País con su correspondiente Departamento.

Consistencia_Intra_Relación_Depto_y_Localidad_ope	
Métrica	Consistencia_Intra_Relacion_conj_celdas
Datos	operadores.'Departamento',operadores.'Localidad'
Método Medición	if val_depto_y_localidad(Departamento,Localidad) == True exactitud = 1 else: exactitud = 0

```
def val_depto_y_localidad(depto,localidad):
  for index, row in df_localidades.iterrows():
    if row['departamento'] == depto.upper() and row['localidad'] == localidad.upper():
      return True

  return False
```

Consistencia, integridad de dominio - Métricas Instanciadas

Para los datos de Latitud y Longitud de los valores ingresados en el archivo “operadores.csv” definimos una función que pretende validar por lo menos que el dato ingresado posiblemente pertenezca a coordenadas de Uruguay.

```
#Latitudes: entre -30.130153 y -35.061084
#Longitud: entre -58.483502 y -53.082044
def longitud_esperada(val):
  val = val.replace(",", ".")
  if float(val) <= -58.483502 and float(val) >= -53.082044:
    return True
  else:
    return False

def latitud_esperada(val):
  val = val.replace(",", ".")
  if float(val) <= -35.061084 and float(val) >= -30.130153:
    return True
  else:
    return False
```

Métrica instanciada	
Consistencia_Int_Dominio_Latitud_uy_ope	
Métrica	Consistencia_Int_Dominio
Datos	operadores.'Latitud'
Método Medición	if latitud_esperada(Latitud) == True exactitud = 1 else: exactitud = 0

Métrica instanciada	
Consistencia_Int_Dominio_Longitud_uy_ope	
Métrica	Consistencia_Int_Dominio
Datos	operadores.'Longitud'
Método Medición	if longitud_esperada(Longitud) == True exactitud = 1 else: exactitud = 0

5. UNICIDAD

No duplicación - Métricas Instanciadas

Métrica instanciada	
Unicidad_no_duplicacion_tupla_operadores	
Métrica	Unicidad_no_duplicacion_tupla
Datos	Operadores.TipoOperador,Operadores.Operador, Operadores.Departamento,Operadores.Localidad, Operadores.Direccion,Operadores.Telefono,Operadores.Web, Operadores.EMail,Operadores.Longitud,Operadores.Latitud
Método Medición	unicidad = duplicados_row(indice)['no_duplicacion']

Métrica instanciada	
Unicidad_no_contradiccion_tupla_operadores	
Métrica	Unicidad_no_contradiccion_tupla
Datos	Operadores.TipoOperador,Operadores.Operador, Operadores.Departamento,Operadores.Localidad, Operadores.Direccion,Operadores.Telefono,Operadores. Web,Operadores.EMail,Operadores.Longitud,Operadores. Latitud
Método Medición	unicidad = duplicados_row(indice)['no_contradiccion']

Nota: Se utiliza la misma función que para el caso de emisivos, pero modificando las columnas que no se desea considerar para la búsqueda de duplicados, (en este caso se desea considerar todas).

```python
def duplicados_row (df,row):
    subset_columns = list(df.columns)
    cols_not_to_consider = []

    # Genero listado de columnas a partir de las cuales se buscan duplicados
    for col in cols_not_to_consider:
        subset_columns.pop(subset_columns.index(col))

    # Genero dataframe con duplicados
    df_duplicates = df[df.duplicated(subset=subset_columns,keep=False)].sort_values(by = subset_columns)

    df_dup_row = df_duplicates[(df_duplicates[subset_columns] == df.loc[row][subset_columns]).all(axis=1)] # Dataframe con duplicados para cada registro duplicado

    if len(df_dup_row) == 0:
        no_duplicacion = 1
        no_contradiccion = 1
        return {'no_duplicacion': no_duplicacion, 'no_contradiccion': no_contradiccion}

    # Genero listas con los atributos que definiran si el duplicado es contradictorio o no

    check_list = [[]]

    to_remove = [None, 'Sin Datos', 'Desconocido / Sin Datos', 'S/D'] # valores a remover para chequear si los restantes son contradictorios

    # Chequeo si los duplicados son contradictorios
    for lista in check_list:
        contradiccion = []
        for value in to_remove:
            lista.remove(value) if value in lista else lista
        if len(lista) == len(set(lista)):
            contradiccion.append(True) # Existen valores distintos luego de sacar los nulos o faltantes
        else:
            contradiccion.append(False) # No existen valores distintos luego de sacar los nulos o faltantes

    if True in contradiccion:
         no_duplicacion = 1
         no_contradiccion = 0
    else:
        no_duplicacion = 0
        no_contradiccion = 1

    return {'no_duplicacion': no_duplicacion, 'no_contradiccion': no_contradiccion}
```

Receptivos

1. EXACTITUD

Exactitud sintáctica - Métricas Instanciadas

Métrica instanciada	
Exact_sintactica_bool_Departamento_receptivos	
Métrica	Exact_sintactica_bool_json
Datos	receptivos.Departamento
Método Medición	if Departamento not in list_departamentos or Departamento != ‘Transito’ exactitud = 0 else: exactitud = 1

Métrica instanciada	
Exact_sintactica_bool_Otro_Departamento_receptivos	
Métrica	Exact_sintactica_bool_json
Datos	receptivos.’Otro Departamento’
Método Medición	if ‘Otro Departamento’ not in list_departamentos or Departamento != ‘Transito’ or Departamento != ‘No corresponde’ exactitud = 0 else: exactitud = 1

Nota: list_departamentos corresponde a una lista exhaustiva de los departamentos del Uruguay. Dicha lista es indicada a continuación. Se agrega el valor “Transito” y “No corresponde” a los valores posibles ya que se entiende que dichos valores son correctos.
list_departamentos = ['Montevideo', 'Maldonado', 'Canelones', 'Salto', 'Lavalleja', 'San Jose', 'Rio Negro', nan, 'Durazno', 'Cerro Largo', 'Colonia', 'Tacuarembo', 'Treinta y Tres', 'Rocha', 'Artigas', 'Paysandu', 'Rivera', 'Flores', 'Soriano', 'Florida']

Métrica instanciada	
Exact_sintactica_bool_FechaIngreso_receptivos	
Métrica	Exact_sintactica_bool_json
Datos	receptivos.'FechaIngreso'
Método Medición	if check_date_format(FechaIngreso) == True exactitud = 1 else: exactitud = 0

Métrica instanciada	
Exact_sintactica_bool_FechaEgreso_receptivos	
Métrica	Exact_sintactica_bool_json
Datos	receptivos.'FechaEgreso'
Método Medición	if check_date_format(FechaEgreso) == True exactitud = 1 else: exactitud = 0

Nota: la función referida en el método de medición es la siguiente:

```
def check_date_format(string):
    pattern = r"\d{4}-\d{2}-\d{2}"
    match = re.fullmatch(pattern, string)
    if match:
        return True
    else:
        return False
```

Exactitud semántica - Métricas Instanciadas

Métrica instanciada	
Exact_semántica_bool_`Lugar Ingreso`_receptivos	
Métrica	Exact_semantica_bool_json
Datos	receptivos.`Lugar Ingreso`
Método Medición	if `Lugar Ingreso` not in list_lugares exactitud = 0 else: exactitud = 1

Métrica instanciada	
Exact_semántica_bool_`Lugar Egreso`_receptivos	
Métrica	Exact_semantica_bool_json
Datos	receptivos.`Lugar Egreso`
Método Medición	if `Lugar Egreso` not in list_lugares exactitud = 0 else: exactitud = 1

Nota: list_lugares corresponde a una lista de puntos de ingreso y egreso al país que se toma como referencial. Dicha lista es indicada a continuación.
list_lugares_salida = ['Chuy', 'Río Branco', 'Aeropuerto de Carrasco', 'Aeropuerto de Punta del Este', 'Fray Bentos', 'Bella Unión', 'Colonia', 'Salto', 'Paysandú', 'Carmelo', 'Melo - Aceguá', 'Artigas', 'Rivera', 'Puerto de montevideo', 'Nueva Palmira']

Métrica instanciada	
Exact_semantica_bool_Localidad_receptivos	
Métrica	Exact_semantica_bool_json
Datos	receptivos.'Localidad'

Método Medición	if val_localidad(Localidad) == True exactitud = 1 else: exactitud = 0

Métrica instanciada	
Exact_semantica_bool_Otra_Localidad_receptivos	
Métrica	Exact_semantica_bool_json
Datos	receptivos.'Otra Localidad'
Método Medición	if val_localidad('Otra Localidad') == True exactitud = 1 else: exactitud = 0

Nota: Se utiliza la función de validación de localidad presentada anteriormente para la validación de los atributos de "Localidad" y "Otra Localidad"

Precisión - Métricas Instanciadas

La métrica se aplica a la columna de País de la siguiente manera:

Métrica instanciada	
Exact_precision_Pais_receptivos	
Métrica	Exact_precision_json
Datos	receptivos.Pais

Método Medición	if Pais is pais:
	exactitud = 1.00
	elif Pais is Gran Bretaña:
	exactitud = 0.75
	elif "Otro" in Pais:
	exactitud = 0.50
	elif Pais is “Africa u Oceania”
	exactitud = 0.25
	else:
	exactitud = 0

Nota: nuevamente, la adjudicación de un valor de precisión menor para “Gran Bretaña” es realizada en base al entendido de que “Gran Bretaña” hace referencia a cualquiera de los países que forman la región (Inglaterra, Escocia y Gales).

