

Trabajo practico N°4: Analisis de Estudiantes del IES21

La idea es que lo exploren, que se familiaricen con su contenido, y que en nuestra próxima clase podamos establecer la 1ra de dos "entrevistas" que tendrán conmigo, en las que oficiaré para esta actividad como el "Cliente" de sus Servicios de Análisis de Datos.

Deberán entrevistarme con el objetivo de comprender la naturaleza de mi negocio, y mis intenciones informativas sobre el conjunto de datos que les he presentado...

Objetivos preliminares (pueden surgir otros durante la entrevista):

Presentar gráficamente información que me permita conocer la cantidad de alumnos por año, o por semestre/año; estableciendo curvas comparativas con diferentes formas de desagregamiento: por sexos, por edades, por carreras, etc...

Obtener asimismo información estadística de resumen respecto de las materias que los alumnos cursan, sus notas promedio, desvíos, y muy especialmente los outliers si los hubiera.

Finalmente, el más ambicioso de los objetivos, será el de encontrar alguna suerte de "correlación" (si es que la hubiere) entre el desempeño académico durante el cursado de los alumnos, considerando variables como cantidad de materias en cada periodo y calificaciones obtenidas, y su probabilidad de deserción.

Sobre los datos

La estructura de los datos es suficientemente explícita; sólo caben algunas aclaraciones importantes:

Las calificaciones son notas comprendidas entre 0 (cero) y 10 (diez). Encontrarán en las columnas correspondientes otros valores: 11 = Ausente
Null = Nota no cargada/No procedente

Una cuestión de jerga: las carreras se componen de "cuatrimestres", y los años de "semestres". Así, por ejemplo, en el plan de estudio de su carrera tienen las materias de 1er cuatrimestre, las de 2do, 3ro, 4to, 5to y 6to. En cambio los años tienen 2 mitades: 1er semestre y 2do semestre. No es que en un año quepan 3 cuatrimestres!

Cada alumno es identificado en el sistema mediante su DNI, pero por cuestiones de confidencialidad, ese dato ha sido reemplazado por un valor cifrado de manera uniforme: Mismo alumno = Misma clave. Dos alumnos diferentes = Dos claves diferentes. Respecto de la tabla de Alu_Materias, casi nada que aclarar:

A lo largo del tiempo, algunas carreras van modificando su plan de estudios, por lo que no puede inferirse que un alumno se haya recibido si tiene cursadas todas las materias existentes en su carrera ya que puede que algunas de ellas pertenezcan a un plan de estudios diferente del que él o ella cursó.

A los efectos del análisis, la condición de "Egresado" no es relevante; sí lo es que el alumno haya llegado a cursar el último cuatrimestre de su carrera.

Las calificaciones pueden distinguirse por materia en: Aprobada (superior o igual a 5), Reprobada (menor a 5).

Como les decía al principio, la idea es analizar -aunque sea someramente- el DataSet que les entrego para poder formular preguntas pertinentes en nuestra entrevista. Si vienen con algunas gráficas ya planteadas, mejor: podremos realizar una etapa fundamental en este tipo de trabajo que es la "validación".

En este espacio vamos a cargar las bibliotecas que vamos a usar.

```
In [1]: 1 import pandas as pd
        2 import numpy as np
        3 import plotly.express as px
        4 from plotly.offline import plot
```

Cargada de los datos

Primero cargamos el primer archivo con los datos de los alumnos por carreras.

```
In [2]: 1 #Cargamos el df alu_carrera
        2 df_alu_carrera = pd.read_csv("data/Alu_Carrera.csv", sep = ';')
        3 df_alu_carrera.head(2)
```

Out[2]:

	Id_Alumno	Id_Carrera	Semestre_Ingreso	Año_Ingreso	Id_Ctro_Ext	Id_Modalidad	Fecha_Ingreso	Activo	Sexo	F
0	0x12DA11908B130262F20F714810A49680	6	2	2010	CO	P	2010-07-30 00:00:00	S	M	
1	0xFCDE4E2EACC469337D296ECE6B5083C6	17	1	2006	CO	P	2006-02-24 00:00:00	S	M	

```
In [3]: 1 df_alu_carrera.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6227 entries, 0 to 6226
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Id_Alumno              6227 non-null   object 
1   Id_Carrera             6227 non-null   int64  
2   Semestre_Ingreso       6227 non-null   int64  
3   Año_Ingreso            6227 non-null   int64  
4   Id_Ctro_Ext            6227 non-null   object 
5   Id_Modalidad           6227 non-null   object 
6   Fecha_Ingreso          6227 non-null   object 
7   Activo                 6227 non-null   object 
8   Sexo                  6227 non-null   object 
9   Fecha_Nac              6227 non-null   object 
dtypes: int64(3), object(7)
memory usage: 486.6+ KB
```

Cargamos el archivo con los datos de las carreras y lo relacionamos al archivo con los datos de alumnos por carreras, generando un archivo con la información de los 2.

```
In [4]: 1 #Cargamos el df carreras.
2 df_carreras = pd.read_csv("data/Carreras.csv", sep=';')
3 df_carreras
```

Out[4]:

	Id_Carrera	Nombre_Largo	Nombre_Corto	Abreviatura	Cuatrimestres
0	9	TECNICO SUPERIOR EN ADMINISTRACION AGROPECUARIA	AGRONEGOCIOS	AA	6
1	3	TECNICO SUPERIOR EN PUBLICIDAD	PUBLICIDAD	PUB	6
2	12	ANALISTA EN GESTION DE VENTAS	VENTAS	VTA	4
3	6	TECNICO SUPERIOR EN RELACIONES PUBLICAS E INST...	RELACIONES PÚBLICAS	RRPP	6
4	7	ADMINISTRADOR DE RECURSOS HUMANOS	RECURSOS HUMANOS	RRHH	6
5	1	ANALISTA DE SISTEMAS DE COMPUTACION	ANALISTA DE SISTEMAS	INF	6
6	18	TECNICO SUPERIOR EN LOGISTICA	LOGÍSTICA	LOG	6
7	10	TECNICO SUPERIOR EN TURISMO	TURISMO	TUR	6
8	41	AGENTE DE TRANSPORTE ADUANERO Y DESPACHANTE DE...	AGENTE DE TRANSP. ADUANERO Y DESP. DE ADUANA	ATA	4
9	4	TECNICO SUPERIOR EN ADMINISTRACION DE EMPRESAS	ADMINISTRACIÓN DE EMPRESAS	ADMIN	6
10	13	TECNICO SUPERIOR EN MULTIMEDIOS	MULTIMEDIOS	MM	6
11	5	DISEÑO GRAFICO Y PUBLICITARIO	DISEÑO GRÁFICO	DG	6
12	59	TÉCNICO SUPERIOR EN SERVICIOS DE PAISAJISMO	SERVICIOS DE PAISAJISMO	SERPAI	2
13	39	ADMINISTRADOR DE REDES	ADMINISTRADOR DE REDES	AR	4
14	16	TECNICO SUPERIOR EN GESTION JURIDICA	GESTIÓN JURÍDICA	GJ	6
15	2	ANAL.DE MERC.Y ESTRAT. DE COMERCIALIZACION	MARKETING	MKT	6
16	17	TECNICO SUPERIOR EN GESTION CONTABLE Y FINANCIERA	GESTIÓN CONTABLE Y FINANCIERA	GCF	6
17	60	TÉC. SUP. EN DES. DE SIMULACIONES VIRTUALES Y ...	VIDEOJUEGOS	SVIRT	6
18	40	TÉCNICO EN SERVICIOS AGROPECUARIOS	TÉCNICO EN SERVICIOS AGROPECUARIOS	TSAGR	4
19	11	ASISTENTE GERENCIAL TRILINGÜE	TRILINGÜE	TRI	4

	Id_Carrera		Nombre_Largo	Nombre_Corto	Abreviatura	Cuatrimestres
20	14		TECNICATURA SUPERIOR EN GESTION AMBIENTAL	GESTIÓN AMBIENTAL	GA	6
21	8		ANALISTA EN COMERCIO INTERNACIONAL	COMERCIO INTERNACIONAL	CI	6

```
In [5]: 1 # Formamos un nuevo df de alu_carreras pero agregando las columnas de nombre corto y cantidad de cuatrimestres.
2 df_alu_carrera2 = pd.DataFrame()
3 df_alu_carrera2 = pd.merge(df_alu_carrera,df_carreras[['Id_Carrera','Nombre_Corto','Cuatrimestres']],
4                             on = ['Id_Carrera'] )
5 #Cambio el nombre de la columna cuatrimestre por una que marque que es la cantidad de cuatrimestres que la componen
6 df_alu_carrera2 = df_alu_carrera2.rename(columns={'Cuatrimestres':'CantCuatrimestres'})
7 df_alu_carrera2.head()
```

Out[5]:

		Id_Alumno	Id_Carrera	Semestre_Ingreso	Año_Ingreso	Id_Ctro_Ext	Id_Modalidad	Fecha_Ingreso	Activo	Sexo	Fi
0		0x12DA11908B130262F20F714810A49680	6	2	2010	CO	P	2010-07-30 00:00:00	S	M	
1		0xA3CF8BA9A8EB103AE6F25BB33B79BB71	6	2	2001	CO	P	2001-08-06 00:00:00	S	F	
2		0x9530CE8299C7D0B9D1FE0AB99CAF3810	6	1	2001	CO	P	2001-03-02 00:00:00	S	F	
3		0x9E60CEA4305B11EAEB97925533D0A266	6	1	2001	CO	P	2001-04-03 00:00:00	S	F	
4		0x590850F0100DAEBEF0711D047DB5593E	6	1	2004	CO	P	2004-02-24 00:00:00	S	F	1

Luego cargamos el archivo de alumnos por materias y lo relacionamos con el archivo anterior que habiamos generado.

In [6]:

```
1 #Cargamos el df de alu_materia
2 df_Alu_Materia = pd.read_csv("data/Alu_Materia.csv", sep = ';')
3 df_Alu_Materia.head()
4
```

Out[6]:

		Id_Alumno	Id_Carrera	Id_Materia	Semestre	Año	Fecha_Inscrip	Ctro_Ext	Id_Modalidad	Id_Comision	Nota_1p	F
0	0x003014EF4D7D79987F42DE8E3080AEFE	6	115	2	2006	2006-08-16 15:02:00	CO	P	7	11.0		
1	0x003014EF4D7D79987F42DE8E3080AEFE	6	121	1	2005	2005-03-10 18:12:00	CO	P	7	11.0		
2	0x003014EF4D7D79987F42DE8E3080AEFE	6	121	2	2005	2005-08-17 18:50:00	CO	P	7	11.0		
3	0x003014EF4D7D79987F42DE8E3080AEFE	6	198	1	2005	2005-03-10 18:12:00	CO	P	7	7.0		
4	0x003014EF4D7D79987F42DE8E3080AEFE	6	374	1	2006	2006-03-06 12:33:00	CO	P	7	6.0		



In [7]:

```
1 #Genero la union entre el df df_alu_carrera2 y alu_materias
2 df_alu_carrera_Alu_Materia = pd.DataFrame()
3 df_alu_carrera_Alu_Materia = pd.merge(df_alu_carrera2, df_Alu_Materia[['Id_Alumno','Id_Carrera','Id_Materia', 'Semestre',
4     'Fecha_Inscrip','Id_Modalidad', 'Id_Comision', 'Nota_1p',
5     'Fecha_1p', 'Nota_2p', 'Fecha_2p', 'Nota_TP', 'Fecha_TP', 'Nota_Recup',
6     'Fecha_Recup', 'Cuatrimestre']], on = ['Id_Alumno','Id_Carrera'])
7 df_alu_carrera_Alu_Materia.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 131919 entries, 0 to 131918
Data columns (total 27 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Id_Alumno              131919 non-null object
1   Id_Carrera             131919 non-null int64
2   Semestre_Ingreso       131919 non-null int64
3   Año_Ingreso            131919 non-null int64
4   Id_Ctro_Ext            131919 non-null object
5   Id_Modalidad_x         131919 non-null object
6   Fecha_Ingreso          131919 non-null object
7   Activo                 131919 non-null object
8   Sexo                   131919 non-null object
9   Fecha_Nac              131919 non-null object
10  Nombre_Corto           131919 non-null object
11  CantCuatrimestres      131919 non-null int64
12  Id_Materia             131919 non-null int64
13  Semestre               131919 non-null int64
14  Año                    131919 non-null int64
15  Fecha_Inscrip          131919 non-null object
16  Id_Modalidad_y         131919 non-null object
17  Id_Comision            131919 non-null int64
18  Nota_1p                129337 non-null float64
19  Fecha_1p               129310 non-null object
20  Nota_2p                110844 non-null float64
21  Fecha_2p               110844 non-null object
22  Nota_TP                130044 non-null float64
23  Fecha_TP               130044 non-null object
24  Nota_Recup             56424 non-null float64
25  Fecha_Recup            56424 non-null object
26  Cuatrimestre           131919 non-null int64
```

```
dtypes: float64(4), int64(9), object(14)
memory usage: 28.2+ MB
```

```
In [8]: 1 df_alu_carrera_Alu_Materia.head()
```

```
Out[8]:
```

