

Visualización de datos existentes y simulados | Retorno a las clases presenciales

En este trabajo se tratará el tema de retorno de clases presenciales dentro de los años 2020 y 2021 frente al impacto de la enfermedad Covid 19, con esto en mente, se plantea la implementación de un datamart y así obtener indicadores del retorno a clases presenciales.

Diseño de Entidades (tablas)

Nombre de la entidad	Atributos
AÑO_CURSADO_UNIVERSIDAD	Año (PK)
	Nombre de la universidad
	Cantidad de profesores contratados
	Cantidad de matriculados por año
	Tipo de modalidad ("Semipresencial", "Virtual")

Nombre de la entidad	Atributos
UNIVERSIDAD	Nombre universidad (PK)
	Distrito
	Provincia
	Departamento
	Gestión ("Pública", "Privada")

Nombre de la entidad	Atributos
AÑO_CURSADO_ESTUDIANTE	Dni estudiante
	año
	Promedio del estudiante
	Situación académica ("normal", "observado", "suspendido")

Nombre de la entidad	Atributos
----------------------	-----------

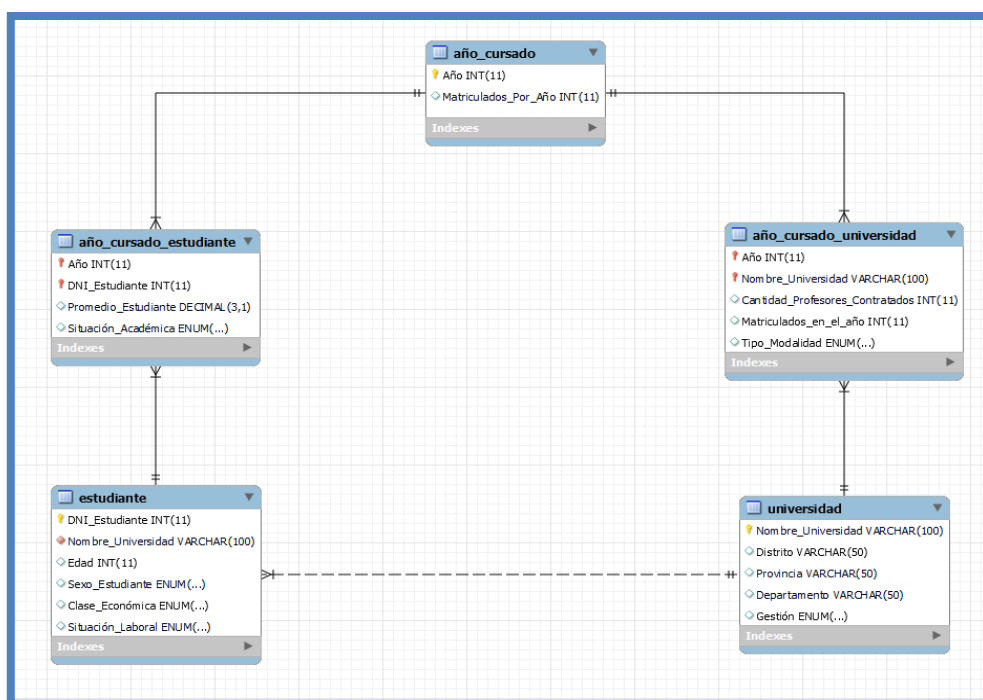
ESTUDIANTE	Dni (PK)
	Nombre de la universidad (FK)
	edad
	Sexo del estudiante
	Clase económica (A hasta E)
	Situación laboral ("Solo estudia", "Estudia y trabaja")

Nombre de la entidad	Atributos
AÑO_CURSADO	Año (PK)
	Cantidad de matriculados por año

1.1.1. Modelo Entidad Relación

El diseño de la base de datos debe estar acorde con los requerimientos planteados para acceder a la información con la que cuenta la organización.

A continuación, se muestra las entidades mencionadas anteriormente y las relaciones establecidas entre ellas con el fin de facilitar la posterior conexión de los datos.