2. COMPLETITUD

Cobertura - Métricas Instanciadas

Métrica instanciada	
Completitud_cobertura_receptivos	
Métrica	Completitud_cobertura_json
Datos	receptivos
Método Medición	cobertura = sum(receptivos[‘Gente’]) / L_ref_estim

Nota: sum(receptivos[‘gente’]) es la cantidad de turistas registrados por la encuesta, idealmente luego de identificar y filtrar los duplicados. Asimismo, L_ref_estim corresponde a una estimación de la cantidad real de turistas que viajaron en el período Dic/16 - Mayo/23. Esta cantidad es estimada en base al informe de turismo receptivo del Ministerio de Turismo del Uruguay, obtenido de la siguiente web: https://www.gub.uy/ministerio-turismo/datos-y-estadisticas/estadisticas?page=0
La cantidad de turistas por año se resume en la siguiente tabla:

Dic 2016	94.131	No hay datos, se estima como dic-21
2017	3.940.790	
2018	3.711.948	
2019	3.220.602	
2020	1.000.908	1er trimestre (hasta comienzo de pandemia)

2021	233.505	Noviembre y diciembre (“fin de pandemia”)
2022	2.466.929	
2023	391.683	1er trimestre
Tot	15.060.496	**Valor asignado para L_ref_estim**

Densidad - Métricas Instanciadas

Métrica instanciada	
Completitud_densidad_IdIngresos_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdIngresos
Método Medición	densidad = suma_nulos(df[IdIngresos])

Métrica instanciada	
Completitud_densidad_Lugar_Ingreso_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.”Lugar Ingreso”
Método Medición	densidad = suma_nulos(df[“Lugar Ingreso”])

Métrica instanciada	
Completitud_densidad_IdTranspIngreso_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdTranspIngreso
Método Medición	densidad = suma_nulos(df[IdTranspIngreso])

Métrica instanciada	
Completitud_densidad_Transporte_Internacional_Ingreso_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.”Transporte Internacional de Ingreso”
Método Medición	densidad = suma_nulos(df[”Transporte Internacional de Ingreso”])

Métrica instanciada	
Completitud_densidad_FechaIngreso_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.FechaIngreso
Método Medición	densidad = suma_nulos(df[FechaIngreso])

Métrica instanciada	
Completitud_densidad_IdFecIng_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdFecIng
Método Medición	densidad = suma_nulos(df[IdFecIng])

Métrica instanciada	
Completitud_densidad_FechaEgreso_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.FechaEgreso
Método Medición	densidad = suma_nulos(df[FechaEgreso])

Métrica instanciada	
Completitud_densidad_IdFecEgr_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdFecEgr
Método Medición	densidad = suma_nulos(df[IdFecEgr])

Métrica instanciada	
Completitud_densidad_IdNacionalidad_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdNacionalidad
Método Medición	densidad = suma_nulos(df[IdNacionalidad])

Métrica instanciada	
Completitud_densidad_Pais_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.Pais
Método Medición	densidad = suma_nulos(df[Pais])

Métrica instanciada	
Completitud_densidad_IdResidencia_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdResidencia
Método Medición	densidad = suma_nulos(df[IdResidencia])

Métrica instanciada	
Completitud_densidad_Residencia_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.Residencia
Método Medición	densidad = suma_nulos(df[Residencia])

Métrica instanciada	
Completitud_densidad_IdMotivo_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdMotivo
Método Medición	densidad = suma_nulos(df[IdMotivo])

Métrica instanciada	
Completitud_densidad_Motivo_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.Motivo
Método Medición	densidad = suma_nulos(df[Motivo])

Métrica instanciada	
Completitud_densidad_IdOcupacion_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdOcupacion
Método Medición	densidad = suma_nulos(df[IdOcupacion])

Métrica instanciada	
Completitud_densidad_Ocupacion_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.Ocupacion
Método Medición	densidad = suma_nulos(df[Ocupacion])

Métrica instanciada	
Completitud_densidad_IsEstudio_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IsEstudio
Método Medición	densidad = suma_nulos(df[IsEstudio])

Métrica instanciada	
Completitud_densidad_Estudio_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.Estudio
Método Medición	densidad = suma_nulos(df[Estudio])

Métrica instanciada	
Completitud_densidad_IdDestinoLocalidad_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdDestinoLocalidad
Método Medición	densidad = suma_nulos(df[IdDestinoLocalidad])

Métrica instanciada	
Completitud_densidad_Localidad_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.Localidad
Método Medición	densidad = suma_nulos(df[Localidad])

Métrica instanciada	
Completitud_densidad_IdDepartamentoDestino_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdDepartamentoDestino
Método Medición	densidad = suma_nulos(df[IdDepartamentoDestino])

Métrica instanciada	
Completitud_densidad_Departamento_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.Departamento
Método Medición	densidad = suma_nulos(df[Departamento])

Métrica instanciada	
Completitud_densidad_IdOtroDepartamento_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdOtroDepartamento
Método Medición	densidad = suma_nulos(df[IdOtroDepartamento])

Métrica instanciada	
Completitud_densidad_Otro_Departamento_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.”Otro Departamento”
Método Medición	densidad = suma_nulos(df[“Otro Departamento”])

Métrica instanciada	
Completitud_densidad_IdOtraLocalidad_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdOtraLocalidad
Método Medición	densidad = suma_nulos(df[IdOtraLocalidad)

Métrica instanciada	
Completitud_densidad_Otra_Localidad_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.”Otra Localidad”
Método Medición	densidad = suma_nulos(df[“Otra Localidad”])

Métrica instanciada	
Completitud_densidad_IdAlojamiento_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdAlojamiento
Método Medición	densidad = suma_nulos(df[IdAlojamiento])

Métrica instanciada	
Completitud_densidad_Alojamiento_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.Alojamiento
Método Medición	densidad = suma_nulos(df[Alojamiento])

Métrica instanciada	
Completitud_densidad_IdTranspLocal_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdTranspLocal
Método Medición	densidad = suma_nulos(df[IdTranspLocal])

Métrica instanciada	
Completitud_densidad_TransporteLocal_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.TransporteLocal
Método Medición	densidad = suma_nulos(df[TransporteLocal])

Métrica instanciada	
Completitud_densidad_IdEgresos_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdEgresos
Método Medición	densidad = suma_nulos(df[IdEgresos])

Métrica instanciada	
Completitud_densidad_Lugar_Egreso_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.”Lugar Egreso”
Método Medición	densidad = suma_nulos(df[“Lugar Egreso”])

Métrica instanciada	
Completitud_densidad_Transporte_Internacional_Egreso_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.”Transporte Internacional de Egreso”
Método Medición	densidad = suma_nulos(df[“Transporte Internacional de Egreso”])

Métrica instanciada	
Completitud_densidad_IdTranspEgreso_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdTranspEgreso
Método Medición	densidad = suma_nulos(df[IdTranspEgreso])

Métrica instanciada	
Completitud_densidad_IdDestino_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.IdDestino
Método Medición	densidad = suma_nulos(df[IdDestino])

Métrica instanciada	
Completitud_densidad_Destino_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.Destino
Método Medición	densidad = suma_nulos(df[Destino])

Métrica instanciada	
Completitud_densidad_Estadia_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.Estadia
Método Medición	densidad = suma_nulos(df[Estadia])

Métrica instanciada	
Completitud_densidad_Gente_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.Gente
Método Medición	densidad = suma_nulos(df[Gente])

Métrica instanciada	
Completitud_densidad_GastoTotal_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.GastoTotal
Método Medición	densidad = suma_nulos(df[GastoTotal])

Métrica instanciada	
Completitud_densidad_GastoAlojamiento_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.GastoAlojamiento
Método Medición	densidad = suma_nulos(df[GastoAlojamiento])

Métrica instanciada	
Completitud_densidad_GastoAlimentacion_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.GastoAlimentacion
Método Medición	densidad = suma_nulos(df[GastoAlimentacion])

Métrica instanciada	
Completitud_densidad_GastoTransporte_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.GastoTransporte
Método Medición	densidad = suma_nulos(df[GastoTransporte])

Métrica instanciada	
Completitud_densidad_GastoCultural_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.GastoCultural
Método Medición	densidad = suma_nulos(df[GastoCultural])

Métrica instanciada	
Completitud_densidad_GastoTours_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.GastoTours
Método Medición	densidad = suma_nulos(df[GastoTours])

Métrica instanciada	
Completitud_densidad_GastoCompras_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.GastoCompras
Método Medición	densidad = suma_nulos(df[GastoCompras])

Métrica instanciada	
Completitud_densidad_GastoOtros_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.GastoOtros
Método Medición	densidad = suma_nulos(df[GastoOtros])

Métrica instanciada	
Completitud_densidad_Coef_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.Coef
Método Medición	densidad = suma_nulos(df[Coef])

Métrica instanciada	
Completitud_densidad_CoefTot_Receptivos	
Métrica	Completitud_densidad_json
Datos	receptivos.CoefTot
Método Medición	densidad = suma_nulos(df[CoefTot])

Nota: La métrica es definida en base a la función suma_nulos, la cual es presentada a continuación.

```
def suma_nulos(column):
    null_count = column.isnull().sum()
    sin_datos_count = (column == "Sin Datos").sum()
    desc_sin_datos_count = (column == "Desconocido / Sin Datos").sum()
    s_d_count = (column == "S/D").sum()

    cant_nulos = null_count + sin_datos_count + desc_sin_datos_count + s_d_count
    return round(cant_nulos/len(column),4)
```

Cabe destacar que, en este caso, la variable “column” corresponde a una lista con los valores del atributo correspondiente de todos los diccionarios del documento. La longitud de dicha columna es igual a la cantidad total de diccionarios. La lista deberá contener un valor nulo por cada diccionario para el cual el atributo en cuestión no esté definido.