	Id_Alumno	Id_Carrera	Semestre_Ingreso	Año_Ingreso	Id_Ctro_Ext	Id_Modalidad_x	Fecha_Ingreso	Activo	Sexo
0	0x12DA11908B130262F20F714810A49680	6	2	2010	CO	P	2010-07-30 00:00:00	S	M
1	0x12DA11908B130262F20F714810A49680	6	2	2010	CO	P	2010-07-30 00:00:00	S	M
2	0x12DA11908B130262F20F714810A49680	6	2	2010	CO	P	2010-07-30 00:00:00	S	M
3	0x12DA11908B130262F20F714810A49680	6	2	2010	CO	P	2010-07-30 00:00:00	S	M
4	0xA3CF8BA9A8EB103AE6F25BB33B79BB71	6	2	2001	CO	P	2001-08-06 00:00:00	S	F

5 rows × 27 columns



Despues, cargamos el archivo con el nombre de las materias y relacionamos con el archivo anterior. Generando un archivo general con los 4 que habiamos cargado.


```
In [9]: 1 #Cargo el df materias y lo uno con df_alu_carrera_Alu_Materia
2 df_Materias = pd.read_csv("data/Materias.csv", sep=';')
3 df_Materias
```

Out[9]:

	Id_Materia	Nombre
0	2	HISTORIA DEL DISEÑO
1	3	INTRODUCCIÓN A LA PUBLICIDAD
2	4	ELEMENTOS DE DISEÑO I
3	5	DISEÑO Y COMUNICACIÓN
4	6	DISEÑOMÁTICA I
...
800	833	ELEGIR-TE: TALLER DE ORIENTACIÓN VOCACIONAL
801	834	EXCEL INTERMEDIO
802	835	DESARROLLO DE "SOFT SKILLS
803	836	Introducción al modelado 3D en 3Ds Max
804	837	COMPETENCIAS DIGITALES PARA PROFESIONALES

805 rows × 2 columns

```
In [10]: 1 df_alu_carr_mate = pd.merge(df_alu_carrera_Alu_Materia,df_Materias[['Id_Materia','Nombre']], on= 'Id_Materia' )
2 df_alu_carr_mate.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 131919 entries, 0 to 131918
Data columns (total 28 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Id_Alumno             131919 non-null object  
1   Id_Carrera            131919 non-null int64   
2   Semestre_Ingreso      131919 non-null int64   
3   Año_Ingreso           131919 non-null int64   
4   Id_Ctro_Ext           131919 non-null object  
5   Id_Modalidad_x        131919 non-null object  
6   Fecha_Ingreso         131919 non-null object  
7   Activo                131919 non-null object  
8   Sexo                 131919 non-null object  
9   Fecha_Nac             131919 non-null object  
10  Nombre_Corto          131919 non-null object  
11  CantCuatrimestres     131919 non-null int64   
12  Id_Materia            131919 non-null int64   
13  Semestre              131919 non-null int64   
14  Año                  131919 non-null int64   
15  Fecha_Inscrip         131919 non-null object  
16  Id_Modalidad_y        131919 non-null object  
17  Id_Comision           131919 non-null int64   
18  Nota_1p              129337 non-null float64  
19  Fecha_1p             129310 non-null object  
20  Nota_2p              110844 non-null float64  
21  Fecha_2p             110844 non-null object  
22  Nota_TP              130044 non-null float64  
23  Fecha_TP             130044 non-null object  
24  Nota_Recup           56424 non-null float64  
25  Fecha_Recup          56424 non-null object  
26  Cuatrimestre         131919 non-null int64   
27  Nombre               131919 non-null object  
dtypes: float64(4), int64(9), object(15)
memory usage: 29.2+ MB
```

Generamos una agrupacion por carreras para saber la cantidad de alumnos que tienen. Luego de eso generamos un grafico donde muestra el

porcentaje de ellos.

```
In [11]: 1 cantidadAlumnosTotal = df_alu_carr_mate.groupby(['Nombre_Corto'],as_index=False)['Id_Alumno'].count()
2 cantidadAlumnosTotal
```

Out[11]:

	Nombre_Corto	Id_Alumno
0	ADMINISTRACIÓN DE EMPRESAS	13384
1	ADMINISTRADOR DE REDES	64
2	AGENTE DE TRANSP. ADUANERO Y DESP. DE ADUANA	22
3	AGRONEGOCIOS	10563
4	ANALISTA DE SISTEMAS	10881
5	COMERCIO INTERNACIONAL	4733
6	DISEÑO GRÁFICO	21445
7	GESTIÓN AMBIENTAL	929
8	GESTIÓN CONTABLE Y FINANCIERA	2543
9	GESTIÓN JURIDICA	347
10	LOGÍSTICA	80
11	MARKETING	14536
12	MULTIMEDIOS	6777
13	PUBLICIDAD	11882
14	RECURSOS HUMANOS	13020
15	RELACIONES PÚBLICAS	13840
16	SERVICIOS DE PAISAJISMO	8
17	TRILINGÜE	24
18	TURISMO	6120
19	TÉCNICO EN SERVICIOS AGROPECUARIOS	12
20	VENTAS	23
21	VIDEOJUEGOS	686

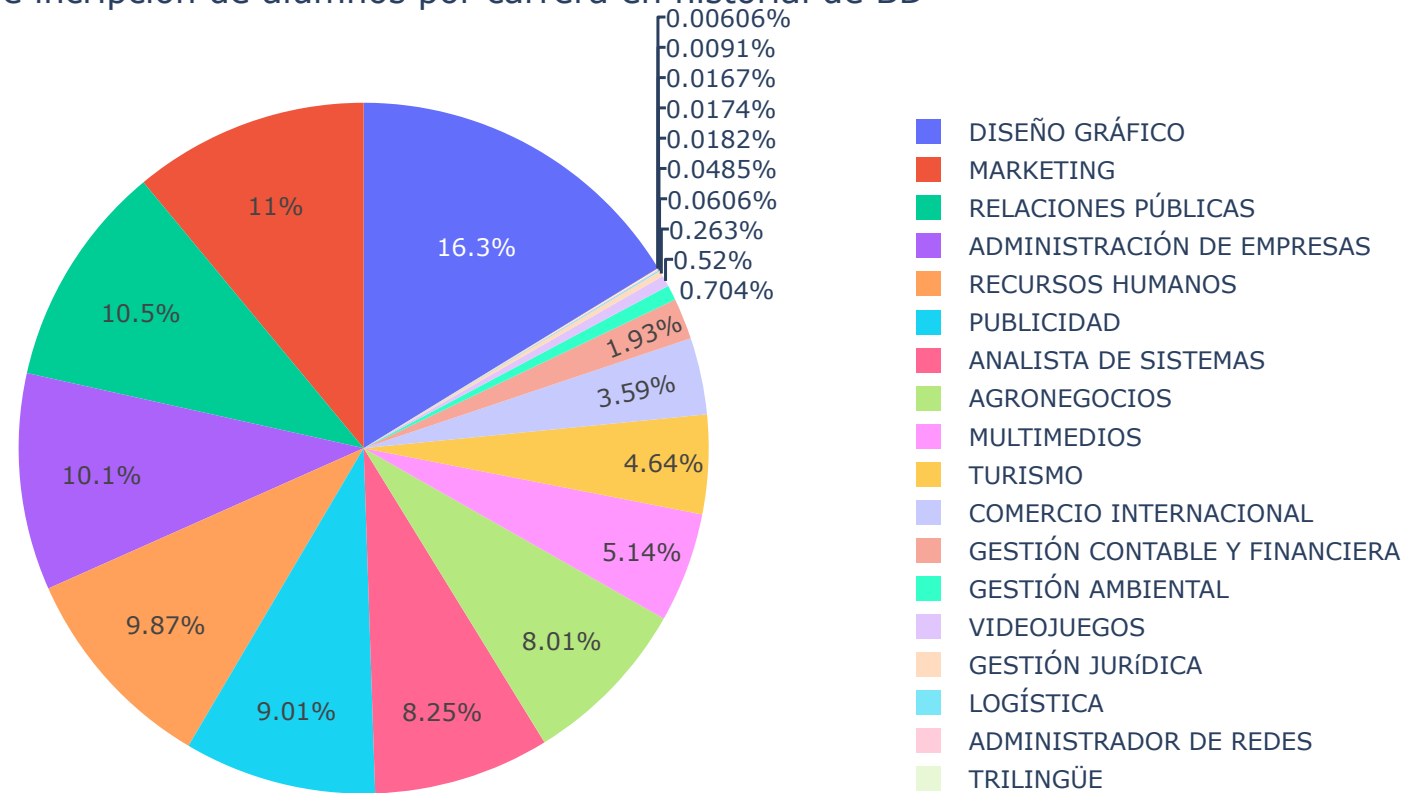
```
In [12]: 1 cant_alu_carr_años = df_alu_carr_mate.groupby(['Id_Carrera', 'Nombre_Corto'], as_index=False)['Id_Alumno'].count()
2 cant_alu_carr_años
```

Out[12]:

	Id_Carrera	Nombre_Corto	Id_Alumno
0	1	ANALISTA DE SISTEMAS	10881
1	2	MARKETING	14536
2	3	PUBLICIDAD	11882
3	4	ADMINISTRACIÓN DE EMPRESAS	13384
4	5	DISEÑO GRÁFICO	21445
5	6	RELACIONES PÚBLICAS	13840
6	7	RECURSOS HUMANOS	13020
7	8	COMERCIO INTERNACIONAL	4733
8	9	AGRONEGOCIOS	10563
9	10	TURISMO	6120
10	11	TRILINGÜE	24
11	12	VENTAS	23
12	13	MULTIMEDIOS	6777
13	14	GESTIÓN AMBIENTAL	929
14	16	GESTIÓN JURÍDICA	347
15	17	GESTIÓN CONTABLE Y FINANCIERA	2543
16	18	LOGÍSTICA	80
17	39	ADMINISTRADOR DE REDES	64
18	40	TÉCNICO EN SERVICIOS AGROPECUARIOS	12
19	41	AGENTE DE TRANSP. ADUANERO Y DESP. DE ADUANA	22
20	59	SERVICIOS DE PAISAJISMO	8
21	60	VIDEOJUEGOS	686

```
In [13]: 1 fig = px.pie(cant_alu_carr_años, values='Id_Alumno', names='Nombre_Corto', title='Porcentaje de inscripcion de alumno
2 fig.show()
```

Porcentaje de inscripcion de alumnos por carrera en historial de BD



Clasificación.

Generamos una clasificación por alumnos para saber cuáles de ellos son los que están promocionados, regulares, libres y los más importantes cuales fueron los que desertaron la carrera.

Para hacer esto lo que hacemos es definir una función, que a través de un pensamiento lógico vaya descartando las opciones que tenemos para cada caso combinando las variables que se nos pueden presentar en dicho procedimiento.

```
In [14]: 1 def ClasificAcadem(p1,p2,tp,rec):
2         #limpieza de datos.
3         if p1 ==11 : p1 = np.nan
4         if p2 ==11 : p2 = np.nan
5         if tp ==11 : tp = np.nan
6         if rec ==11 : rec = np.nan
7
8         #Clasificacion de alumnos.
9         if (p1 >= 7) and (p2 >= 7) and (tp >= 7):
10            return 'P'
11        elif (p1 >= 5) and (p2 >= 5) and (tp >= 5):
12            return 'R'
13        elif (tp >= 5) and (rec >= 5):
14            return 'R'
15        elif (tp == 11) or (np.isnan(tp)):
16            return 'D'
17        elif (rec == 11) or (np.isnan(rec)):
18            return 'D'
19        else:
20            return 'L'
```

Luego de generar el algoritmo que haga la clasificación, lo que hacemos es generar una nueva columna en nuestro DF con el nombre **“Est_mat”** donde se van a ir cargando los estados de cada alumno con respecto a las materias que fue cursando y cual fue su estado final con el que quedo, dependiendo de las notas del semestre, esto es sin contar la nota de finales.

```
In [15]: df_alu_carr_mate['Est_mat'] = df_alu_carr_mate.apply(lambda f: ClasificAcademy
1         (f['Nota_1p'],
2          f['Nota_2p'],
3          f['Nota_TP'],
4          f['Nota_Recup']],axis=1)
5
6
```

Realizo un DF en el cual se muestre una agrupación de todos los estados y se realice un conteo para saber el estado generador de todos los alumnos dentro de IES y los mismos los acomodo por orden de prioridades en los estados.