Se simularon los datos de las siguientes entidades:

Estudiante:

En el caso de los datos para estudiantes, se realizaron simulaciones de datos con la función `sample()` en R y demás. La creación se realizó de la siguiente manera:

Primero necesitaremos la llave foránea `Nombre_Universidad` de la entidad `Universidad`, pero como únicamente realizaremos la simulación con universidades las cuales cuenten con número de matriculados mayor a 0, usaremos la entidad `Año_cursado_universidad` en donde realizaremos el filtro deseado.

```
año_cursado_universidad <- read.csv2("year_universidad.csv")
año_cursado_universidad_mat <- año_cursado_universidad[año_cursado_universidad$Matriculados_en_el_año > 0,]
set.seed(2022)
DNI_Estudiante <- sample(70000000:79999999, 1375101, replace = F)
Edad <- sample(c(16:30), 1375101, replace = T)
Sexo_Estudiante <- sample(c("Masculino", "Femenino"), 1375101, replace = T)
Clase_Económica <- sample(LETTERS[1:5], 1375101, replace = T)
Situación_Laboral <- sample(c("Solo estudia", "Estudia y trabaja"), 1375101, replace = T)
Nombre_Universidad <- sample(año_cursado_universidad_mat$Nombre_Universidad, 1375101, replace = T)
estudiantes <- data.frame(DNI_Estudiante, Nombre_Universidad, Edad, Sexo_Estudiante, Clase_Económica,
Situación_Laboral)
```

Una vez usado solo las universidades con matriculados mayores a 0 y realizadas las simulaciones tal cual nos muestra la imagen del código, ya estamos listos para exportar nuestra data con la función `write.csv2()` la cual `mysql workbench` importará.

```
#Write en csv para leerlo en nuestro mysql
write.csv2(estudiantes, "estudiantes.csv", fileEncoding = "UTF-8", quote = F, row.names = F)
```

Año_cursado_estudiante:

En el caso de la entidad `"Año_cursado_estudiante"` también se realizaron simulaciones, para esto usamos la data de la entidad `"estudiantes"` ya que de esos estudiantes simulados se simulará la data. Para realizar esto lo dividimos en dos partes, año 2020 y 2021 para luego juntarlos con una función de R llamada `rbind()`.

```
18 ##Año_Estudiante
19
20 #Para el año estudiante 2020
21
22 estudiante <- read.csv2("estudiantes.csv")
23 Año <- rep(2020, 1234549)
24 DNI_Estudiante <- sample(estudiante$DNI_Estudiante, 1234549, replace = F)
25 Promedio_Estudiante <- round(runif(1234549, min = 8, max = 20), 1)
26 cantidad_r <- c(-Inf, 10, 12, Inf)
27 valores <- c('Suspendido', 'Observado', 'Normal')
28 Situación_Académica <- cut(Promedio_Estudiante, breaks = cantidad_r, labels = valores)
29 dataframe1 <- data.frame(Año, DNI_Estudiante, Promedio_Estudiante, Situación_Académica)
30
31
32 #Para el año estudiante 2021
33 Año <- rep(2021, 1375101)
34 DNI_Estudiante <- sample(estudiante$DNI_Estudiante, 1375101, replace = F)
35 Promedio_Estudiante <- round(runif(1375101, min = 8, max = 20), 1)
36 Situación_Académica <- cut(Promedio_Estudiante, breaks = cantidad_r, labels = valores)
37 dataframe2 <- data.frame(Año, DNI_Estudiante, Promedio_Estudiante, Situación_Académica)
38 year_estudiantes <- rbind(dataframe1, dataframe2)
39
```

Culminando la creación de los datos, lo guardamos con la función `write.table()` de R, hacemos uso de esta función y no de `write.csv2()` ya que en la primera podemos especificar el separador de decimales, con esto nos evitamos problemas a la hora de inserción de datos a `MySQL Workbench`.

```

39
40 #Write en csv para leerlo en nuestro mysql
41
42 write.table(year_estudiantes, "year_estudiantes.csv", fileEncoding = "UTF-8",
43             quote = F, row.names = F, dec = ".", sep = ";")
44

```







Finalmente, solo queda exportar nuestra data para luego ser insertada en nuestro MySQL Workbench. (Se usa de nuevo write.table() debido a que se necesita especificar decimales nuevamente).

```

60
61 write.table(tabla_hechos, "tabla_hechos.csv", fileEncoding = "UTF-8",
62             quote = F, row.names = F, dec = ".", sep = ";")
63

```

Con esto tenemos lista toda nuestra data para ser insertada a cada entidad que crearemos a continuación en nuestro MySQL Workbench.