3. FRESCURA

Actualidad - Métricas Instanciadas

Métrica instanciada	
ratio_actualidad_receptivos	
Métrica	ratio_actualidad_json
Datos	receptivos
Método Medición	frescura = 1 - (t2-t0) / Δt0

Nota:

- t0: fecha de actualización de tabla utilizada
- t2: fecha de realización de consulta
- Δt0: período de tiempo entre actualizaciones de tabla (inverso de frecuencia

de actualización).

Dado que se asume que hay turistas entrando y saliendo del país todos los días, la información de la tabla queda desactualizada respecto del mundo real en el momento en el cual se publica. Utilizando “días” como unidad de medida, dado que la actualización de los datos es anual según el Catálogo Nacional de Datos Abiertos, se tiene que Δt0 = 365 días. Luego, si se realiza la consulta en el momento de actualización de los datos (t2 = t0), la métrica toma el valor de 1. A medida que la consulta se realiza días después de la última actualización, t2 aumenta y por tanto t2-t0 aumenta, lo que reduce la métrica de actualidad. De esta manera, si se realiza la consulta el último día antes de la actualización de la tabla (t2-t0 = Δt0), la métrica toma el valor de 0. Dado que la actualización es periódica, la métrica debe tomar el valor nulo si se realiza la consulta justo antes de la siguiente actualización de la tabla, ya que realizar la consulta un día después implicaría tener los datos lo más actualizados posible.

4. CONSISTENCIA

Consistencia, integridad intra - relación - Métricas Instanciadas

Métrica instanciada	
Consistencia_Intra_Relacion_IdFecIng_receptivos	
Métrica	Consistencia_Intra_Relacion_json
Datos	receptivos.'IdFecIng'

Método Medición	if validarDistintosFecIng(IdFecIng) > 1:
	consistencia = 0
	else:
	consistencia = 1

Métrica instanciada	
Consistencia_Intra_Relacion_IdFecEgr_receptivos	
Métrica	Consistencia_Intra_Relacion_json
Datos	receptivos.'IdFecEgr'

Método Medición	if validarDistintosFecEgr(IdFecEgr) > 1:
	consistencia = 0
	else:
	consistencia = 1

Para evaluar la consistencia de los id y saber si tenemos identificadores ligados a dos valores distintos se elaboran dos funciones que tiene como finalidad dado un id

de fecha obtener la cantidad de valores distintos que hay en su valor de Columna Descriptivo

```
def validarDistintosFecIng(valorId):
  result = df_receptivos.loc[df_receptivos['IdFecIng'] == valorId]\
           .groupby('IdFecIng')['FechaIngreso'].nunique().reset_index(name='CantDistintos')
  for index,row in result.iterrows():
    if row['CantDistintos'] > 1:
      return 0
    else:
      return 1

def validarDistintosFecEgr(valorId):
  result = df_receptivos.loc[df_receptivos['IdFecEgr'] == valorId]\
           .groupby('IdFecEgr')['FechaEgreso'].nunique().reset_index(name='CantDistintos')
  for index,row in result.iterrows():
    if row['CantDistintos'] > 1:
      return 0
    else:
      return 1
```

Métrica instanciada	
Consistencia_Intra_Relacion_Fechas_Estadia_Receptivos	
Métrica	Consistencia_Intra_Relacion_conj_valores_json
Datos	receptivos.FechaIngreso, receptivos.FechaEgreso, receptivos.Estadia

Método Medición	try:
	if FechaEgreso - FechaIngreso != Estadia
	consistencia = 0
	else:
	consistencia = 1
	except:
	consistencia = 1

Nota: Como puede verse en el método de medición planteado, en caso de que algún diccionario no cuente con los tres atributos en simultáneo, se entiende que no habría problemas de inconsistencia, de manera que el valor de la métrica es 1.

La misma métrica es también instanciada para evaluar la consistencia entre las columnas de “Lugar Egreso” y “Transporte Internacional de Egreso”, y entre las columnas de “Lugar Ingreso” y “Transporte Internacional de Ingreso”

Métrica instanciada

Consistencia_Intra_Relacion_lugar_transporte_receptivos_ingreso	
Métrica	Consistencia_Intra_Relacion_conj_valores_json
Datos	receptivos.`Lugar Ingreso`, receptivos.`Transporte Internacional de Ingreso`

Método Medición	try:
	if (`Lugar Ingreso` == 'Aeropuerto de Carrasco' or 'Lugar Ingreso' == 'Aeropuerto de Punta del Este') and `Transporte Internacional de Ingreso` != 'Aereo'
	consistencia = 0
	elif `Lugar Ingreso` == 'Puerto de montevideo' and `Transporte Internacional de Ingreso` != 'Maritimo - Fluvial'
	consistencia = 0
	else:
	consistencia = 1
	except:
	consistencia = 1

Métrica instanciada	
Consistencia_Intra_Relacion_lugar_transporte_receptivos_egreso	
Métrica	Consistencia_Intra_Relacion_conj_valores_json
Datos	receptivos.`Lugar Egreso`, receptivos.`Transporte Internacional de Egreso`

Método Medición	try:
	if (`Lugar Egreso` == 'Aeropuerto de Carrasco' or 'Lugar Egreso == 'Aeropuerto de Punta del Este') and `Transporte Internacional de Egreso` != 'Aereo'
	consistencia = 0
	elif `Lugar Egreso` == 'Puerto de montevideo' and `Transporte Internacional de Egreso` != 'Maritimo - Fluvial'
	consistencia = 0
	else:
	consistencia = 1
	except:
	consistencia = 1

Nota: Como puede verse en el método de medición planteado, en caso de que algún diccionario no cuente con alguno de los atributos correspondientes, se entiende que no habría problemas de inconsistencia, de manera que el valor de la métrica es 1.

Métrica instanciada	
Consistencia_Intra_Relacion_Otro_Departamento_receptivos	
Métrica	Consistencia_Intra_Relacion_conj_valores_json
Datos	receptivos.IdOtroDepartamento, receptivos.'Otro Departamento'

Método Medición	if (select count(distinct(`Otro Departamento`))
	from receptivos
	where IdOtroDepartamento = id) > 1:
	consistencia = 0
	else:
	consistencia = 1

Métrica instanciada	
Consistencia_Intra_Relacion_Otra_Localidad	
Métrica	Consistencia_Intra_Relacion_conj_valores_json
Datos	receptivos.IdOtraLocalidad, receptivos.'Otra Localidad'

Método Medición	if (select count(distinct(`Otra Localidad`))
	from receptivos
	where IdOtraLocalidad = id) > 1:
	consistencia = 0
	else:
	consistencia = 1

Nota: la variable "id" indicada en el método de medición indica el valor de IdOtroDepartamento o IdOtraLocalidad para el cual se quiere realizar la medición. Es decir, si un valor de id está asociado a más de un valor distinto del atributo correspondiente, la medida de consistencia será nula.

5. UNICIDAD

Métrica instanciada	
Unicidad_no_duplicacion_documento_receptivos	
Métrica	Unicidad_no_duplicacion_documento_json
Datos	receptivos.IdIngresos,receptivos.`Lugar Ingreso`, receptivos.IdTranspIngreso, receptivos.`Transporte Internacional de Ingreso`, receptivos.FechaIngreso, receptivos.IdFecIng,receptivos.FechaEgreso,receptivos.IdFecEgr, receptivos.IdNacionalidad,receptivos.Pais,receptivos.IdResidencia, receptivos.Residencia,receptivos.IdMotivo,receptivos.Motivo, receptivos.IdOcupacion,receptivos.Ocupacion, receptivos.IsEstudio, receptivos.Estudio, receptivos.IdDestinoLocalidad,receptivos.Localidad, receptivos.IdDepartamentoDestino,receptivos.Departamento, receptivos.IdOtroDepartamento,receptivos.`Otro Departamento`, receptivos.IdOtraLocalidad,receptivos.`Otra Localidad`, receptivos.IdAlojamiento,receptivos.Alojamiento, receptivos.IdTranspLocal, receptivos.TransporteLocal, receptivos.IdEgresos, receptivos.`Lugar Egreso` ,receptivos.IdTranspEgreso, receptivos.`Transporte Internacional de Egreso`, receptivos.IdDestino, receptivos.Destino,receptivos.Estadia,receptivos.Gente, receptivos.GastoTotal,receptivos.GastoAlojamiento, receptivos.GastoAlimentacion,receptivos.GastoTransporte,receptivos.GastoCultural,receptivos.GastoTours,receptivos.GastoCompras,receptivos.GastoOtros, receptivos.Coef,receptivos.CoefTot
Método Medición	unicidad = duplicados_row(indice)['no_duplicacion']

Métrica instanciada	
Unicidad_no_contradiccion_documento_receptivos	
Métrica	Unicidad_no_contradiccion_documento_json