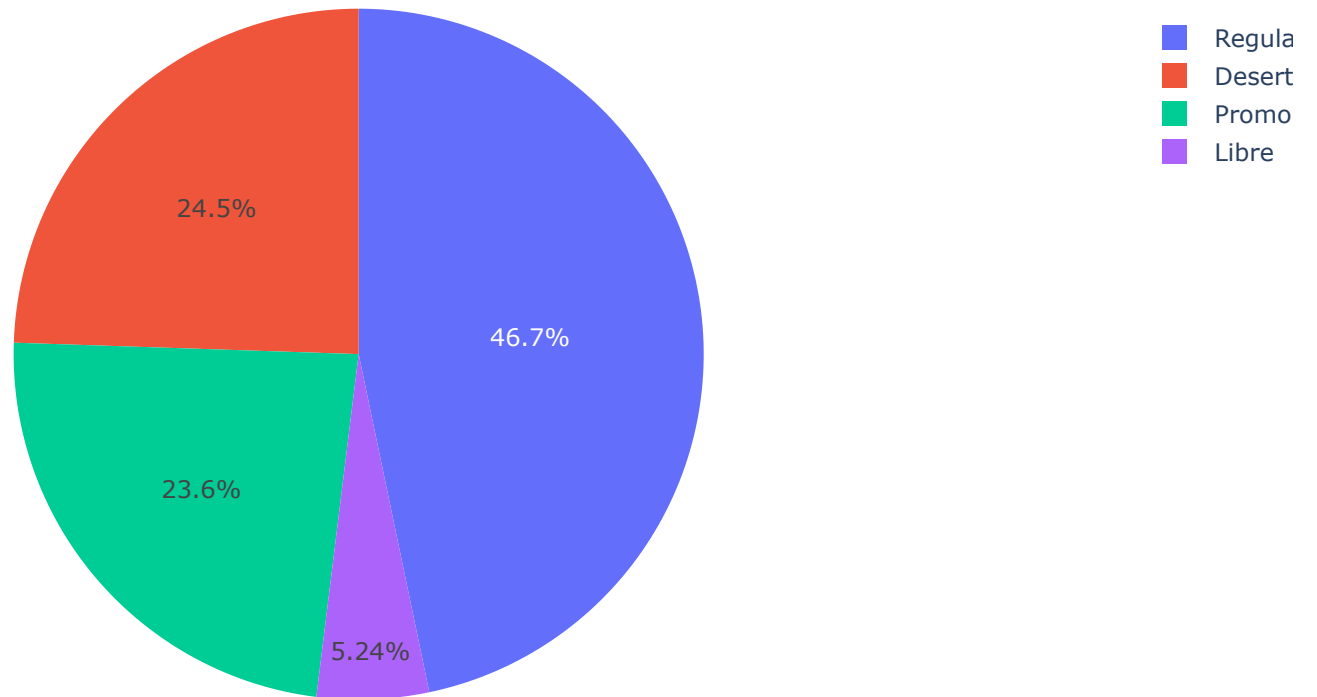
```
In [16]: 1 Con_Est_mat = df_alu_carr_mate.groupby(['Est_mat'],as_index=False)['Id_Alumno'].count()  
2 Con_Est_mat['Estados'] = ['Desertor','Libre','Promocionado','Regular']  
3 Con_Est_mat['Orden'] = [4,3,1,2]  
4 Con_Est_mat = Con_Est_mat.sort_values('Orden')  
5 Con_Est_mat
```

Out[16]:

	Est_mat	Id_Alumno	Estados	Orden
2	P	31092	Promocionado	1
3	R	61628	Regular	2
1	L	6915	Libre	3
0	D	32284	Desertor	4


```
In [17]: 1 fig1 = px.pie(Con_Est_mat, values='Id_Alumno', names= 'Estados', title='Porcentaje estado de alumnos.')
2 fig1.show()
```

Porcentaje estado de alumnos.



Lo que podemos observar en el historial de alumnos de IES es lo siguiente:

- Los Alumnos **Promocionados** cuentan con un porcentaje del 23.6 %, es decir del total de personas que cursaron un aproximado de 31.000 lograron conseguir este estado.

- Los Alumnos **Regulares** cuentan con un porcentaje del 46.7%, es decir del total de personas que cursaron un aproximado de 61.500 lograron conseguir este estado. Siendo el porcentaje más alto de todos.
- Los Alumnos en estado **Libre** con una cantidad aproximada de 6900, siendo el estado con menos porcentaje de todos, representado con un 5.24 % del total.
- Y por último tenemos a los **Desertores** en su caso tienen un total de 32200 aproximadamente, que vendría a ser un 24.5 del conjunto de datos. Es decir, su cantidad supera por poco a los alumnos que consiguieron el estado de promocionados.

In [18]:

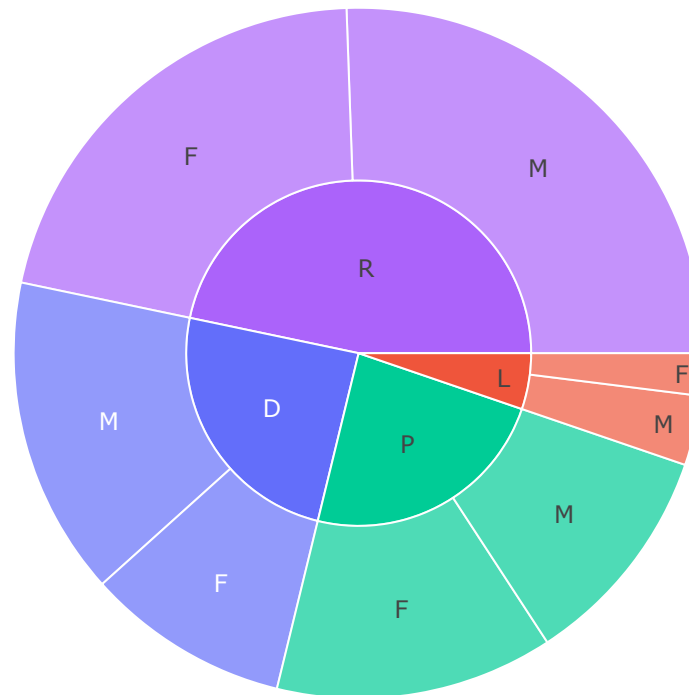
```
1 Con_Est_mat1 = df_alu_carr_mate.groupby(['Est_mat', 'Sexo'], as_index=False)['Id_Alumno'].count()
2 Con_Est_mat1
```

Out[18]:

	Est_mat	Sexo	Id_Alumno
0	D	F	12576
1	D	M	19708
2	L	F	2609
3	L	M	4306
4	P	F	17177
5	P	M	13915
6	R	F	27907
7	R	M	33721

```
In [19]: 1 df = px.data.tips()
2 fig2 = px.sunburst(Con_Est_mat1,path=['Est_mat', 'Sexo'], values= 'Id_Alumno', color = 'Est_mat', title= 'Porcentaje
3 fig2.show()
```

Porcentaje estado de alumnos por sexo.



Es este grafico repetimos lo que mostramos anteriormente, pero le agregamos la clasificación de hombres y mujeres que componen cada estado. Lo que logramos ver es:

- Para **Promocionados** el mayor porcentaje lo tienen las mujeres teniendo un numero aproximado de 17.100 alumnos y los hombres con 13.900 aprox.
- En los **Regulares** la mayor cantidad les queda a los hombres con 33.700 aprox. y las mujeres con 27.900 aprox.
- Para los **Libres** los hombres lideran este lugar con 4.300 alumnos y las mujeres con 2.600 aprox.
- Por último, los hombres son los mayores **Desertores** con 19.700 alumnos y las mujeres los siguen con 12.500 aprox.

DF finales.

```
In [20]: 1 df_finales = pd.read_csv("data/Finales.csv", sep = ';')
        2 df_finales
```

Out[20]:

	Id_Alumno	Id_Carrera	Id_Materia	Semestre_Cursada	Año_Cursada	Fecha_Examen	Nota	Fecha_Inscrip_I
0	0xD2B32B6CE5A79C6AF97CACC9E4BAE8A4	2	68	1	2001	2001-07-27 00:00:00.000	9.0	
1	0xD2B32B6CE5A79C6AF97CACC9E4BAE8A4	2	69	1	2001	2001-07-24 00:00:00.000	10.0	
2	0xD2B32B6CE5A79C6AF97CACC9E4BAE8A4	2	70	1	2001	2001-07-19 00:00:00.000	7.0	
3	0xD2B32B6CE5A79C6AF97CACC9E4BAE8A4	2	71	2	2001	2001-12-07 00:00:00.000	5.0	
4	0xD2B32B6CE5A79C6AF97CACC9E4BAE8A4	2	72	2	2001	2001-12-14 00:00:00.000	11.0	
...	
151763	0x24C9300567A882456A355E68B2980753	5	253	1	2004	2004-07-30 00:00:00.000	5.0	
151764	0x24C9300567A882456A355E68B2980753	5	256	2	2003	2003-12-15 00:00:00.000	6.0	
151765	0x24C9300567A882456A355E68B2980753	5	256	2	2003	2003-12-15 00:00:00.000	6.0	
151766	0x24C9300567A882456A355E68B2980753	5	259	1	2001	2001-07-26 00:00:00.000	9.0	
151767	0x24C9300567A882456A355E68B2980753	5	262	1	2002	2002-12-20 00:00:00.000	5.0	

151768 rows × 8 columns

Al DF que ya teníamos, le agregamos el archivo de finales, trayendo del mismo solamente las columnas que vamos a utilizar obviando columnas ya repetidas por los anteriores archivos, pero usándolas para que se unan los datos en donde corresponden.

```
In [21]: 1 df_final = pd.merge(df_alu_carr_mate,df_finales[['Id_Alumno','Id_Materia','Semetre_Cursada','Año_Cursada','Fecha_Exa
2         left_on = ['Id_Alumno','Id_Materia','Semestre','Año' ],
3         right_on= ['Id_Alumno','Id_Materia','Semetre_Cursada','Año_Cursada'] )
4 df_final
```

Out[21]:

		Id_Alumno	Id_Carrera	Semestre_Ingreso	Año_Ingreso	Id_Ctro_Ext	Id_Modalidad_x	Fecha_Ingreso	Activ
0		0x9530CE8299C7D0B9D1FE0AB99CAF3810	6	1	2001	CO	P	2001-03-02 00:00:00	
1		0x9E60CEA4305B11EAEB97925533D0A266	6	1	2001	CO	P	2001-04-03 00:00:00	
2		0x590850F0100DAEBEF0711D047DB5593E	6	1	2004	CO	P	2004-02-24 00:00:00	
3		0xBF600A4900159A1665015DF67D6073E2	6	1	2004	CO	P	2004-03-18 00:00:00	
4		0x9FB1B0F211AE649DA2DCE7774C79632A	6	1	2001	CO	P	2001-03-13 00:00:00	
...		
152014		0x09CA5ADDE315A3A6BF515D5A9B8F04AC	41	1	2006	CO	P	2006-01-18 00:00:00	
152015		0x09CA5ADDE315A3A6BF515D5A9B8F04AC	41	1	2006	CO	P	2006-01-18 00:00:00	
152016		0xB0A03BBB6C73C7774562C55F5FC33493	59	1	2010	CO	P	2010-03-16 00:00:00	
152017		0xB0A03BBB6C73C7774562C55F5FC33493	59	1	2010	CO	P	2010-03-16 00:00:00	

	Id_Alumno	Id_Carrera	Semestre_Ingreso	Año_Ingreso	Id_Ctro_Ext	Id_Modalidad_x	Fecha_Ingreso	Activ
	152018	0xB0A03BBB6C73C7774562C55F5FC33493	59	1	2010	CO	P	2010-03-16 00:00:00

152019 rows × 33 columns

```
In [22]: 1 def ClasificAcademyFinal(p):
2         #limpieza de datos.
3         if p ==11 : p = np.nan
4
5         #Clasificacion de alumnos.
6         if (p <= 3):
7             return 'R'
8         elif (p >= 4):
9             return 'A'
```

```
In [23]: 1 df_final['Est_mat_Final'] = df_final.apply(lambda f: ClasificAcademyFinal(f['Nota']),axis=1)
```

Dependiendo de la nota que se saco cada alumno en el final, genero un clasificador, con la condición de **R**(Reprobado) y **A**(Aprobado), con esto puede tener un conocimiento un de cómo les va a los alumnos en los finales que rinden. Generando de igual forma una separación entre Hombres y mujeres para comprarlos a cada uno.

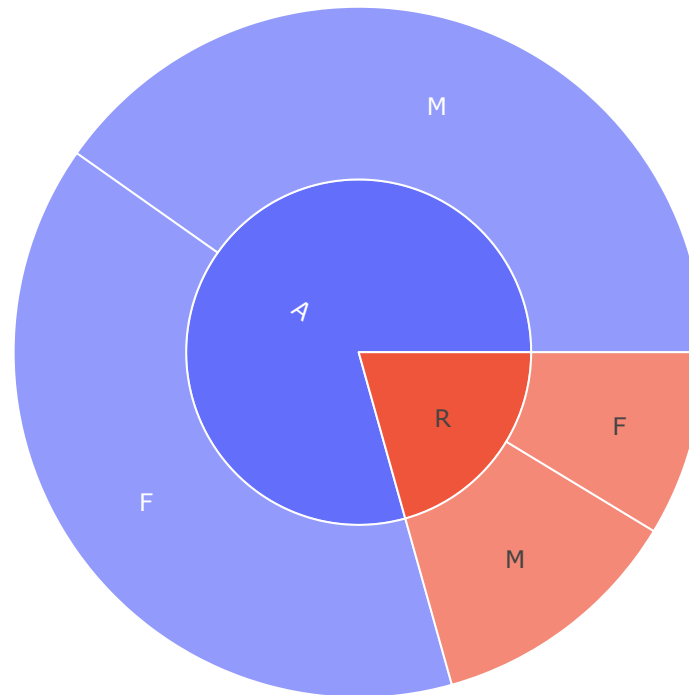
```
In [24]: 1 conteo_estado = df_final.groupby(['Est_mat_Final','Sexo'],as_index = False)['Nota'].count()
2         conteo_estado
```

Out[24]:

	Est_mat_Final	Sexo	Nota
0	A	F	48404
1	A	M	49763
2	R	F	10719
3	R	M	14822

```
In [25]: 1 df = px.data.tips()
2 fig3 = px.sunburst(conteo_estado,path=['Est_mat_Final', 'Sexo'], values= 'Nota', color = 'Est_mat_Final', title= 'Por
3 fig3.show()
```

Porcentaje estado de alumnos por sexo en nota final.