Nombre	Fecha de modificación	Tipo	Tamaño
 estudiantes.csv	12/01/2023 13:50	Archivo de valores...	108,913 KB
 tabla_hechos.csv	12/01/2023 14:28	Archivo de valores...	246,288 KB
 universidades.csv	10/01/2023 22:23	Archivo de valores...	11 KB
 year_estudiantes.csv	12/01/2023 15:25	Archivo de valores...	70,904 KB
 year_universidad.csv	10/01/2023 21:34	Archivo de valores...	18 KB
 years.csv	10/01/2023 22:31	Archivo de valores...	1 KB

Creación de las entidades:

Para la creación de las entidades primero crearemos nuestra base de datos, la llamaremos "tf_bd2_", luego la seleccionaremos y podremos empezar a crear nuestras tablas de las entidades de la siguiente manera.

```
1 #####Creamos la base de datos para nuestro modelo entidad relación
2 • CREATE DATABASE IF NOT EXISTS tf_bd2_entidadR;
3
4 #Seleccionar la base de datos
5 • USE tf_bd2_entidadR;
6
7 #Creamos las tabla Universidad
8 • CREATE TABLE IF NOT EXISTS Universidad(
9     Nombre_Universidad varchar(100) NOT NULL,
10    Distrito varchar(50),
11    Provincia varchar(50),
12    Departamento varchar(50),
13    Gestión ENUM("Público","Privado"),
14    PRIMARY KEY(Nombre_Universidad)
15 )ENGINE=INNODB;
16
17 #Creamos las tabla Estudiante
18 • CREATE TABLE IF NOT EXISTS Estudiante(
19    DNI_Estudiante INT NOT NULL,
20    Nombre_Universidad varchar(100) not null,
21    constraint Fk_Unversidad
22    foreign key(Nombre_Universidad)
23    references Universidad(Nombre_Universidad),
24    Edad int,
25    Sexo_Estudiante ENUM("Masculino","Femenino"),
26    Clase_Económica ENUM("A","B","C","D","E"),
27    Situación_Laboral ENUM("Sólo estudia","Estudia y trabaja"),
28    PRIMARY KEY(DNI_Estudiante)
29 )ENGINE=INNODB;
```

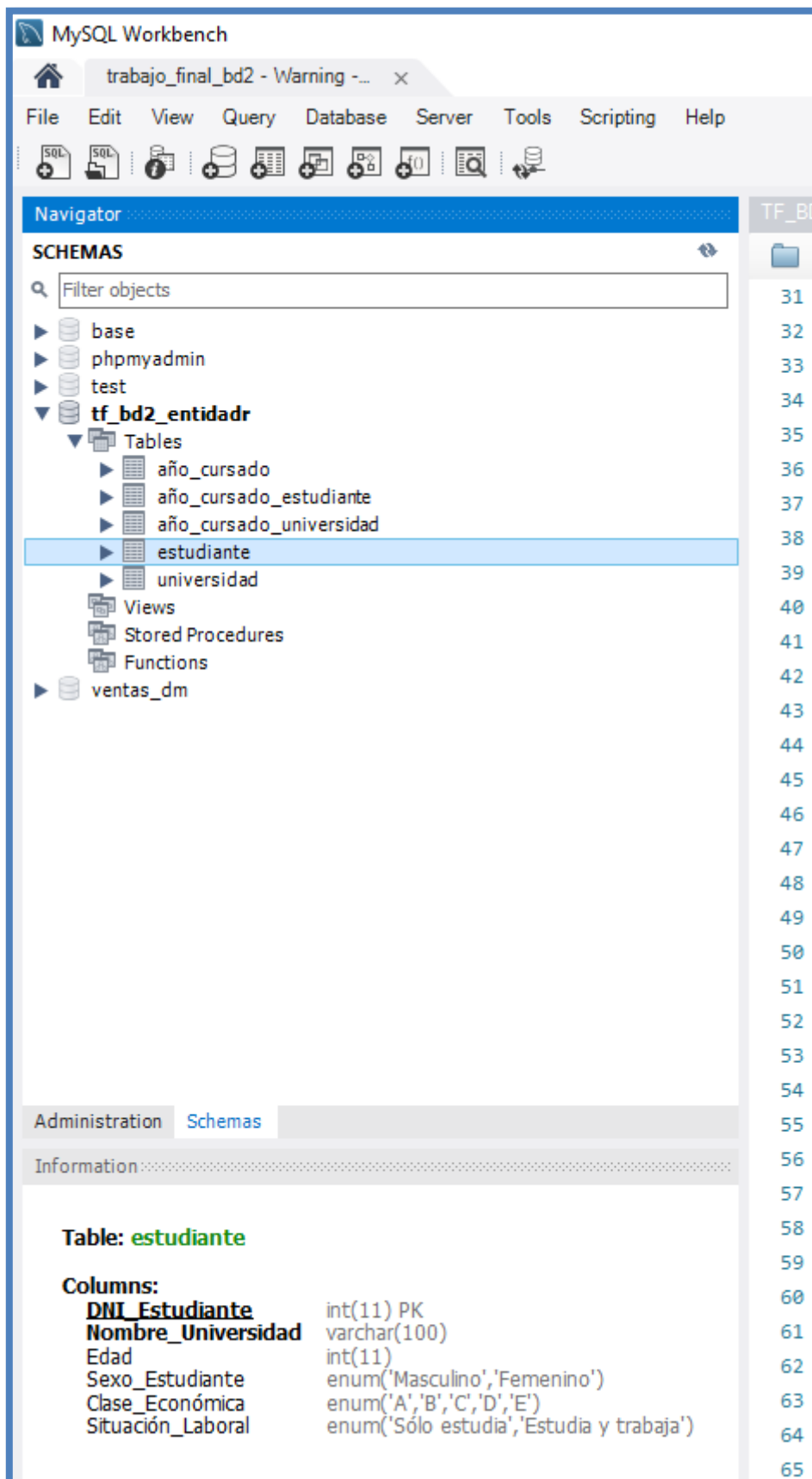
```

31  #Creamos las tabla Año-cursado
32  ● ○ CREATE TABLE IF NOT EXISTS Año_cursado(
33      Año INT NOT NULL,
34      Matriculados_Por_Año int,
35      PRIMARY KEY(Año)
36  )ENGINE=INNODB;
37
38  #Creamos las tabla Año_cursado_Estudiante
39  ● ○ CREATE TABLE IF NOT EXISTS Año_cursado_Estudiante(
40      Año int not null,
41      constraint Fk_Año_cursado
42      foreign key(Año)
43      references Año_cursado(Año),
44      DNI_Estudiante int not null,
45      constraint Fk_Estudiante
46      foreign key(DNI_Estudiante)
47      references Estudiante(DNI_Estudiante),
48      Promedio_Estudiante numeric (3,1),
49      Situación_Académica ENUM("Normal","Observado", "Suspendido"),
50      PRIMARY KEY(Año, DNI_Estudiante)
51  )ENGINE=INNODB;
52
53  #Creamos las tabla Año_cursado_Universidad
54  ● ○ CREATE TABLE IF NOT EXISTS Año_cursado_Universidad(
55      Año int not null,
56      constraint Fk_Año_cursado2
57      foreign key(Año)
58      references Año_cursado(Año),
59      Nombre_Universidad varchar(100) NOT NULL,
60      constraint Fk_Universidad2
61      foreign key(Nombre_Universidad)
62      references Universidad(Nombre_Universidad),
63      Cantidad_Profesores_Contratados int,
64      Matriculados_en_el_año int,
65      Tipo_Modalidad ENUM("Semi-presencial","Virtual"),
66      PRIMARY KEY(Año, Nombre_Universidad)
67  )ENGINE=INNODB;