Datos	receptivos.IdIngresos,receptivos.`Lugar Ingreso`, receptivos.IdTranspIngreso, receptivos.`Transporte Internacional de Ingreso`, receptivos.FechaIngreso, receptivos.IdFecIng,receptivos.FechaEgreso,receptivos.IdFecEgr, receptivos.IdNacionalidad,receptivos.Pais,receptivos.IdResidencia, receptivos.Residencia,receptivos.IdMotivo,receptivos.Motivo, receptivos.IdOcupacion,receptivos.Ocupacion, receptivos.IsEstudio, receptivos.Estudio, receptivos.IdDestinoLocalidad,receptivos.Localidad, receptivos.IdDepartamentoDestino,receptivos.Departamento, receptivos.IdOtroDepartamento,receptivos.`Otro Departamento`, receptivos.IdOtraLocalidad,receptivos.`Otra Localidad`, receptivos.IdAlojamiento,receptivos.Alojamiento, receptivos.IdTranspLocal, receptivos.TransporteLocal, receptivos.IdEgresos, receptivos.`Lugar Egreso` ,receptivos.IdTranspEgreso, receptivos.`Transporte Internacional de Egreso`, receptivos.IdDestino, receptivos.Destino,receptivos.Estadia,receptivos.Gente, receptivos.GastoTotal,receptivos.GastoAlojamiento, receptivos.GastoAlimentacion,receptivos.GastoTransporte ,receptivos.GastoCultural,receptivos.GastoTours,receptivos.GastoCompras,receptivos.GastoOtros, receptivos.Coef,receptivos.CoefTot
Método Medición	unicidad = duplicados_row(indice)['no_contradiccion']

Especificación del Modelo

Especificación modelo Emisivos y Operadores

Especificación de modelo de calidad - Emisivos.csv y Receptivos.csv					
Dimensión	**Factor**	**Métrica**	**Granularida**	**Métrica Instanciada**	**Método Medición**
Exactitud	Exactitud sintáctica	Exact_sintactica_bool	Celda	Exact_sintactica_bool_FechaSalida	if check_date_format(FechaSalida) == True exactitud = 1 else: exactitud = 0
				Exact_sintactica_bool_Departamento	if Departamento not in list_departamentos exactitud = 0 else: exactitud = 1
				Exact_sintactica_bool_telefono_ope	if val_tel(telefono) == True exactitud = 1 else: exactitud = 0
				Exact_sintactica_bool_Sitio_web_ope	if es_sitio_web(Web) == True exactitud = 1 else: exactitud = 0
				Exact_sintactica_bool_Email_ope	if es_correo_electronico(Email) == True exactitud = 1 else: exactitud = 0
		Exact_sintactica_dist	Celda	Exact_sintactica_dist_direccion	exactitud = 1 - min_levenshtein_distance (Direccion, referencial)
	Exactitud semántica	Exact_semantica_bool	Celda	Exact_semántica_bool_`Lugar Salida`	if `Lugar Salida` not in list_lugares_salida exactitud = 0 else: exactitud = 1
				Exact_semantica_bool_Telefono_ope	Llamar al número ingresado al menos 3 veces. Si no hay línea o el número está equivocado: exactitud = 0 Si hay línea pero nadie atiende: llamar nuevamente en el mismo día y 2 veces por día en los próximos 5 días. Si nadie atiende nunca: exactitud = 0 Si atienden pero el número está equivocado: exactitud = 0 Sino: exactitud = 1
				Exact_semantica_bool_Web_ope	if validar_website(Web) == True exactitud = 1 else: exactitud = 0
				Exact_sintactica_bool_Dir_ope	Buscar dirección en google maps. Si la dirección existe: exactitud = 1 else: exactitud = 0
				Exact_semantica_bool_EMail_ope	if validar_mail(EMail) == True exactitud = 1 else: exactitud = 0
				Exact_semantica_bool_Depto_ope	if val_depto(Departamento) == True exactitud = 1 else: exactitud = 0
				Exact_semantica_bool_Localidad_ope	if val_localidad(Localidad) == True exactitud = 1 else: exactitud = 0

	Precisión	Exact_precision	Celda	Exact_precision_Destino	if Destino is pais:
					precision = 1.00
					elif Destino is region:
					precision = 0.66
					elif Destino is continente:
					precision = 0.33
					else:
					precision = 0

				Exact_precision_Pais	if Pais is pais:
					exactitud = 1.00
					elif Pais is Gran Bretaña:
					exactitud = 0.66
					elif "Otro" in Pais:
					exactitud = 0.33
					else:
					exactitud = 0

Completitud	Cobertura	Completitud_cobertura	Tabla	Completitud_cobertura_emisivos	cobertura = sum(emisivos['Gente']) / L_ref_estim
	Densidad	Completitud_densidad	Columna	Completitud_densidad_Lugar Salida_emisivos	densidad = suma_nulos(df['Lugar Salida'])
				Completitud_densidad_IdDeptoResidencia_emisivos	densidad = suma_nulos(df['IdDeptoResidencia'])
				Completitud_densidad_Departamento_emisivos	densidad = suma_nulos(df['Departamento'])
				Completitud_densidad_IdMotivo_emisivos	densidad = suma_nulos(df['IdMotivo'])
				Completitud_densidad_Motivo_emisivos	densidad = suma_nulos(df['Motivo'])
				Completitud_densidad_Ocupacion_emisivos	densidad = suma_nulos(df['Ocupacion'])
				Completitud_densidad_Estudio_emisivos	densidad = suma_nulos(df['Estudio'])
				Completitud_densidad_Destino_emisivos	densidad = suma_nulos(df['Destino'])
				Completitud_densidad_Alojamiento_emisivos	densidad = suma_nulos(df['Alojamiento'])
				Completitud_densidad_Trasporte Local_emisivos	densidad = suma_nulos(df['Trasporte Local'])
				Completitud_densidad_Tipo_Operador_Ope	densidad = suma_nulos(df['TipoOperador'])
				Completitud_densidad_Operador_Ope	densidad = suma_nulos(df['Operador'])
				Completitud_densidad_Departamento_Ope	densidad = suma_nulos(df[Departamento])
				Completitud_densidad_Localidad_Ope	densidad = suma_nulos(df[Localidad])
				Completitud_densidad_Direccion_Ope	densidad = suma_nulos(df[Direccion])
				Completitud_densidad_Telefono_Ope	densidad = suma_nulos(df[Telefono])
				Completitud_densidad_Web_Ope	densidad = suma_nulos(df[Web])
				Completitud_densidad_EMail_Ope	densidad = suma_nulos(df[EMail])

				Completitud_densidad_EMail_Ope	densidad = suma_nulos(df[EMail])
				Completitud_densidad_Latitud_Ope	densidad = suma_nulos(df[Latitud])
				Completitud_densidad_Longitud_Ope	densidad = suma_nulos(df[Longitud])
Frescura	Actualidad	ratio_actualidad	Tabla	ratio_actualidad_emisivos	frescura = 1 - (t2-t0) / Δt0

	Integridad intra - relación	Consistencia_Intra_Relacion	Celda	Consistencia_Intra_Relacion_Lugar.Salida_emisivos	if 'Lugar Salida' != dict_lugares_salida[IdLugarSalida]:
					  consistencia = 0
					else:
					  consistencia = 1

		Consistencia_Intra_Relacion_columna	Columna	Consistencia-Intra-Relacion_columna-Lugar.Salida_Emisivos	if set(df['Lugar Salida'].unique()) != set(df['Lugar Ingreso'].unique())
					  consistencia = 0
					else:
					  consistencia = 1

				Consistencia_Intra_Relacion_Fechas_Estadia_Emisivos	if FechaEntrada - FechaSalida != Estadia
					  consistencia = 0
					else:
					  consistencia = 1

				Consistencia_Intra_Relacion_Gastos_Emisivos	if round(GastoAlojamiento + GastoAlimentacion + GastoCompras +
					  GastoCultural + GastoResto + GastoTours +
					  GastoTransporteInternac +
					  GatoTransporteLocal,0) != round(GastoTotal,0)
					  consistencia = 0
					else:
					  consistencia = 1

Consistencia	Integridad intra - relación	Consistencia_Intra_Relacion_conj_celdas	Grupo Celdas	Consistencia_Intra_Relacion_lugar_transporte_emisivos_salida	if `Lugar Salida` == 'Aeropuerto de Carrasco'
					  and `Transporte Internacional de Salida` != 'Aereo'
					  consistencia = 0
					elif `Lugar Salida` == 'Puerto de montevideo'
					  and `Transporte Internacional de Salida` != 'Maritimo - Fluvial'
					  consistencia = 0
					else:
					  consistencia = 1

				Consistencia_Intra_Relacion_lugar_transporte_emisivos_ingreso	if `Lugar Ingreso` == 'Aeropuerto de Carrasco
					  and `Transporte Internacional de Ingreso` != 'Aereo'
					  consistencia = 0
					elif `Lugar Ingreso` == 'Puerto de montevideo'
					  and `Transporte Internacional de Ingreso` != 'Maritimo - Fluvial'
					  consistencia = 0
					else:
					  consistencia = 1

				Consistencia_Intra_Relación_Depto_y_Localidad_ope	if val_depto_y_localidad(Departamento,Localidad) == True
					  exactitud = 1
					else:
					  exactitud = 0

	Integridad de dominio	Consistencia_Int_Dominio	Celda	Consistencia_Int_Dominio_Latitud_emisivos	consistencia = 1 - dist_lat_norm (latitud, departamento)
				Consistencia_Int_Dominio_Longitud_emisivos	consistencia = 1 - dist_lon_norm (latitud, departamento)

				Consistencia_Int_Dominio_Latitud_uy_ope	if latitud_esperada(Latitud) == True
					  exactitud = 1
					else:
					  exactitud = 0

				Consistencia_Int_Dominio_Longitud_uy_ope	if longitud_esperada(Longitud) == True
					  exactitud = 1
					else:
					  exactitud = 0