Dependiendo de la nota que se saco cada alumno en el final, genero un clasificador, con la condición de **R**(Reprobado) y **A**(Aprobado), con esto puede tener un conocimiento un de como les va a los alumnos en los finales que rinden. Generando de igual forma una separación entre Hombres y mujeres para comprarlos a cada uno.

Alumnos por materias.

En este caso vamos a generar una nueva variable que muestre que muestre por agrupamiento las carreras y por sexo cada una, para poder saber la cantidad de alumnos y mujeres que las componen.

Esto nos permitiría saber que publico es el que más le interesa cada carrera y de ahí poder generar una estrategia para mejorar la forma en la que captamos la atención de cada tipo de personas.

```
In [26]: 1 aluCarreraSexo = df_final.groupby(['Nombre_Corto', 'Sexo'], as_index=False)['Id_Alumno'].count()
2 aluCarreraSexo
```

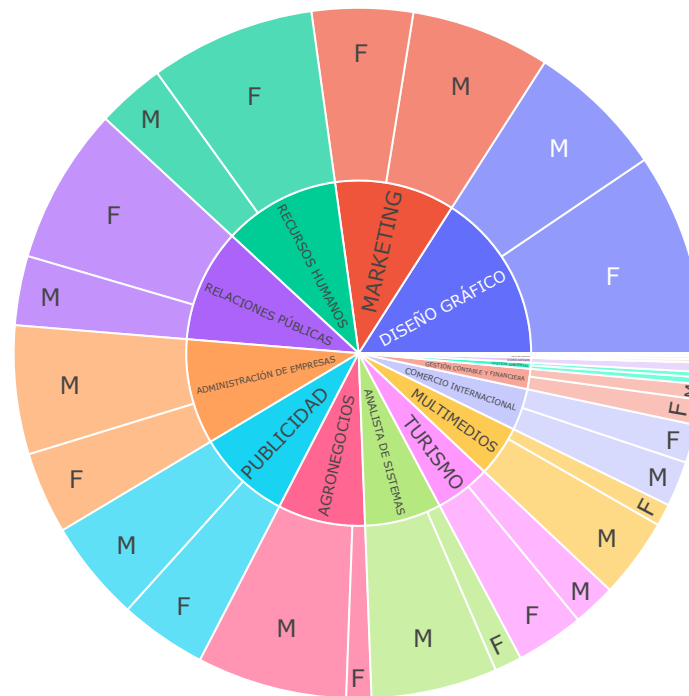
Out[26]:

	Nombre_Corto	Sexo	Id_Alumno
0	ADMINISTRACIÓN DE EMPRESAS	F	5841
1	ADMINISTRACIÓN DE EMPRESAS	M	9155
2	ADMINISTRADOR DE REDES	M	65
3	AGENTE DE TRANSP. ADUANERO Y DESP. DE ADUANA	M	27
4	AGRONEGOCIOS	F	1789
5	AGRONEGOCIOS	M	10682
6	ANALISTA DE SISTEMAS	F	1939
7	ANALISTA DE SISTEMAS	M	8984
8	COMERCIO INTERNACIONAL	F	2710
9	COMERCIO INTERNACIONAL	M	3226
10	DISEÑO GRÁFICO	F	14378
11	DISEÑO GRÁFICO	M	9843
12	GESTIÓN AMBIENTAL	F	485
13	GESTIÓN AMBIENTAL	M	378
14	GESTIÓN CONTABLE Y FINANCIERA	F	1799
15	GESTIÓN CONTABLE Y FINANCIERA	M	1093
16	GESTIÓN JURÍDICA	F	211
17	GESTIÓN JURÍDICA	M	198
18	LOGÍSTICA	F	54
19	LOGÍSTICA	M	30
20	MARKETING	F	7198
21	MARKETING	M	9921
22	MULTIMEDIOS	F	1570

	Nombre_Corto	Sexo	Id_Alumno
23	MULTIMEDIOS	M	5686
24	PUBLICIDAD	F	6197
25	PUBLICIDAD	M	7218
26	RECURSOS HUMANOS	F	11766
27	RECURSOS HUMANOS	M	4847
28	RELACIONES PÚBLICAS	F	11155
29	RELACIONES PÚBLICAS	M	4933
30	SERVICIOS DE PAISAJISMO	F	4
31	TRILINGÜE	F	23
32	TURISMO	F	4848
33	TURISMO	M	3056
34	TÉCNICO EN SERVICIOS AGROPECUARIOS	M	7
35	VENTAS	M	18
36	VIDEOJUEGOS	F	18
37	VIDEOJUEGOS	M	667

```
In [27]: 1 df3 = px.data.tips()
2 fig5 = px.sunburst(aluCarreraSexo,path=['Nombre_Corto','Sexo'], values= 'Id_Alumno', title= 'Cantidad de alumnos por
3 fig5.show()
4 #plot(fig5)
```

Cantidad de alumnos por carrera y sexo



Los datos que nos muestra la variable anterior, ahora lo podemos reflejar en un gráfico. En el cual lo que podemos observar es que para carreras como puede ser Recursos Humanos, Relaciones públicas, diseño gráfico, entre otras, tienen una mayor cantidad de Mujeres las que se inscriben.

Y por ejemplo el caso de Agronegocios, Marketing, Analista de sistemas o Videojuegos, la mayor cantidad de inscriptos son hombres.

Esto nos permitiría entender, el comportamiento de las personas y dependiendo lo que busquemos generar una estrategia para poder equipara la tasa de inscriptos por carrera.

```
In [28]: 1 aluMateriaSexo = df_final.groupby(['Nombre_Corto', 'Nombre', 'Sexo'], as_index=False)['Id_Alumno'].count()  
        2 aluMateriaSexo
```

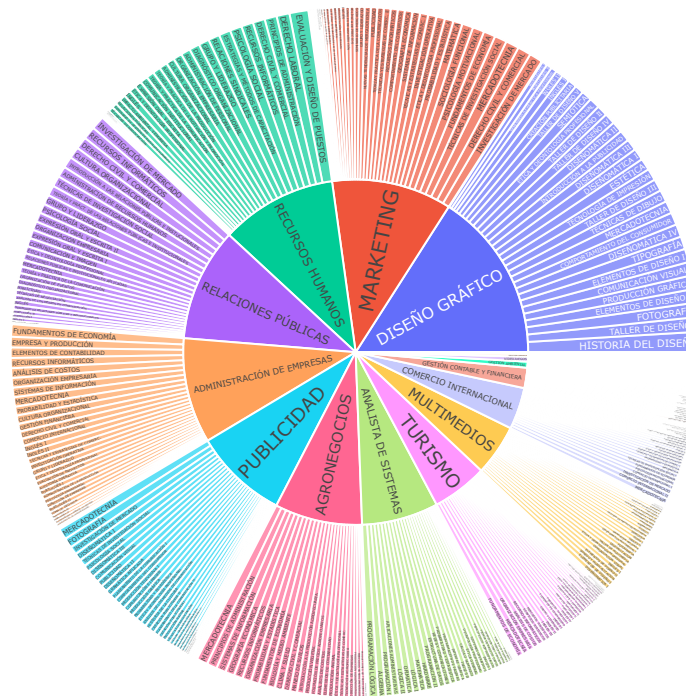
Out[28]:

	Nombre_Corto	Nombre	Sexo	Id_Alumno
0	ADMINISTRACIÓN DE EMPRESAS	ADMINISTRACIÓN DE LA PRODUCCIÓN	F	151
1	ADMINISTRACIÓN DE EMPRESAS	ADMINISTRACIÓN DE LA PRODUCCIÓN	M	187
2	ADMINISTRACIÓN DE EMPRESAS	ANÁLISIS DE COSTOS	F	280
3	ADMINISTRACIÓN DE EMPRESAS	ANÁLISIS DE COSTOS	M	410
4	ADMINISTRACIÓN DE EMPRESAS	AUDITORÍA OPERATIVA	F	153
...
1254	VIDEOJUEGOS	SONIDO Y MUSICALIZACIÓN	M	18
1255	VIDEOJUEGOS	TALLER DE SIMULACIONES VIRTUALES Y VIDEOJUEGOS	M	19
1256	VIDEOJUEGOS	TÉCNICAS DE DIBUJO	F	1
1257	VIDEOJUEGOS	TÉCNICAS DE DIBUJO	M	27
1258	VIDEOJUEGOS	ÉTICA Y DEONTOLOGÍA PROFESIONAL	M	18

1259 rows × 4 columns

```
In [29]: 1 df4 = px.data.tips()
2 fig6 = px.sunburst(aluMateriaSexo,path=['Nombre_Corto','Nombre'], values= 'Id_Alumno', title= 'Cantidad de alumnos p
3 fig6.show()
4 plot(fig6)
```

Cantidad de alumnos por carrera y sexo



Out[29]: 'temp-plot.html'

Luego, con los datos anterior lo que hago es sumarle las materias por carrera, esto permite tener un control sobre los alumnos que van cursando

cada materia. También permite deducir una facturación estimada de cuanto se puede llegar a generar. #materias_Semestres_duracion = df_final.

Llegan al final.

Generamos un FD donde muestre la cantidad de alumnos que empiezan cada carrera y cuantos son los que van dejando la misma.

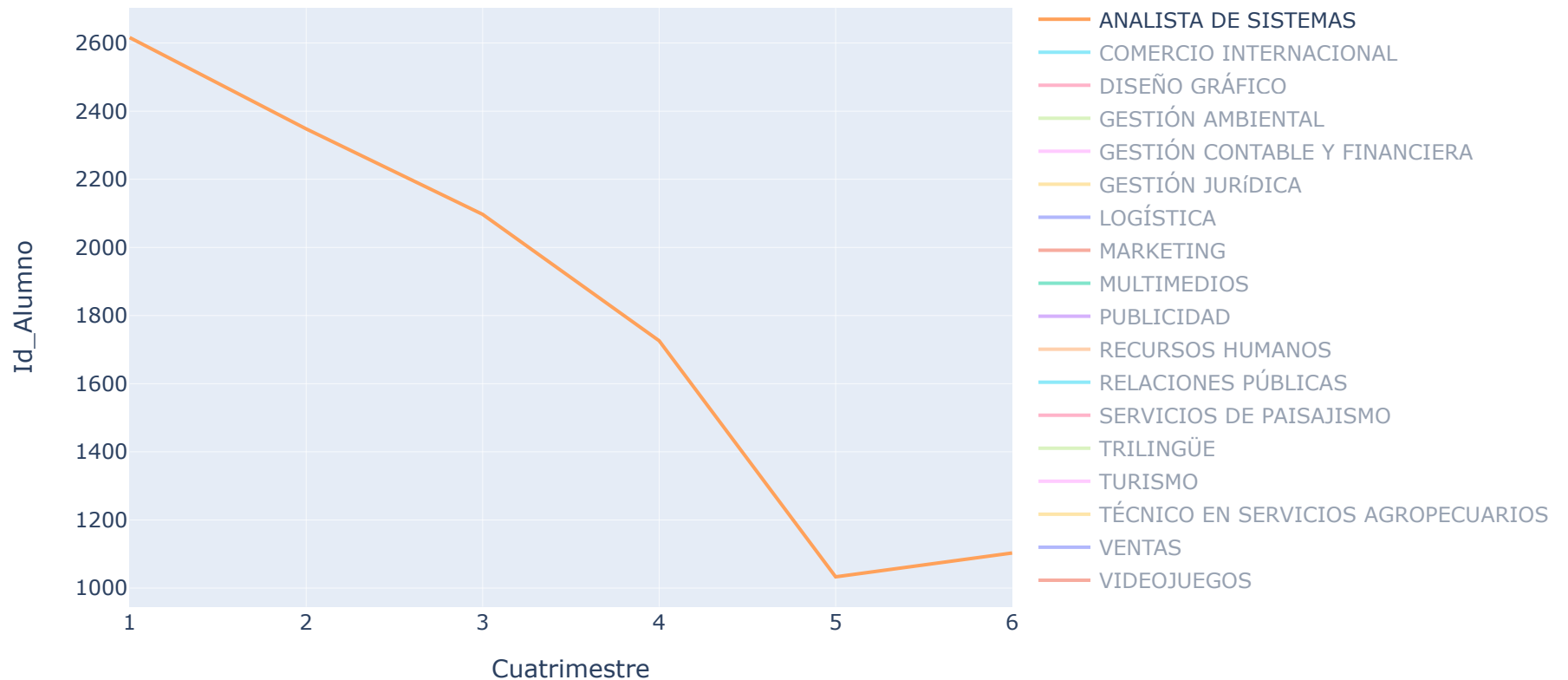
```
In [30]: 1 aluSemestre = df_final.groupby(['Nombre_Corto', 'CantCuatrimestres', 'Cuatrimestre'], as_index=False)['Id_Alumno'].count()
2 aluSemestre[aluSemestre['Nombre_Corto'] == 'ANALISTA DE SISTEMAS']
```

Out[30]:

	Nombre_Corto	CantCuatrimestres	Cuatrimestre	Id_Alumno
20	ANALISTA DE SISTEMAS	6	1	2616
21	ANALISTA DE SISTEMAS	6	2	2348
22	ANALISTA DE SISTEMAS	6	3	2097
23	ANALISTA DE SISTEMAS	6	4	1726
24	ANALISTA DE SISTEMAS	6	5	1033
25	ANALISTA DE SISTEMAS	6	6	1103

```
In [31]: 1 df6 = px.data.gapminder()
2 fig6 = px.line(aluSemestre, x='Cuatrimestre', y='Id_Alumno', color='Nombre_Corto',title='Evolucion de los alumnos in
3 fig6.show()
```

Evolucion de los alumnos inscriptos por carrera que terminan.