Unicidad	No duplicación	Unicidad_no_duplicacion_tupla	Tupla	Unicidad_no_duplicacion_tupla_emisivos	unicidad = duplicados_row(indice)['no_duplicacion']
				Unicidad_no_duplicacion_tupla_operadores	unicidad = duplicados_row(indice)['no_duplicacion']
	No contradicción	Unicidad_no_contradiccion_tupla	Tupla	Unicidad_no_contradiccion_tupla_emisivos	unicidad = duplicados_row(indice)['no_contradiccion']
				Unicidad_no_contradiccion_tupla_operadores	unicidad = duplicados_row(indice)['no_contradiccion']

Especificación modelo Receptivos

Especificación de modelo de calidad - Receptivos.json					
Dimensión	**Factor**	**Métrica**	**Granularidad**	**Métrica Instanciada**	**Método Medición**
Exactitud	Exactitud sintáctica	Exact_sintactica_bool_json	Valor	Exact_sintactica_bool_Departamento_receptivos	if Departamento not in list_departamentos or Departamento != 'Transito' exactitud = 0 else: exactitud = 1
				Exact_sintactica_bool_Otro_Departamento_receptivos	if 'Otro Departamento' not in list_departamentos or Departamento != 'Transito' or Departamento != 'No corresponde' exactitud = 0 else: exactitud = 1
				Exact_sintactica_bool_FechaIngreso_receptivos	if check_date_format(FechaIngreso) == True exactitud = 1 else: exactitud = 0
				Exact_sintactica_bool_FechaEgreso_receptivos	if check_date_format(FechaEgreso) == True exactitud = 1 else: exactitud = 0
	Exactitud semántica	Exact_semantica_bool_json	Valor	Exact_semántica_bool_`Lugar Ingreso`_receptivos	if `Lugar Ingreso` not in list_lugares exactitud = 0 else: exactitud = 1
				Exact_semántica_bool_`Lugar Egreso`_receptivos	if `Lugar Egreso` not in list_lugares exactitud = 0 else: exactitud = 1
				Exact_semantica_bool_Localidad_receptivos	if val_localidad(Localidad) == True exactitud = 1 else: exactitud = 0
				Exact_semantica_bool_Otra_Localidad_receptivos	if val_localidad('Otra Localidad') == True exactitud = 1 else: exactitud = 0

	Precisión	Exact_precision_json	Valor	Exact_precision_Pais_receptivos	if Pais is pais:
					exactitud = 1.00
					elif Pais is Gran Bretaña:
					exactitud = 0.75
					elif "Otro" in Pais:
					exactitud = 0.50
					elif Pais is "Africa u Oceania"
					exactitud = 0.25
					else:
					exactitud = 0

	Cobertura	Completitud_cobertura_json	Archivo	Completitud_cobertura_receptivos	cobertura = sum(receptivos['Gente']) / L_ref_estim
				Completitud_densidad_IdIngresos_Receptivos	densidad = suma_nulos(df[IdIngresos])
				Completitud_densidad_Lugar_Ingreso_Receptivos	densidad = suma_nulos(df["Lugar Ingreso"])
				Completitud_densidad_IdTranspIngreso_Receptivos	densidad = suma_nulos(df[IdTranspIngreso])
Completitud	Densidad	Completitud_densidad_json	Atributo	Completitud_densidad_Transporte_Internacional_Ingreso_Receptivos	densidad = suma_nulos(df["Transporte Internacional de Ingreso"])
				Completitud_densidad_FechaIngreso_Receptivos	densidad = suma_nulos(df[FechaIngreso])
				Completitud_densidad_IdFecIng_Receptivos	densidad = suma_nulos(df[IdFecIng])
				Completitud_densidad_FechaEgreso_Receptivos	densidad = suma_nulos(df[FechaEgreso])
				Completitud_densidad_IdFecEgr_Receptivos	densidad = suma_nulos(df[IdFecEgr])
				Completitud_densidad_IdNacionalidad_Receptivos	densidad = suma_nulos(df[IdNacionalidad])
				Completitud_densidad_Pais_Receptivos	densidad = suma_nulos(df[Pais])
				Completitud_densidad_IdResidencia_Receptivos	densidad = suma_nulos(df[IdResidencia])
				Completitud_densidad_Residencia_Receptivos	densidad = suma_nulos(df[Residencia])
				Completitud_densidad_IdMotivo_Receptivos	densidad = suma_nulos(df[IdMotivo])
				Completitud_densidad_Motivo_Receptivos	densidad = suma_nulos(df[Motivo])
				Completitud_densidad_IdOcupacion_Receptivos	densidad = suma_nulos(df[IdOcupacion])
				Completitud_densidad_Ocupacion_Receptivos	densidad = suma_nulos(df[Ocupacion])
				Completitud_densidad_IsEstudio_Receptivos	densidad = suma_nulos(df[Estudio])
				Completitud_densidad_IdDestinoLocalidad_Receptivos	densidad = suma_nulos(df[IdDestinoLocalidad])
				Completitud_densidad_Localidad_Receptivos	densidad = suma_nulos(df[Localidad])
				Completitud_densidad_IdDepartamentoDestino_Receptivos	densidad = suma_nulos(df[IdDepartamentoDestino])
				Completitud_densidad_Departamento_Receptivos	densidad = suma_nulos(df[Departamento])
				Completitud_densidad_IdOtroDepartamento_Receptivos	densidad = suma_nulos(df[IdOtroDepartamento])
				Completitud_densidad_Otro_Departamento_Receptivos	densidad = suma_nulos(df["Otro Departamento"])
				Completitud_densidad_IdOtraLocalidad_Receptivos	densidad = suma_nulos(df[IdOtraLocalidad)
				Completitud_densidad_Otra_Localidad_Receptivos	densidad = suma_nulos(df["Otra Localidad"])
				Completitud_densidad_IdAlojamiento_Receptivos	densidad = suma_nulos(df[IdAlojamiento])
				Completitud_densidad_Alojamiento_Receptivos	densidad = suma_nulos(df[Alojamiento])
				Completitud_densidad_IdTranspLocal_Receptivos	densidad = suma_nulos(df[IdTranspLocal])
				Completitud_densidad_TransporteLocal_Receptivos	densidad = suma_nulos(df[TransporteLocal])
				Completitud_densidad_IdEgresos_Receptivos	densidad = suma_nulos(df[IdEgresos])
				Completitud_densidad_Lugar_Egreso_Receptivos	densidad = suma_nulos(df["Lugar Egreso"])
				Completitud_densidad_Transporte_Internacional_Egreso_Receptivos	densidad = suma_nulos(df["Transporte Internacional de Egreso"])
				Completitud_densidad_IdTranspEgreso_Receptivos	densidad = suma_nulos(df[IdTranspEgreso])
				Completitud_densidad_IdDestino_Receptivos	densidad = suma_nulos(df[IdDestino])
				Completitud_densidad_Destino_Receptivos	densidad = suma_nulos(df[Destino])
				Completitud_densidad_Estadia_Receptivos	densidad = suma_nulos(df[Estadia])
				Completitud_densidad_Gente_Receptivos	densidad = suma_nulos(df[Gente])
				Completitud_densidad_GastoTotal_Receptivos	densidad = suma_nulos(df[GastoTotal])
				Completitud_densidad_GastoAlojamiento_Receptivos	densidad = suma_nulos(df[GastoAlojamiento])
				Completitud_densidad_GastoAlimentacion_Receptivos	densidad = suma_nulos(df[GastoAlimentacion])
				Completitud_densidad_GastoTransporte_Receptivos	densidad = suma_nulos(df[GastoTransporte])
				Completitud_densidad_GastoCultural_Receptivos	densidad = suma_nulos(df[GastoCultural])
				Completitud_densidad_GastoTours_Receptivos	densidad = suma_nulos(df[GastoTours])
				Completitud_densidad_GastoCompras_Receptivos	densidad = suma_nulos(df[GastoCompras])
				Completitud_densidad_GastoOtros_Receptivos	densidad = suma_nulos(df[GastoOtros])
				Completitud_densidad_Coef_Receptivos	densidad = suma_nulos(df[Coef])
				Completitud_densidad_CoefTot_Receptivos	densidad = suma_nulos(df[CoefTot])

Frescura	Actualidad	ratio_actualidad	Archivo	ratio_actualidad_receptivos	frescura = 1 - (t2-t0) / Δt0

		Consistencia_Intra_Relacion_json	Valor	Consistencia_Intra_Relacion_IdFecIng_receptivos	if validarDistintosFecIng(IdFecIng) > 1:
					consistencia = 0
					else:
					consistencia = 1

				Consistencia_Intra_Relacion_Fechas_Estadia_Receptivos	try:
					if FechaEgreso - FechaIngreso != Estadia
					consistencia = 0
					else:
					consistencia = 1
					except:
					consistencia = 1

				Consistencia_Intra_Relacion_lugar_transporte_receptivos_ingreso	try:
					if (`Lugar Ingreso` == 'Aeropuerto de Carrasco' or 'Lugar Ingreso' == 'Aeropuerto de Punta del Este') and `Transporte Internacional de Ingreso` != 'Aereo'
					consistencia = 0
					elif `Lugar Ingreso` == 'Puerto de montevideo' and `Transporte Internacional de Ingreso` != 'Maritimo - Fluvial'
					consistencia = 0
					else:
					consistencia = 1
					except:
					consistencia = 1

Consistencia	Consistencia Intra Relación	Consistencia_Intra_Relacion_conj_valores_json	Conjunto de valores	Consistencia_Intra_Relacion_lugar_transporte_receptivos_egreso	try:
					if (`Lugar Egreso` == 'Aeropuerto de Carrasco' or 'Lugar Egreso == 'Aeropuerto de Punta del Este') and `Transporte Internacional de Egreso` != 'Aereo'
					consistencia = 0
					elif `Lugar Egreso` == 'Puerto de montevideo' and `Transporte Internacional de Egreso` != 'Maritimo - Fluvial'
					consistencia = 0
					else:
					consistencia = 1
					except:
					consistencia = 1

				Consistencia_Intra_Relacion_Otro_Departamento_receptivos	if (select count(distinct(`Otro Departamento`))
					from receptivos
					where IdOtroDepartamento = id) > 1:
					consistencia = 0
					else:
					consistencia = 1

				Consistencia_Intra_Relacion_Otra_Localidad	if (select count(distinct(`Otra Localidad`))
					from receptivos
					where IdOtraLocalidad = id) > 1:
					consistencia = 0
					else:
					consistencia = 1

Unicidad	No Duplicación	Unicidad_no_duplicacion_documento_json	Documento	Unicidad_no_duplicacion_documento_receptivos	unicidad = duplicados_row(indice)['no_duplicacion']
	No Contradicción	Unicidad_no_contradiccion_documento_json	Documento	Unicidad_no_contradiccion_documento_receptivos	unicidad = duplicados_row(indice)['no_contradiccion']

Agregaciones de medidas

1. EXACTITUD

Nombre	Métrica	Fórmula	Granularidad	Descripción
ratio_exactitud_sintactica_col	Exact_sintactica_bool	(1/n) Σexactitud	Columna	Suma de valores de medidas de exactitud sintáctica dividido cantidad de valores de la columna
ratio_precision_col	Exact_precision	(1/n) Σprecision	Columna	Suma de valores de medidas de precisión dividido cantidad de valores de la columna
ratio_exactitud_semántica_col	Exact_semantica_bool	(1/n) Σexactitud	Columna	Suma de valores de medidas de exactitud semántica dividido cantidad de valores de la columna

2. COMPLETITUD

Nombre	Métrica	Fórmula	Granularidad	Descripción

ratio_densidad_tabla	Completitud_densidad	(1/n) Σratio_densidad_columna	Tabla	Promedio de ratios de densidad de las columnas

3. FRESCURA

Dado que se define una métrica de frescura únicamente para la tabla de emisivos, no se realizan agregaciones para esta dimensión.

4. CONSISTENCIA

Nombre	Métrica	Fórmula	Granularidad	Descripción
ratio_consist_columna	Consistencia_Intra_Relacion	(1/n) Σconsistencia_intra_relacion	Columna	Suma de valores de medidas de consistencia de celdas dividido cantidad de valores de la columna
ratio_consist_cols_emisivos	Consistencia_Intra_Relacion_conj_celdas	(1/n) Σ(**0.3***Consistencia_Intra_Relacion_Fechas_Estadia + **0.5***Consistencia_Intra_Relacion_Gastos + **0.1***Consistencia_Intra_Relacion_lugar_transporte_salida + **0.1***Consistencia_Intra_Relacion_lugar_transporte_salida)	Grupo de columnas	Suma de promedios ponderados de reglas de consistencia intra-relación de cada registro dividido el total de registros. Se establecen los pesos indicados en la fórmula en función de la importancia relativa que se le da a la consistencia entre los distintos atributos.
ratio_consist_tuplas_receptivos	Consistencia_Intra_Relacion_conj_celdas	(1/n) Σ(**0.8***Consistencia_Intra_Relacion_Gastos_Receptivos + **0.1***Consistencia_Intra_Relacion_lugar_transporte_receptivos_egreso +	Grupo de columnas	Suma de promedios ponderados de reglas de consistencia intra-relación de cada registro dividido el total de registros. Se establecen los pesos indicados en la fórmula en función de la importancia relativa que se le da a la consistencia entre los distintos atributos.

		0.1*Consistencia_Intra_Relacion_lugar_transporte_receptivos_ingreso)		
ratio_int_dom_columna	Consistencia_Int_Dominio	(1/n) ΣConsistencia_Int_Dominio_Longitud	Columna	Suma de valores de medidas de integridad de dominio dividido cantidad de valores de la columna

5. UNICIDAD

Nombre	Métrica	Fórmula	Granularidad	Descripción
ratio_no_duplicados	Unicidad_no_duplicacion_tupla	(1/n) ΣUnicidad_no_duplicacion_tupla	Tabla	Porcentaje de datos que no están duplicados de forma exacta.
ratio_no_contradicciones	Unicidad_no_contradiccion_tupla	(1/n) ΣUnicidad_no_contradiccion_tupla	Tabla	Porcentaje de datos que no están duplicados con contradicciones

Nota: Si bien se instancian las métricas únicamente para las columnas con valores nulos, para esta agregación se debe instanciar la métrica para todas las columnas, de manera de tener un valor representativo de toda la tabla.

Combinaciones de medidas

Se plantean las siguientes combinaciones entre distintas métricas o agregaciones asociadas a una misma dimensión.

1. EXACTITUD

Una vez calculadas las agregaciones “ratio_exactitud_sintactica_col” y “ratio_exactitud_sintactica_col”, se plantea realizar un promedio ponderado de ambas de manera de obtener una medida de exactitud más general para la columna en la cual fue realizada la agregación. Los pesos que se le atribuyen a cada agregación dependen de la columna agregada. Dado que únicamente se plantearon métricas de exactitud sintáctica y semántica en una misma columna para el archivo de operadores, se presenta en la tabla siguiente los pesos relativos de cada agregación para cada columna.

Atributo	Coeficientes	
	ratio_exactitud_sintactica_col	ratio_exactitud_semantica_col
Telefono	0.30	0.70
Web	0.40	0.60
Email	0.40	0.60

2. UNICIDAD

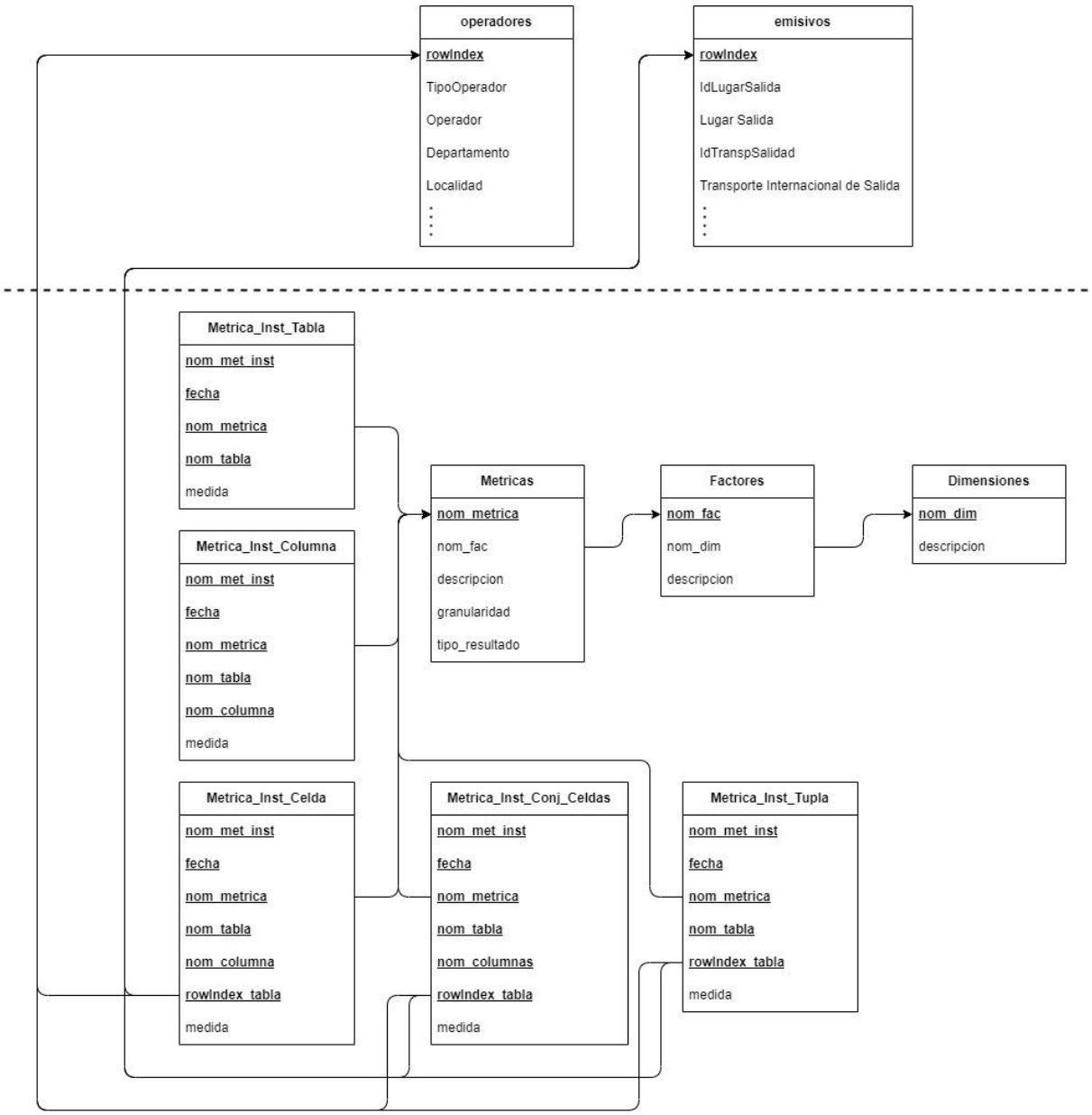
Una vez calculadas las agregaciones “ratio_no_duplicados” y “ratio_no_contradicciones”, se plantea realizar un promedio ponderado de ambas de manera de obtener una medida de unicidad más general para toda la tabla. Dado que se entiende que es un duplicado no contradictorio es significativamente menos importante que un duplicado contradictorio, se asignan pesos relativos a ambas medidas que reflejan esta condición. De esta manera, la combinación planteada toma la siguiente expresión:

ratio_unicidad_tabla = 0.80*ratio_no_duplicados + 0.20*ratio_no_contradicciones

Planificación de la medición

Metadatos de Calidad - Emisivos y Operadores

Para los datos de operadores.csv y emisivos.csv definimos un modelo para almacenar los metadatos de calidad los cuales nos brindará todo el soporte para la ejecución y almacenado de las métricas, instancias y las medidas obtenidas.
Al leer los archivos con python y pandas decidimos utilizar el rowindex como el identificador único de cada registro. Esto es debido a que en los dos archivos no encontramos una clave que determinará un solo registro o porque incluso agrupando todos los registros tenemos casos con registros todos duplicados. Esto nos va a permitir para los casos donde necesitamos evaluar métricas con granularidad Celda, Grupo de Celdas o Tuplas poder identificar el registro inequívocamente.