Lo que notamos en la mayoría de las carreras, que no son la misma cantidad de alumnos que empiezan y los que terminan. A medida que van pasando los semestres el porcentaje de alumnos va decayendo.

Ahora, vamos a ver el caso de la carrera de Analista de sistemas, en la misma podemos observar como los datos van decreciendo. Se ve en el grafico claramente.

Lo que podemos obtener de esto, es que muchos alumnos, se dieron cuenta que no le gustaba la carrera o que le pareció muy difícil el cursado o que les costó adaptarse a los profesores. Estas serian alguna de las hipótesis que podemos sacar analizando este gráfico y los números presentados.

De la variable **aluSemestre** filtramos los alumnos que llegaron al ultimo semestre de la carrera y realizamos un grafico para saber que cantidad de alumnos que completo la carrera.

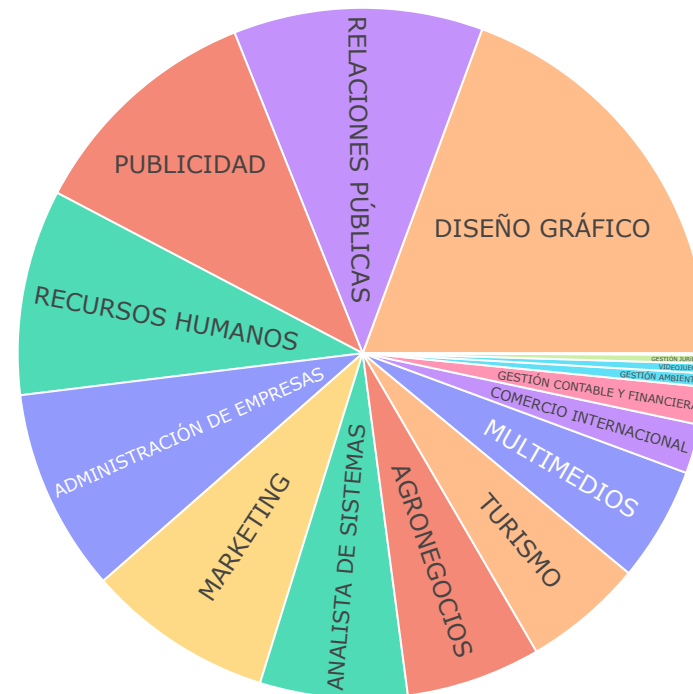
```
In [32]: 1 egresados= aluSemestre[aluSemestre['Cuatrimestre'].isin([6])]
2 egresados.sort_values('Id_Alumno')
```

Out[32]:

	Nombre_Corto	CantCuatrimestres	Cuatrimestre	Id_Alumno
61	LOGÍSTICA	6	6	12
55	GESTIÓN JURÍDICA	6	6	65
116	VIDEOJUEGOS	6	6	72
43	GESTIÓN AMBIENTAL	6	6	101
49	GESTIÓN CONTABLE Y FINANCIERA	6	6	283
31	COMERCIO INTERNACIONAL	6	6	367
73	MULTIMEDIOS	6	6	855
102	TURISMO	6	6	897
19	AGRONEGOCIOS	6	6	1008
25	ANALISTA DE SISTEMAS	6	6	1103
67	MARKETING	6	6	1396
5	ADMINISTRACIÓN DE EMPRESAS	6	6	1520
85	RECURSOS HUMANOS	6	6	1539
79	PUBLICIDAD	6	6	1798
91	RELACIONES PÚBLICAS	6	6	1868
37	DISEÑO GRÁFICO	6	6	3095

```
In [33]: 1 df1 = px.data.tips()
2 fig_egre = px.sunburst(egresados,path=['Nombre_Corto'], values= 'Id_Alumno', color ='Nombre_Corto', title= 'Egresado
3 fig_egre.show()
4
```

Egresados por Carrera.



Inscripciones por año.

```
In [34]: 1 df_final
2 ingreso_por_año = df_final.groupby(['Nombre_Corto', 'Año_Ingreso'], as_index=False)['Id_Alumno'].count()
3 ingreso_por_año
```

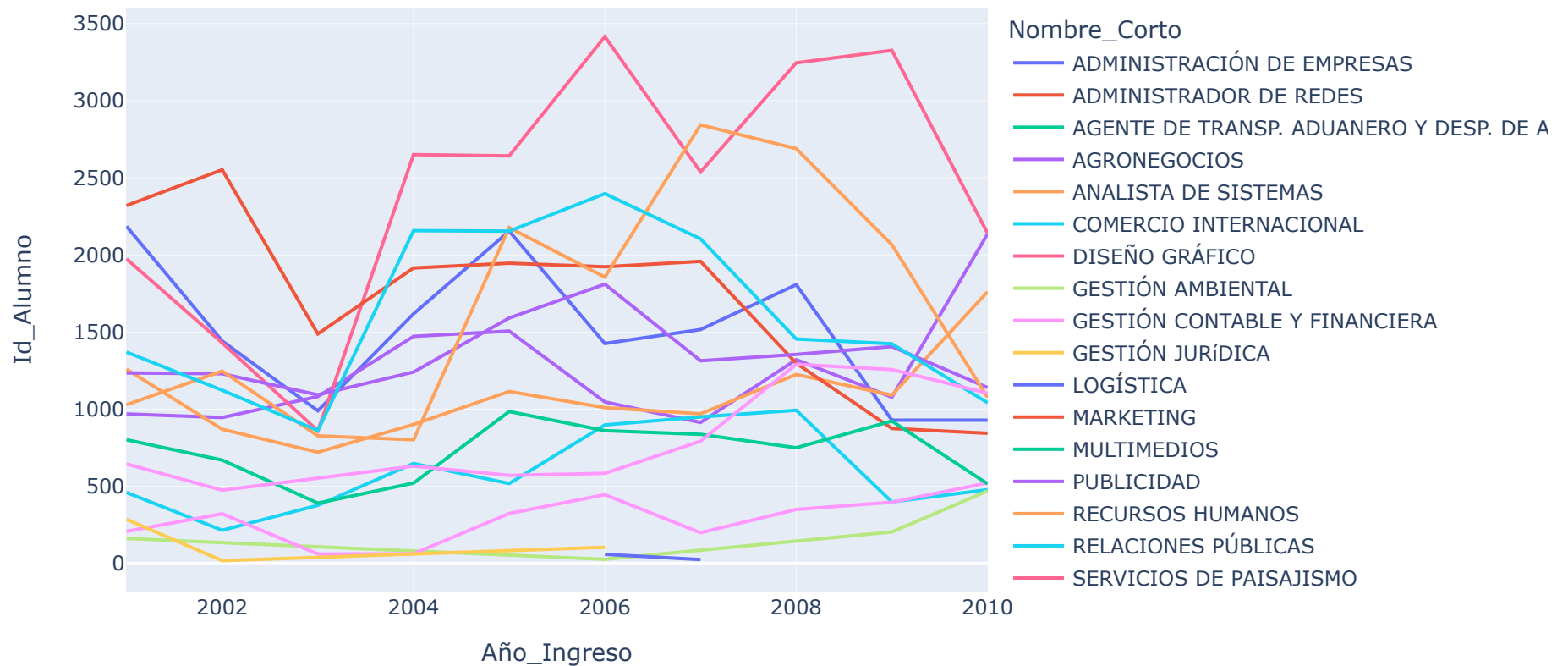
Out[34]:

	Nombre_Corto	Año_Ingreso	Id_Alumno
0	ADMINISTRACIÓN DE EMPRESAS	2001	2185
1	ADMINISTRACIÓN DE EMPRESAS	2002	1442
2	ADMINISTRACIÓN DE EMPRESAS	2003	990
3	ADMINISTRACIÓN DE EMPRESAS	2004	1618
4	ADMINISTRACIÓN DE EMPRESAS	2005	2154
...
131	TURISMO	2009	1257
132	TURISMO	2010	1104
133	TÉCNICO EN SERVICIOS AGROPECUARIOS	2005	7
134	VENTAS	2001	18
135	VIDEOJUEGOS	2010	685

136 rows × 3 columns

```
In [35]: 1 df4 = px.data.gapminder()
2 fig4 = px.line(ingreso_por_año, x='Año_Ingreso', y='Id_Alumno', color='Nombre_Corto',title='Ingreso de alumnos por a
3 fig4.show()
```

Ingreso de alumnos por año y carreras.



En este grafico podemos apreciar como fue el movimiento de inscripciones de alumnos por año en las carreras. Lo que vemos es que muchas carreras tienen muchas variantes a lo largo del conjunto de datos.

También lo que vemos es la aparición de carreras nuevas y como va incrementando su tasa de inscripciones con el tiempo.

Inscripcion por año y semestre.

En este otro conjunto de datos, lo que mostramos es lo mismo que en el anterior, pero ahora lo fraccionamos por semestres.

```
In [36]: 1 df_final
        2 ingreso_por_añoysemestre = df_final.groupby(['Nombre_Corto', 'Año_Ingreso', 'Semestre', ], as_index= False) ['Id_Alumno']
        3 ingreso_por_añoysemestre
```

Out[36]:

	Nombre_Corto	Año_Ingreso	Semestre	Id_Alumno
0	ADMINISTRACIÓN DE EMPRESAS	2001	1	1141
1	ADMINISTRACIÓN DE EMPRESAS	2001	2	1044
2	ADMINISTRACIÓN DE EMPRESAS	2002	1	791
3	ADMINISTRACIÓN DE EMPRESAS	2002	2	651
4	ADMINISTRACIÓN DE EMPRESAS	2003	1	518
...
266	TÉCNICO EN SERVICIOS AGROPECUARIOS	2005	2	3
267	VENTAS	2001	1	11
268	VENTAS	2001	2	7
269	VIDEOJUEGOS	2010	1	320
270	VIDEOJUEGOS	2010	2	365

271 rows × 4 columns

```
In [37]: 1 #primer semestre
2 ingreso_por_año_1ersem = ingreso_por_añoys semestre[ingreso_por_añoys semestre['Semestre'].isin([1])]
3 ingreso_por_año_1ersem = ingreso_por_año_1ersem.sort_values(['Año_Ingreso'])
4 ingreso_por_año_1ersem
```

Out[37]:

	Nombre_Corto	Año_Ingreso	Semestre	Id_Alumno
0	ADMINISTRACIÓN DE EMPRESAS	2001	1	1141
84	DISEÑO GRÁFICO	2001	1	1050
104	GESTIÓN AMBIENTAL	2001	1	91
112	GESTIÓN CONTABLE Y FINANCIERA	2001	1	105
132	GESTIÓN JURÍDICA	2001	1	143
...
82	COMERCIO INTERNACIONAL	2010	1	250
42	AGRONEGOCIOS	2010	1	1233
62	ANALISTA DE SISTEMAS	2010	1	944
242	SERVICIOS DE PAISAJISMO	2010	1	4
269	VIDEOJUEGOS	2010	1	320

136 rows × 4 columns

```
In [38]: 1 #Segundo semestre
2 ingreso_por_año_2dosem = ingreso_por_añoys semestre[ingreso_por_añoys semestre['Semestre'].isin([2])]
3 ingreso_por_año_2dosem = ingreso_por_año_2dosem.sort_values(['Año_Ingreso'])
4 ingreso_por_año_2dosem
```

Out[38]:

	Nombre_Corto	Año_Ingreso	Semestre	Id_Alumno
1	ADMINISTRACIÓN DE EMPRESAS	2001	2	1044
85	DISEÑO GRÁFICO	2001	2	924
105	GESTIÓN AMBIENTAL	2001	2	71
113	GESTIÓN CONTABLE Y FINANCIERA	2001	2	103
133	GESTIÓN JURÍDICA	2001	2	143
...
19	ADMINISTRACIÓN DE EMPRESAS	2010	2	458
181	MULTIMEDIOS	2010	2	231
201	PUBLICIDAD	2010	2	573
103	DISEÑO GRÁFICO	2010	2	1002
270	VIDEOJUEGOS	2010	2	365