Nota: La columna “nom_columnas” de la tabla de “Metrica_Inst_Conj_Celdas” corresponde a la concatenación de los nombres de las columnas para las cuales se toma el conjunto de celdas.

Ejemplo de datos generados en la BD de metadatos de calidad

Dimensiones	
nom_dim	descripcion
Exactitud	Determinar exactitud y correctitud de los datos
Completitud	Completitud indica si el dato contiene toda la

	información de interés

Factores		
nom_fac	nom_dim	descripcion
Exactitud sintáctica	Exactitud	Determinar si los datos tienen errores sintácticos o de formato
Exactitud semántica	Exactitud	Determinar la correctitud semántica de los datos
Cobertura	Completitud	Determinar qué cobertura tengo en mis datos frente a la realidad plateada
Densidad	Completitud	Determinar cuál es el grado de información faltante para mis datos

Metricas				
nom_metrica	nom_fac	descripcion	granularidad	tipo_resultado
Exact_sintactica_bool	Exactitud sintáctica	Evalúa la correctitud sintáctica de una celda	Celda	{0,1}
Exact_semantica_gpo_celdas_bool	Exactitud semántica	Evalúa la correctitud semántica de una celda	Conjunto de Celdas	{0,1}
Completitud_cobertura	Cobertura	Mide si mi tabla cubre todos los registros que debería cubrir.	Tabla	[0...1]
Completitud_densidad	Densidad	Calcula el ratio de valores faltantes en una columna con respecto a la cantidad total de valores	Columna	[0...1]

Metrica_Inst_Tabla				
nom_met_inst	**fecha**	**nom_metrica**	**nom_tabla**	medida
Completitud_cobertura_emisivos	2023/06/24	Completitud_cobertura	emisivos	0,0045
Completitud_cobertura_receptivos	2023/06/24	Completitud_co	receptivos	0,0052

		bertura		

Metrica_Inst_Columna					
nom_met_inst	**fecha**	**nom_metrica**	**nom_tabla**	**nom_columna**	medida
Completitud_densidad_Lugar Salida_emisivos	2023/06/24	Completitud_densidad	emisivos	‘Lugar Salida’	0,09
Completitud_densidad_Tipo_Operador_Ope	2023/06/24	Completitud_densidad	operadores	TipoOperador	1

Metrica_Inst_Celda						
nom_met_inst	**fecha**	**nom_metrica**	**nom_tabla**	**nom_columna**	**rowIndex_celda**	medida
Exact_semantica_bool_Web_ope	2023/06/24	Exact_semantica_bool	operadores	Web	2464	0
Exact_semantica_bool_Web_ope	2023/06/24	Exact_semantica_bool	operadores	Web	2489	1

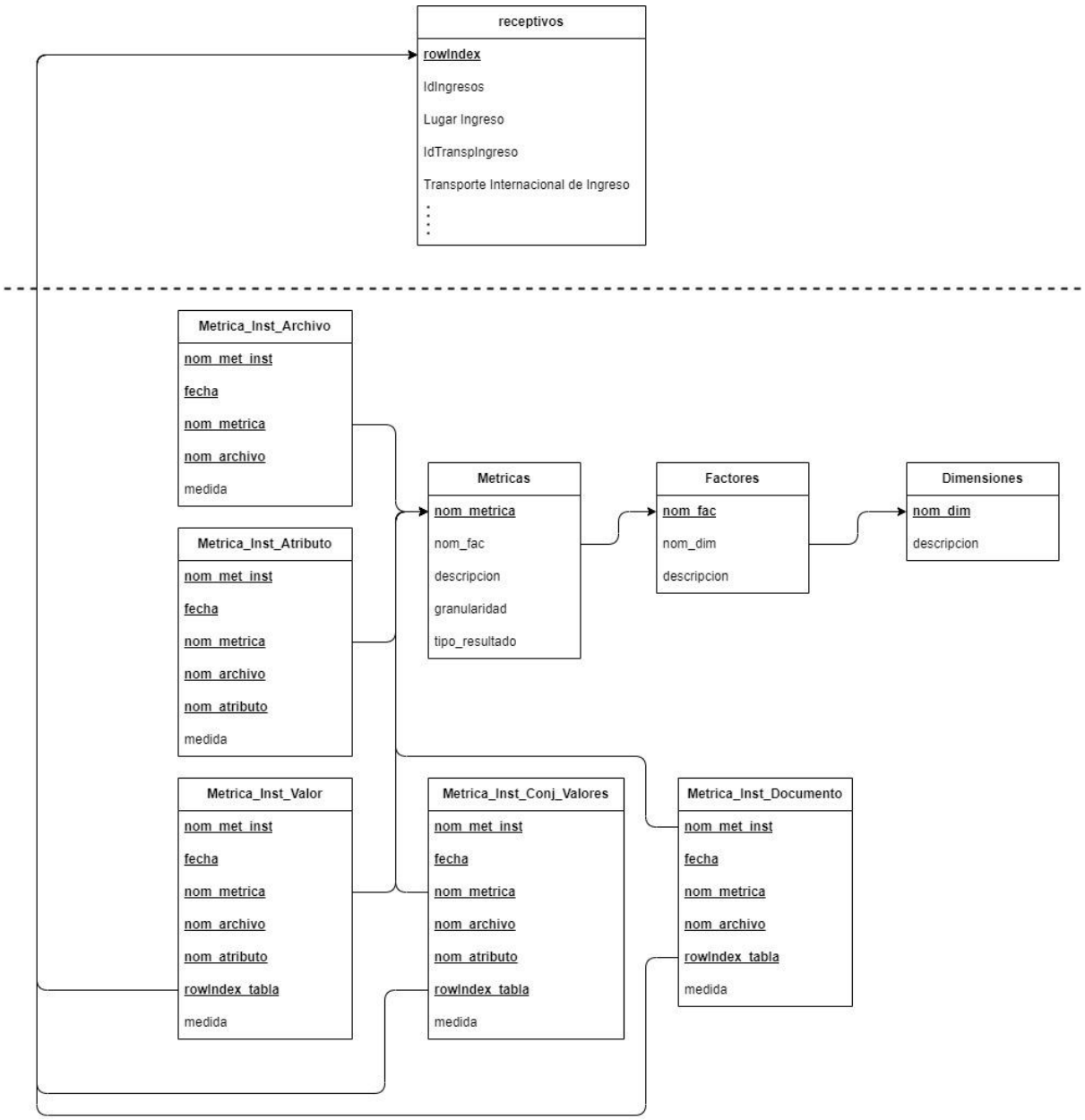
Metrica_Inst_Conj_Celda						
nom_met_inst	**fecha**	**nom_metrica**	**nom_tabla**	**nom_columnas**	**rowIndex_celda**	medida
Consistencia_Intra_Relacion_lugar_transporte_emisivos_salida	2023/06/24	Consistencia_Intra_Relacion_conj_celdas	emisivos	'Lugar Salida','Transporte Internacional de Salida'	20471	1
Consisten	2023/06/2	Consisten	emisivos	‘FechaEntr	456	1

cia_Intra_Relacion_Fechas_Estadia_Emisivos	4	cia_Intra_Relacion_conj_celdas		ada','FechaSalida','Estadia'		

Metrica_Inst_Tupla					
nom_met_inst	**fecha**	**nom_metrica**	**nom_tabla**	**rowIndex_celda**	medida
Unicidad_no_duplicacion_tupla_operadores	2023/06/24	Unicidad_no_duplicacion_tupla	operadores	1491	0
Unicidad_no_duplicacion_tupla_receptivos	2023/06/24	Unicidad_no_duplicacion_tupla	receptivos	819	0

Metadatos de Calidad - Receptivos

Si bien las métricas para este archivo fueron planteadas en base a granularidades definidas para el formato JSON, se optó por un esquema de metadatos relacional. Para ello, se prevé importar el archivo JSON como un dataframe de pandas, pudiendo así representarlo como una tabla en la cual cada fila corresponde a un documento, cada columna a un atributo y cada celda a un valor del archivo JSON original (ingresando valores nulos en las columnas correspondientes a atributos no definidos en el archivo original para la fila en cuestión). De esta manera, es posible establecer el mismo mecanismo de identificación de fila utilizado para los archivos csv. y utilizar un esquema muy similar al utilizado para almacenamiento de metadatos de calidad de los archivos csv.