135 rows × 4 columns

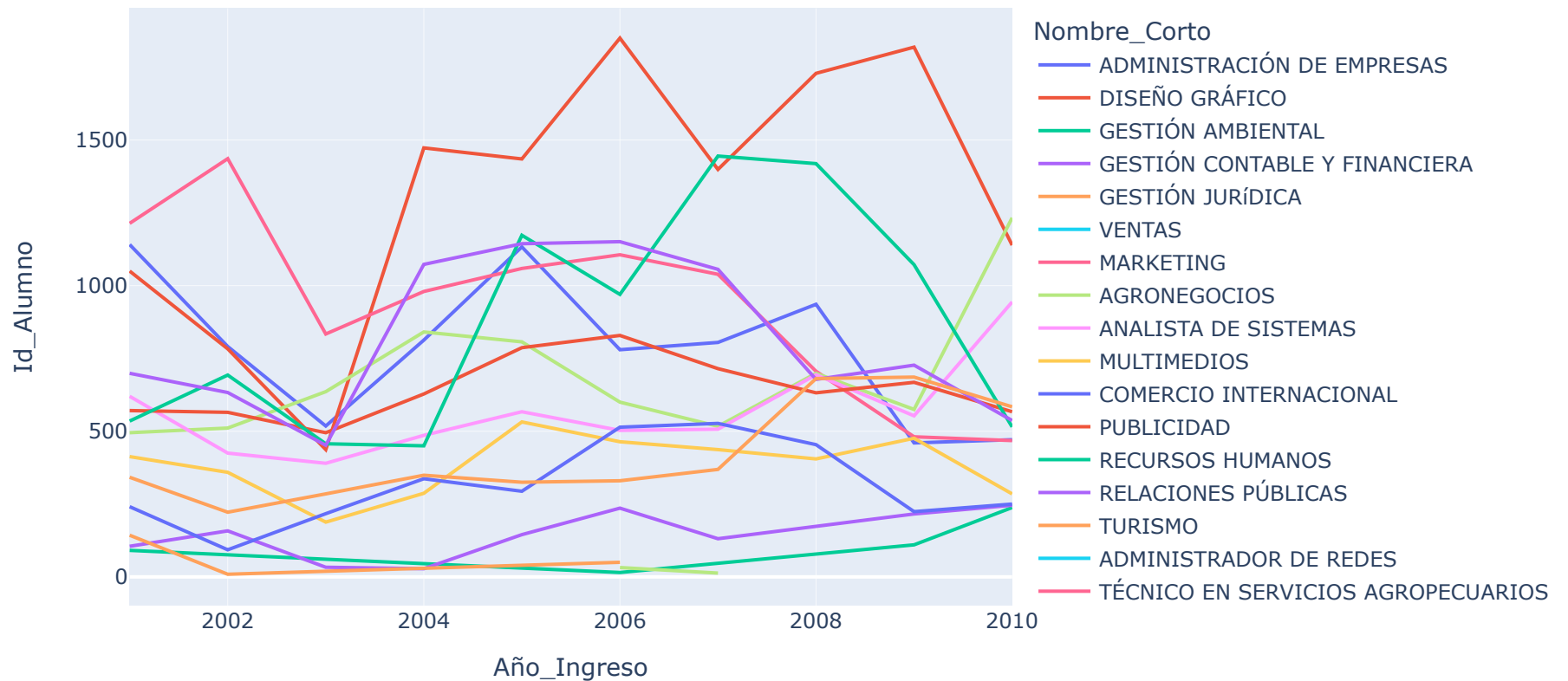
Vamos a calcular un promedio de nota por carrera.


```

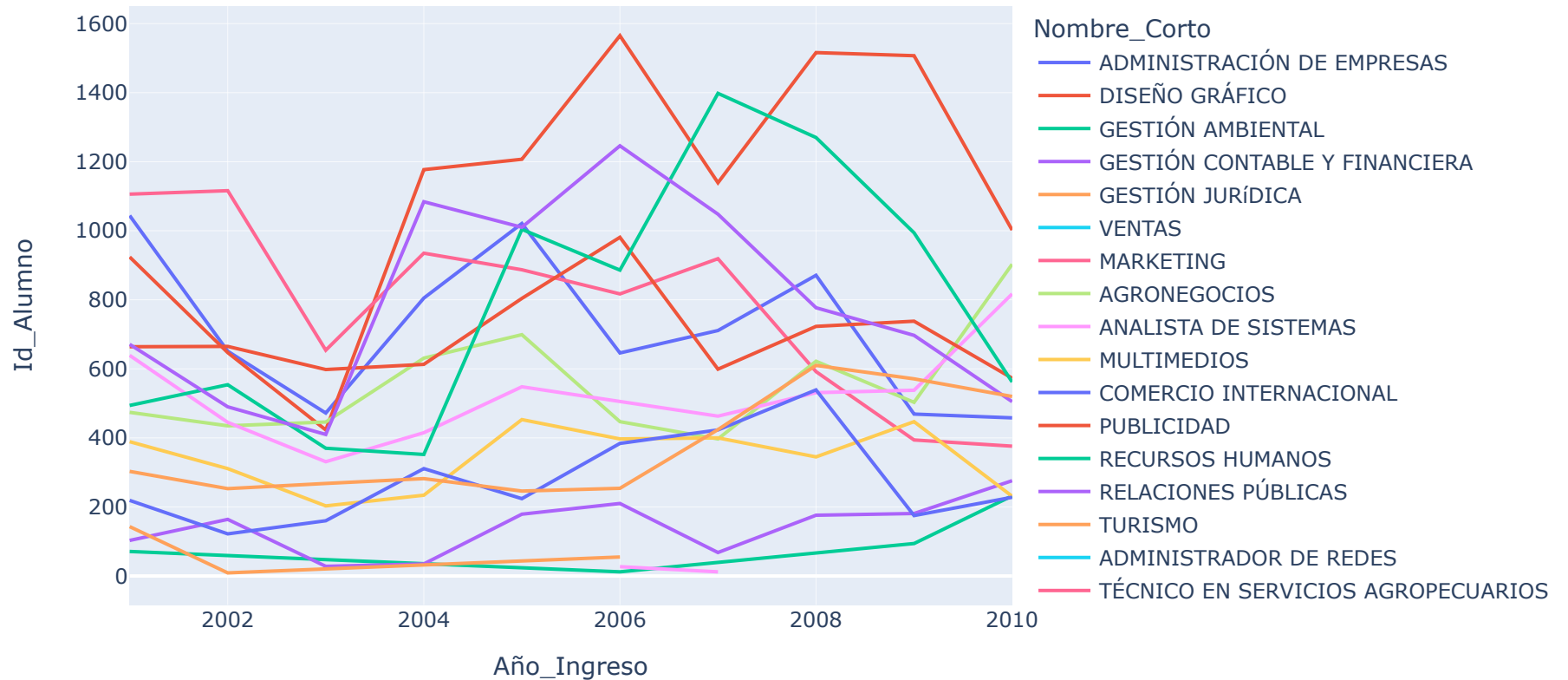
In [39]: 1 #primer semestre
2 df5 = px.data.gapminder()
3 fig5 = px.line(ingreso_por_año_1ersem, x='Año_Ingreso', y='Id_Alumno', color='Nombre_Corto',title='Ingreso de alumno
4 fig5.show()
5 #Segundo semestre
6 df6 = px.data.gapminder()
7 fig6 = px.line(ingreso_por_año_2dosem, x='Año_Ingreso', y='Id_Alumno', color='Nombre_Corto',title='Ingreso de alumno
8 fig6.show()

```

Ingreso de alumnos por año y carreras en el 1er semestre.



Ingreso de alumnos por año y carreras en el 2do semestre.



Lo que notamos, es que la mayor cantidad de alumnos se inscriben en el primer semestre, es decir en la primera parte del año, teniendo una baja cantidad de alumnos inscriptos en la segunda mitad del año.

Ahora, la conclusión que podemos sacar de esto puede ser:

- La mayoría de los alumnos, toman la decisión de iniciar a principio de año por la motivación de ser un año nuevo.
- Se tendría que generar publicidad para mejorar la captación de alumnos en la segunda mitad del año, haciendo referencia, por ejemplo, de que nunca es tarde para empezar a planear tu futuro. La idea seria generar una publicidad que te den ganas de tomar la decisión de estudiar algo para mejorar vos mismo.

- Por otro lado los alumnos de la segunda mitad de año, puede ser que se inscriben poco porque todavía no se pueden decidir a que estudiar.

Promedios por notas en carreras.

Se realizo previo una limpieza de los 11 para que no influyan en los resultados. Vamos a calcular las notas promedio de cada carrera.

```
In [40]: 1 df_final1 = df_final
        2 df_final1.head()
```

Out[40]:

	Id_Alumno	Id_Carrera	Semestre_Ingreso	Año_Ingreso	Id_Ctro_Ext	Id_Modalidad_x	Fecha_Ingreso	Activo	Sexo
0	0x9530CE8299C7D0B9D1FE0AB99CAF3810	6	1	2001	CO	P	2001-03-02 00:00:00	S	F
1	0x9E60CEA4305B11EAEB97925533D0A266	6	1	2001	CO	P	2001-04-03 00:00:00	S	F
2	0x590850F0100DAEBEF0711D047DB5593E	6	1	2004	CO	P	2004-02-24 00:00:00	S	F
3	0xBF600A4900159A1665015DF67D6073E2	6	1	2004	CO	P	2004-03-18 00:00:00	S	M
4	0x9FB1B0F211AE649DA2DCE7774C79632A	6	1	2001	CO	P	2001-03-13 00:00:00	S	F

5 rows × 34 columns



In []:

```
1
```

In [41]:

```
1 #Cambio los 11 de las notas por valores NaN
2 df_final1.loc[df_final1['Nota_1p']==11] = np.nan
3 df_final1.loc[df_final1['Nota_2p']==11] = np.nan
4 df_final1.loc[df_final1['Nota_TP']==11] = np.nan
5 df_final1.loc[df_final1['Nota_Recup']==11] = np.nan
6 df_final1.loc[df_final1['Nota']==11] = np.nan
```

```
In [42]: 1 prom_nota_carrera = df_final1.groupby(['Nombre_Corto'],as_index=False)['Nota_1p','Nota_2p','Nota_TP','Nota_Recup','N
2 prom_nota_carrera
```

<ipython-input-42-2c6ba895a5ab>:1: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

Out[42]:

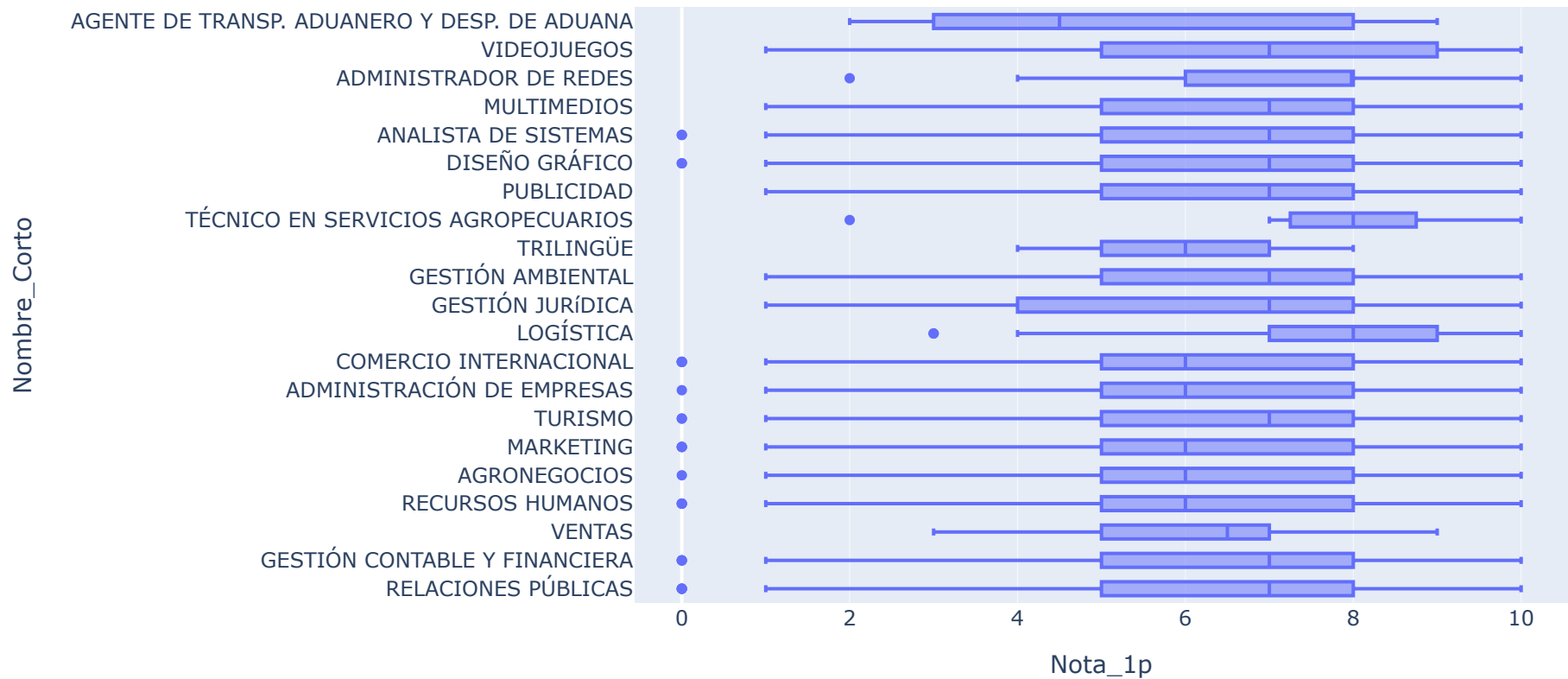
	Nombre_Corto	Nota_1p	Nota_2p	Nota_TP	Nota_Recup	Nota
0	ADMINISTRACIÓN DE EMPRESAS	6.188548	6.325828	8.385955	6.333949	5.857835
1	ADMINISTRADOR DE REDES	7.321429	7.241379	8.714286	7.000000	6.750000
2	AGENTE DE TRANSP. ADUANERO Y DESP. DE ADUANA	5.136364	NaN	7.318182	5.909091	5.500000
3	AGRONEGOCIOS	6.011456	6.310504	8.230600	6.151232	5.427793
4	ANALISTA DE SISTEMAS	6.393333	6.445131	8.334783	6.487814	5.987181
5	COMERCIO INTERNACIONAL	5.893160	6.028272	7.918640	6.365169	5.464947
6	DISEÑO GRÁFICO	6.426010	6.632054	8.056773	6.296819	6.415002
7	GESTIÓN AMBIENTAL	6.708861	6.936809	8.122020	6.195652	6.985975
8	GESTIÓN CONTABLE Y FINANCIERA	6.344628	6.622035	8.418708	6.385113	6.018263
9	GESTIÓN JURÍDICA	6.167722	6.803738	9.329154	5.623762	6.360502
10	LOGÍSTICA	7.805195	8.666667	8.410256	5.750000	8.141026
11	MARKETING	6.145343	6.231967	8.375459	6.340311	5.588441
12	MULTIMEDIOS	6.676213	6.710040	8.374342	6.478862	6.577056
13	PUBLICIDAD	6.397820	6.535063	8.231903	6.458517	6.138363
14	RECURSOS HUMANOS	6.237733	6.341048	7.895585	6.218965	5.787254
15	RELACIONES PÚBLICAS	6.319584	6.585827	8.183698	6.272005	6.021649
16	TRILINGÜE	6.000000	6.750000	7.187500	8.400000	6.625000
17	TURISMO	6.221086	6.315242	8.051210	6.011465	5.696285

	Nombre_Corto	Nota_1p	Nota_2p	Nota_TP	Nota_Recup	Nota
18	TÉCNICO EN SERVICIOS AGROPECUARIOS	7.428571	7.285714	10.000000	6.000000	6.857143
19	VENTAS	6.250000	7.166667	10.000000	6.000000	4.416667
20	VIDEOJUEGOS	6.930728	6.838532	8.129663	6.246269	6.959147

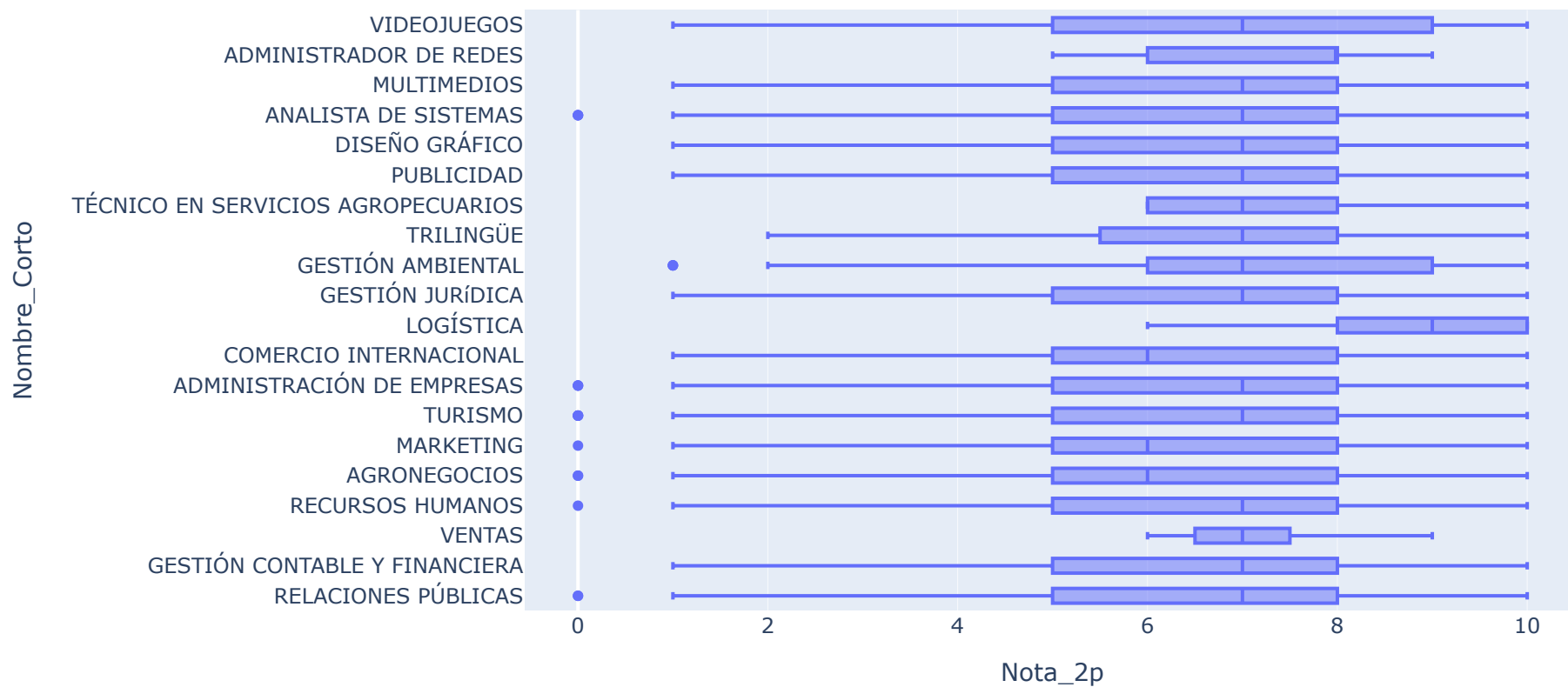


```
In [43]: 1 df = px.data.tips()
2 prom = px.box(df_final1,x='Nota_1p' ,y="Nombre_Corto",title='Primera nota.')
3 prom.show()
4 prom1 = px.box(df_final1,x='Nota_2p' ,y="Nombre_Corto",title='Segunda nota.')
5 prom1.show()
6 prom2 = px.box(df_final1,x='Nota_TP' ,y="Nombre_Corto",title='Nota TP.')
7 prom2.show()
8
```

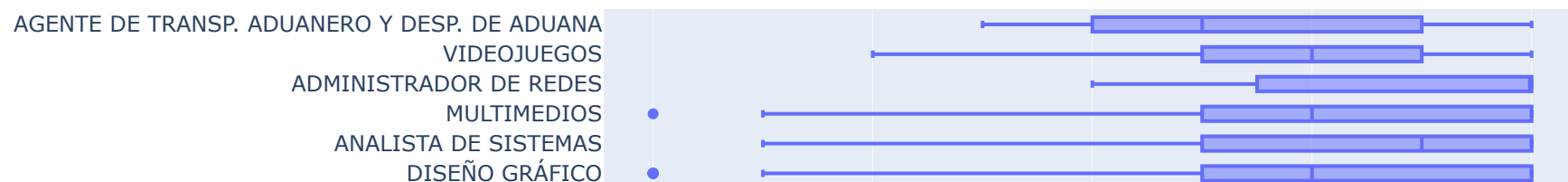
Primera nota.



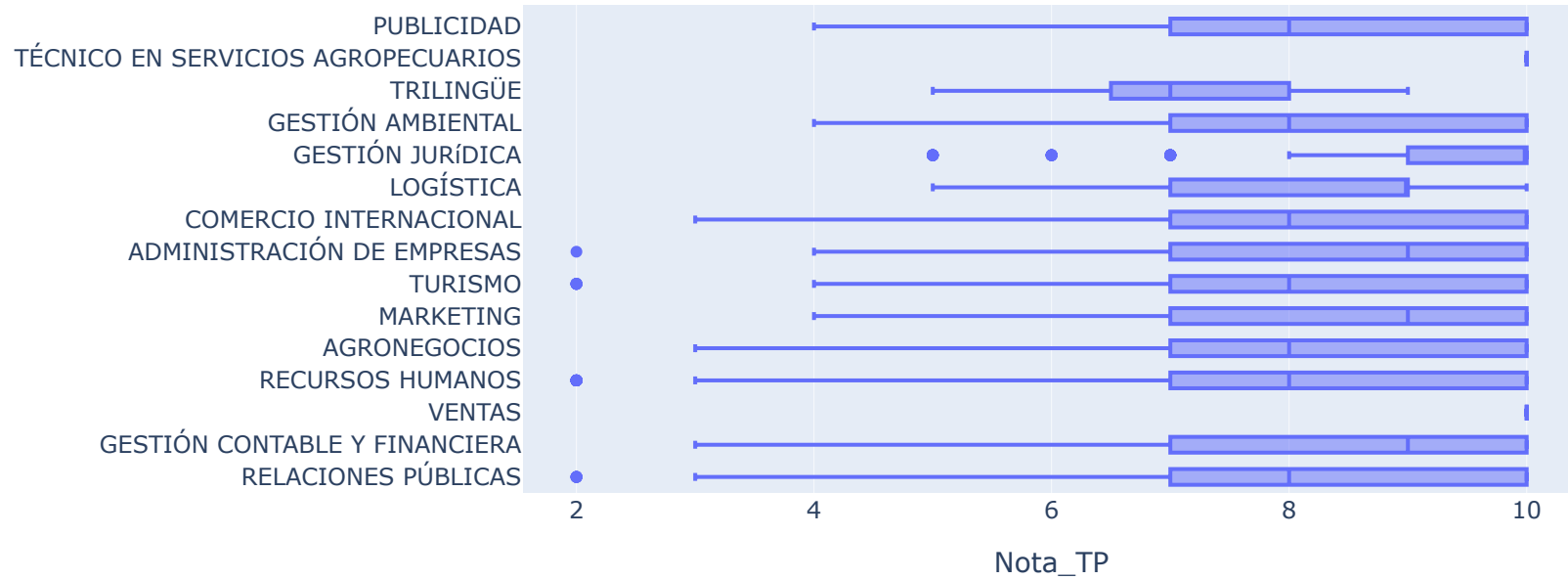
Segunda nota.



Nota TP.