La finalidad de los modelos planteados tanto para los archivos csv como para el json es mantener el resultado de las ejecuciones y resultados de las mediciones de calidad en una BD relacional, tipo MySql, SQLServer, o similar.

Medición de Calidad de Datos dependiente del Contexto

1. Definición de Contexto

Tipo de Componente de Contexto	Etapa: ST1

Dominio	Turismo
Características de Usuario	● U1 - Gerentes ● U2 - Administrativos ● U3 - Usuarios de los datos
Tareas	● T1 - Generación de reportes de los datos ● T2 - Registro de datos en la base de datos ● T3 - Consultas de los datos
Reglas de Negocio	● RN1 - Todas las fechas deben ser distintas de nulo. ● RN2 - Los teléfonos no pueden ser nulos ni vacíos. ● RN3 - Los nombres de ciudades y/o departamentos no pueden ser nulos ni vacíos. ● RN4 - Los días de estadía deben ser de al menos 1. ● RN5 - Los datos registrados no deben contener caracteres especiales. ● RN6 - Ninguno de los datos registrados puede estar abreviado. ● RN7 - Todos los operadores deben registrar valores de latitud y longitud. ● RN8 - Moneda = U$S, es decir, los gastos representan valores en dólares. ● RN9 - Si el alojamiento es una vivienda familiar, entonces el costo es 0. ● RN10 - La localidad de los operadores determina el departamento. ● RN11 - Los estudios de las personas solo pueden ser “primaria”, “secundaria” o “terciaria”.
Req. de Sistema	● RS1 - Sistema debe dar resultados de consultas en un tiempo máximo de 2 segundos.
Req. de Calidad de Datos	● RQ1 - Formatos de fechas AAAA/MM/DD ● RQ2 - En archivo emisivos Fecha Salida tiene que ser menor a Fecha Entrada. ● RQ3 - En archivo receptivos Fecha Ingreso tiene que ser menor a Fecha Egreso ● RQ4 - Al menos 10 registros de turismo emisivo por departamento. ● RQ5 - Al menos 80% de registros con valores en campo “operador” del archivo operadores turísticos. ● RQ6 - Cada operador debe presentar al menos el 50% de sus datos no nulos ● RQ7 - Al menos el 90% de los mails ingresados en operadores debe ser válido.

	• RQ8 - Los gastos de turistas deben ser representados con al menos 2 decimales después de la coma.
Necesidades de Filtrado	• - - -
Metadatos	• M1 - metadatos_parte2.xlsx
Metadatos de Calidad de Datos	• - - -
Otros Datos	• OD1 - agenciasDeViaje.csv • OD2 - agenciasDeViaje-metadatos.xlsx

2. Cambios al modelo de calidad presentado en la parte inicial

RN1, RN2 y RN3 entendemos que son reglas de negocio para las cuales ya se cuentan con mediciones para revisar el cumplimiento de las mismas.

Primer cambio

Si bien se verifica la dependencia funcional entre las fechas de entrada y salida y la duración de la estadía de los turistas, RN4 resulta en una medición que no tenemos elaborada y deberíamos incluirla para controlar dicho requerimiento. Esta nueva métrica es instanciada a continuación

Métrica instanciada	
Exact_semantica_bool_Estadia_emisivos	
Métrica	Exact_semantica_bool
Datos	emisivos.'Estadia'
Método Medición	if Estadia >= 1 exactitud = 1 else: exactitud = 0

Métrica instanciada	
Exact_semantica_bool_Estadia_receptivos	
Métrica	Exact_semantica_bool
Datos	receptivos.'Estadia'

Método Medición	if Estadia >= 1 exactitud = 1 else: exactitud = 0

Segundo cambio

Debido a lo observado en la etapa de data profiling, originalmente se planteó una métrica de exactitud sintáctica para el campo de FechaSalida del archivo de emisivos y otra para los campos FechaIngreso y FechaEgreso del archivo de receptivos, que verificaban que las fechas estén ingresadas en formato AAAA-MM-DD, ya que éste era el formato de fecha predominante. Sin embargo, se deben realizar dos cambios a estas métricas para poder verificar el cumplimiento del requerimiento RQ1. Estos cambios son los siguientes:

- Modificar la función utilizada en el método de medición de manera de chequear que el formato correcto sea MM/DD/AAAA y no AAAA-MM-DD
- Instanciar la métrica para el campo de FechaEntrada del archivo de emisivos, ya que es el único campo de fecha para el cual no se había instanciado la métrica original (debido a que todos sus valores contaban con el formato considerado como correcto originalmente). A continuación se presentan las métricas instanciadas para los cuatro campos en cuestión y se incluye la función modificada.

Métrica instanciada	
Exact_sintactica_bool_FechaSalida	
Métrica	Exact_sintactica_bool
Datos	emisivos.'FechaSalida'
Método Medición	if check_date_format(FechaSalida) == True exactitud = 1 else: exactitud = 0

Métrica instanciada	
Exact_sintactica_bool_FechaEntrada	
Métrica	Exact_sintactica_bool
Datos	emisivos.'FechaEntrada'

Método Medición	if check_date_format(FechaSalida) == True exactitud = 1 else: exactitud = 0

Métrica instanciada	
Exact_sintactica_bool_FechaIngreso_receptivos	
Métrica	Exact_sintactica_bool_json
Datos	receptivos.'FechaIngreso'
Método Medición	if check_date_format(FechaIngreso) == True exactitud = 1 else: exactitud = 0

Métrica instanciada	
Exact_sintactica_bool_FechaEgreso_receptivos	
Métrica	Exact_sintactica_bool_json
Datos	receptivos.'FechaEgreso'
Método Medición	if check_date_format(FechaEgreso) == True exactitud = 1 else: exactitud = 0

La función modificada es la siguiente:

```
def check_date_format(string):
    pattern = r"\d{2}/\d{2}/\d{4}"
    match = re.fullmatch(pattern, string)
    if match:
        return True
    else:
        return False
```

Tercer cambio

RN5 es un requerimiento que no tuvimos presente, quizás justamente por no haber contado con el contexto suficiente, y creemos que es una de las mediciones que podemos integrar. Para ello, se podría instanciar la métrica de exactitud sintáctica Exact_sintactica_bool con un método de medición alternativo que verifique si el valor contiene caracteres especiales. Si bien esta métrica podría ser instanciada para la gran mayoría de los campos ya tratados, a continuación se ejemplifica esta métrica para uno de los campos para el cual sería aplicable:

Métrica instanciada	
Exact_sintactica_bool_direccion	
Métrica	Exact_sintactica_bool
Datos	operadores.Direccion

Método Medición	if check_no_special_characters(Direccion) == True:
	exactitud = 1
	else
	exactitud = 0

La función utilizada es la siguiente

```
def check_no_special_characters(input_string):
    pattern = r'^[a-zA-Z0-9]+$'
    match = re.match(pattern, str(input_string))
    return match is not None
```

Cuestionario

Considere el siguiente cuestionario:
Dado que las respuestas serán el insumo de un análisis estadístico (que no está relacionado con la evaluación de los estudiantes), se solicita que las respuestas sean pensadas de forma individual.

A. **¿De los componentes del contexto visto en clase, cuál(es) considera más relevantes? Justifique brevemente.**

   Federico - Creo que de los componentes de contexto que vimos en clase el más relevante es el Requerimiento de Negocio, porque entiendo que son una buena base para que la calidad de los datos se cumpla y en caso de que no exista tal requerimiento implementado es más factible que se pueda implementar.

Diego - Creo que el componente más importante son las reglas de negocio, ya que estas indican condiciones que deben cumplirse de manera obligatoria. Estas no son negociables y se debe asegurar su cumplimiento siempre, pudiendo incluso obedecer a cuestiones legales, de seguridad o de confidencialidad. Si bien otros requerimientos también pueden ser importantes y hasta pueden implicar grandes mejoras en el uso de los datos, las reglas de negocio probablemente deban tener siempre mayor prioridad

B. ¿Identifica otro(s) componente(s) que le interesaría agregar a la definición del contexto? Cuál(es)? Justifique brevemente.

Federico - Me parece que todos los componentes están bien y cubren todas la necesidades para comenzar a elaborar el modelo de calidad y ejecutar calidad sobre los datos, quizás algo que pueda sumar es tener también información técnica del sistema, capaz pueden ser incluídas en algún componente, pero quizás diferenciarlos pueda ayudar a identificar qué nivel de calidad ya tiene incorporado el sistema/datos.

Diego - Considero que un componente que puede resultar de importancia para la definición del contexto es información sobre el proceso de obtención y recopilación de los datos para los cuáles se realizará el modelo de calidad. Información sobre dicho proceso puede ser útil al momento de evaluar las causas sobre la falta de calidad de los datos, lo que a su vez puede permitir definir acciones preventivas, o de mejora, personalizadas. A su vez, también permitirá identificar limitaciones inherentes al proceso y tratarlas adecuadamente.

C. ¿Considera que al definir un modelo de calidad de datos es importante contar con el contexto de los datos? Justifique brevemente.

Federico - Creo que contar con el contexto ayuda a centrar la calidad de los datos en puntos importantes del negocio sin descuidar otros aspectos que consideremos importantes pero que muchas veces no son visibles o tenidos en cuenta en el contexto.

Diego - Considero que sí es útil porque permite identificar formalmente los aspectos que más van a influir en la calidad de los datos que se tienen para el uso que se le van a dar. Sin contexto, uno debe recurrir a la intuición o a un sobredimensionamiento del modelo de calidad de manera de poder capturar todas las posibles faltas de calidad de los datos, mientras que un contexto permite focalizarse en lo más importante para la situación en cuestión.