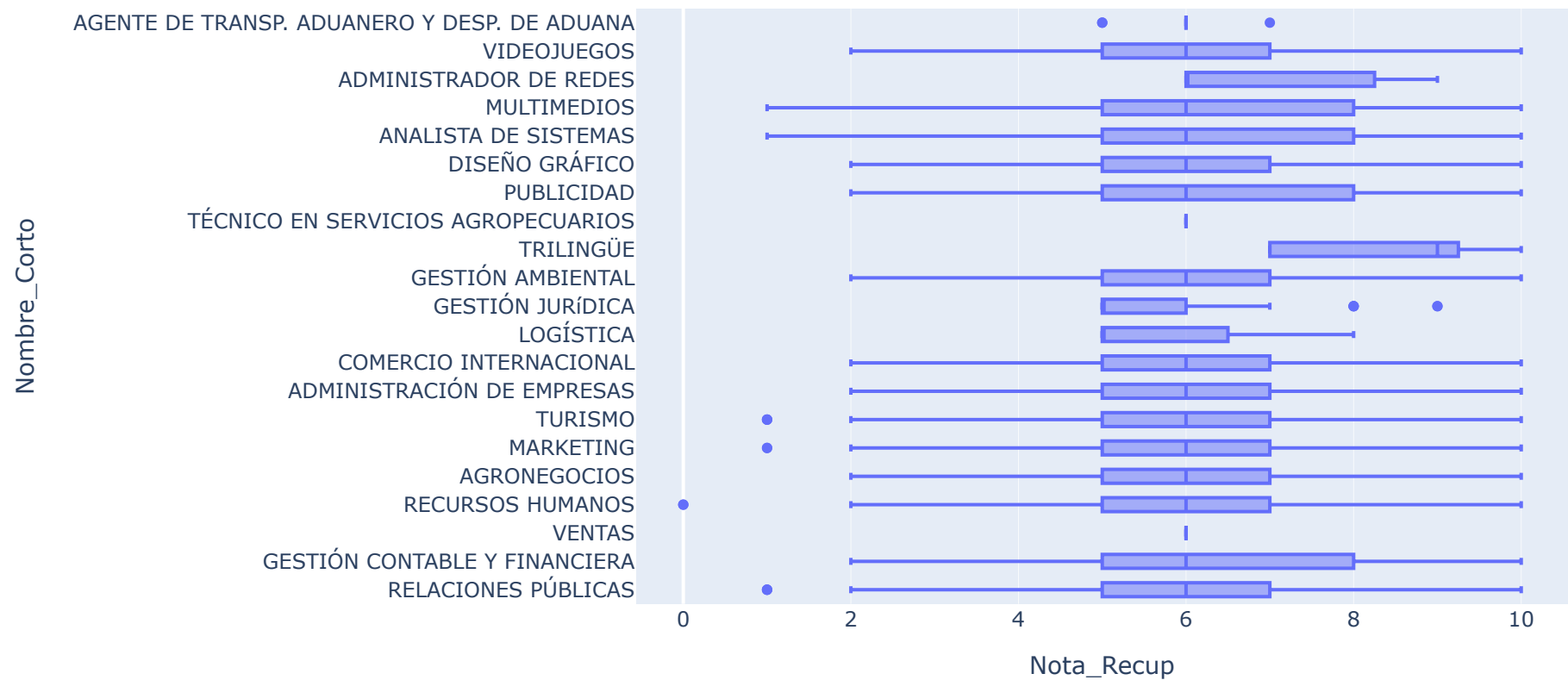


Nombre_Corto

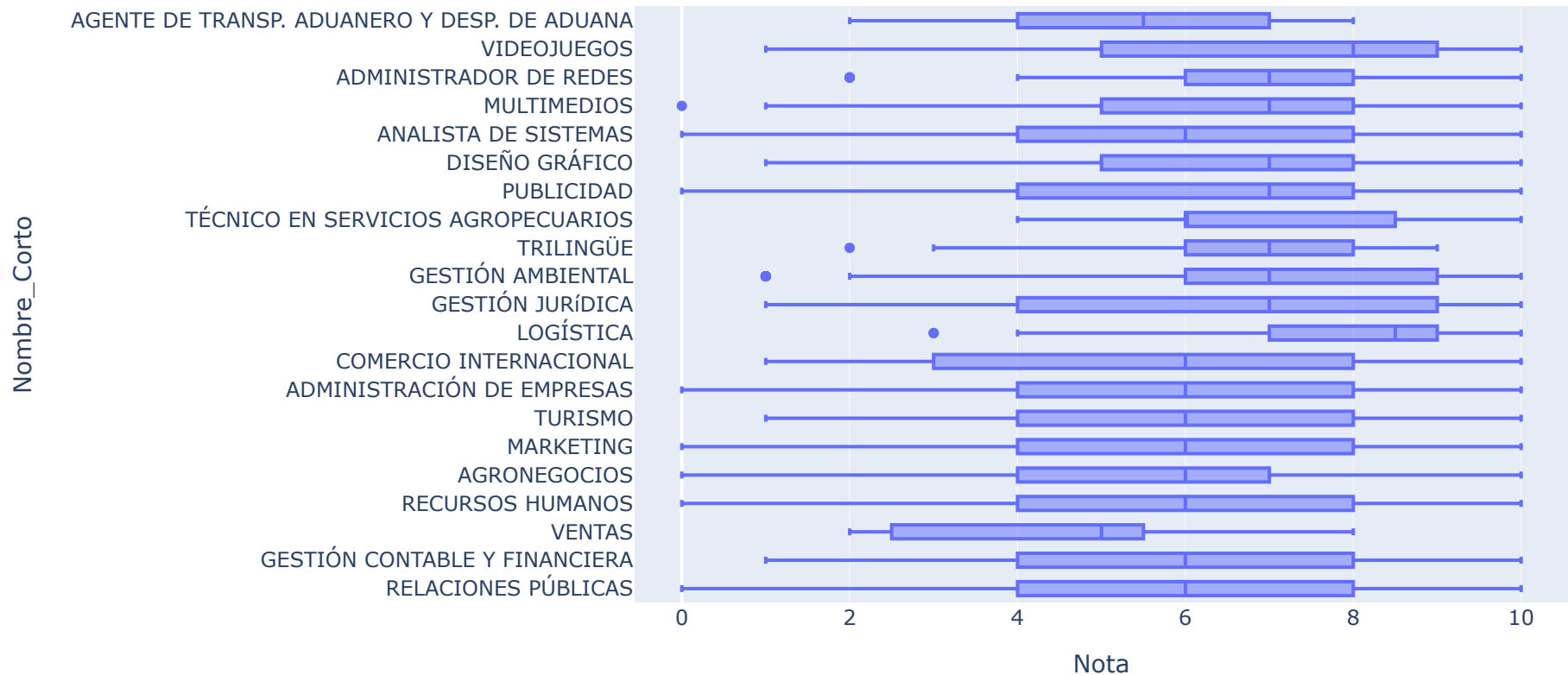



```
In [44]: 1 df = px.data.tips()
2 prom4 = px.box(df_final1,x='Nota_Recup' ,y="Nombre_Corto",title='Nota recuperatorio.')
3 prom4.show()
4
5 prom5 = px.box(df_final1,x='Nota' ,y="Nombre_Corto",title='Nota final.')
6 prom5.show()
7
```

Nota recuperatorio.



Nota final.



Con los gráficos que hemos realizado de los promedios de las notas podemos interpretar o deducir lo siguiente:

- Las carreras a las cuales hay que prestarles más atención por su bajo nivel en el promedio de las notas, eso puede llevar a la deserción de alumnos por la dificultad de la misma. Es decir, lo que tendríamos que hacer, es mejorar el sistema en el cual dictamos las clases, para que se puede lograr un mayor entendimiento de las materias y eso sirva para que los alumnos continúen con la carrera.
- Por otro lado, de acá también podemos llevar un control de como es el rendimiento académico con respecto a las notas que los alumnos van sacando.

Promedios por notas en carreras y sus materias.

```
In [45]: 1 prom_nota_materias = df_final1.groupby(['Nombre_Corto', 'Nombre'], as_index=False)[ 'Nota_1p', 'Nota_2p', 'Nota_TP', 'Nota
2 prom_nota_materias
```

<ipython-input-45-3605928de203>:1: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

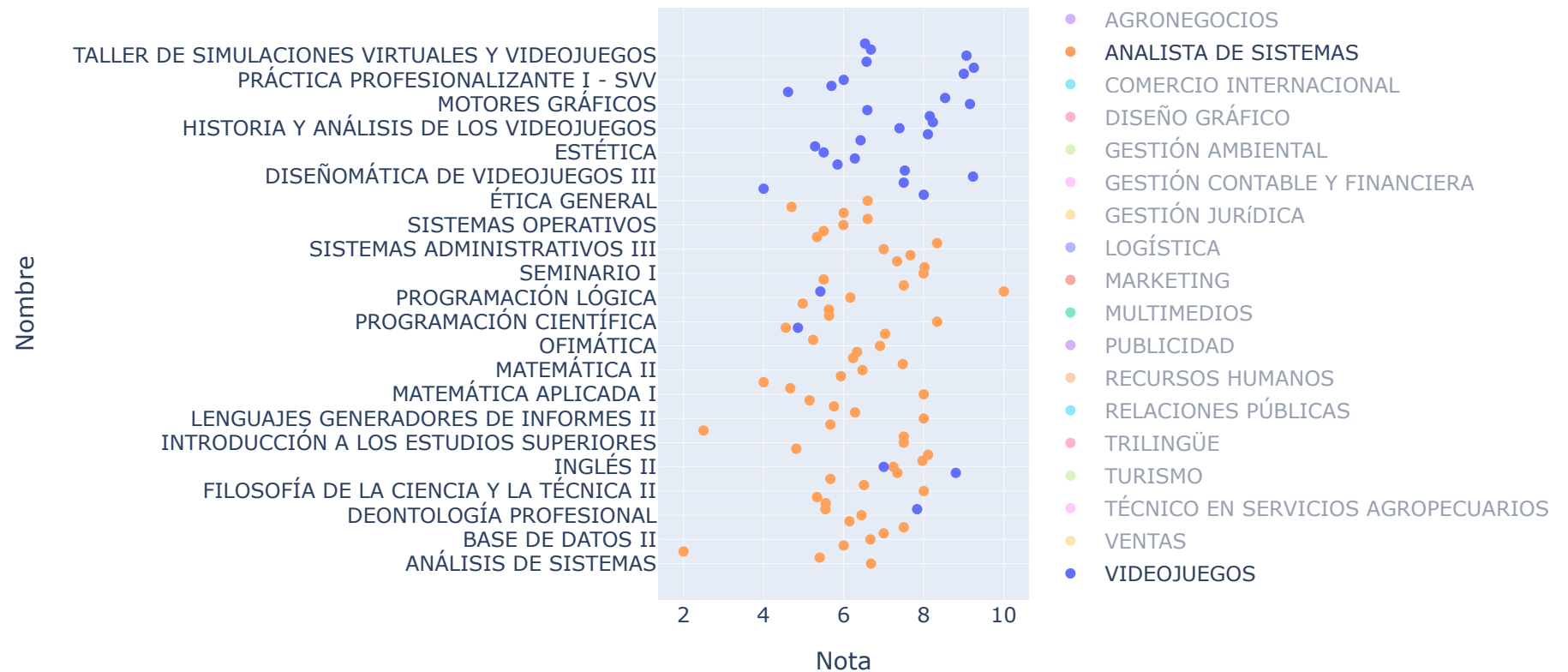
Out[45]:

	Nombre_Corto	Nombre	Nota_1p	Nota_2p	Nota_TP	Nota_Recup	Nota
0	ADMINISTRACIÓN DE EMPRESAS	ADMINISTRACIÓN DE LA PRODUCCIÓN	6.664093	6.956140	8.806950	7.237288	7.355212
1	ADMINISTRACIÓN DE EMPRESAS	ANÁLISIS DE COSTOS	5.512245	5.578341	8.685598	6.786667	4.732252
2	ADMINISTRACIÓN DE EMPRESAS	AUDITORÍA OPERATIVA	6.710843	6.495146	7.827309	6.346939	6.160643
3	ADMINISTRACIÓN DE EMPRESAS	COMERCIO EXTERIOR	7.117647	6.205882	10.000000	7.333333	6.088235
4	ADMINISTRACIÓN DE EMPRESAS	COMERCIO INTERNACIONAL	6.429333	6.065868	8.173333	5.922330	5.861333
...
672	VIDEOJUEGOS	SEMINARIO FINAL DE SIMULACIONES VIRTUALES Y VI...	7.666667	8.083333	7.750000	NaN	9.250000
673	VIDEOJUEGOS	SONIDO Y MUSICALIZACIÓN	6.428571	5.500000	8.214286	7.714286	6.571429
674	VIDEOJUEGOS	TALLER DE SIMULACIONES VIRTUALES Y VIDEOJUEGOS	8.533333	7.400000	8.266667	NaN	9.066667
675	VIDEOJUEGOS	TÉCNICAS DE DIBUJO	6.214286	6.740741	8.142857	6.800000	6.678571
676	VIDEOJUEGOS	ÉTICA Y DEONTOLOGÍA PROFESIONAL	6.266667	4.466667	7.533333	5.727273	6.533333

677 rows × 7 columns

```
In [46]: 1 fig07 = px.scatter(prom_nota_materias, x = 'Nota', y = 'Nombre', color= 'Nombre_Corto', title='Dispercion de promedio
2 fig07.show()
3
```

Dispercion de promedio por carreras.



Ahora creo que esta es la parte mas importante para poder saber la deserción de los alumnos. Digo esto porque acá ya tenemos un control mas fino sobre las materias que producen mayores deserciones por su dificultad. Esto nos permite:

- Mejorar el plan de estudio de las mismas.

- Saber si el docente que esta a cargo esta desempeñando bien su rol.
- Saber si es necesario agregar herramientas para ayudar a la comprensión de los alumnos con dicha materia.
- Identificar el grado de dificultad de la materia con respecto al grado de importancia de dicha materia en esta carrera.
- Saber si es una materia filtro, es decir la materia sacaría a la gente que no esta 100% comprometida con el estudio de esta carrera. Esto también permite mejorar el acompañamiento de los alumnos para saber como afrontar dicha materia y carrera y que no genera que los alumnos dejen la institución.

Documentación.

Recursos tecnológicos utilizados.

- Código desarrollado en Python.
- Se uso el entorno de Anaconda.
- La investigacion se realizo usando Jupyter Notebook.

Contenidos y herramientas integrados para lograr los informes desarrollados.

- Utilizando las librerías:
 - Numpy: Para el manejo de datos estadísticos y matemáticos. www.numpy.org (<http://www.numpy.org>)
 - Pandas: Para manejo de estructura de datos. www.pandas.pydata.org
 - Plotly: Para graficas. www.plotly.com
 - Word: Para realizar la documentación y para la corrección en el caso de haber algún error ortográfico.

Objetivos particulares perseguidos.

- Identificar cuáles son las materias principales que están dando problema a los alumnos y que pueden generar la deserción de los mismos.

Apreciación general sobre la tarea realizada.

Se logro identificar posibles causas por las cuales los alumnos abandonan el cursado de materias y también por los cuales los mismo también dejan la carrera. Llevo todo un tiempo de investigación de los datos. Pero mis objetivos planteados se lograron adquirir.