```

Es importante tener en consideración el orden de creación de las variables y el tipo de formato que se le asigna, ya que de la misma manera deberá estar nuestra data que importamos.

Entidades creadas



MySQL Workbench

trabajo_final_bd2 - Warning - ...

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- base
- phpmyadmin
- test
- tf_bd2_entidadr**
 - Tables
 - año_cursado
 - año_cursado_estudiante
 - año_cursado_universidad
 - estudiante**
 - universidad
 - Views
 - Stored Procedures
 - Functions
- ventas_dm

Administration Schemas Information

Table: **estudiante**

Columns:

DNI_Estudiante	int(11) PK
Nombre_Universidad	varchar(100)
Edad	int(11)
Sexo_Estudiante	enum('Masculino', 'Femenino')
Clase_Económica	enum('A', 'B', 'C', 'D', 'E')
Situación_Laboral	enum('Sólo estudia', 'Estudia y trabaja')

Inserción de datos a las entidades:

Para poder cargar nuestros datos haremos uso del siguiente código (Se hace uso de códigos ya que al intentar hacerlo por la herramienta de importación la carga de datos era demasiada, al realizarlo por código es mucho más rápido):

```
69     ##Ahora subimos los datos
70     #Cargamos para universidad
71     • Load data infile 'C:\\Users\\Asus\\Desktop\\universidades.csv'
72     into table universidad
73     CHARACTER SET UTF8
74     fields terminated by ';'
75     optionally enclosed by '"'
76     lines terminated by '\\r\\n'
77     ignore 1 rows;
78
79     #Cargamos para año_cursado
80     • Load data infile 'C:\\Users\\Asus\\Desktop\\years.csv'
81     into table año_cursado
82     CHARACTER SET UTF8
83     fields terminated by ';'
84     optionally enclosed by '"'
85     lines terminated by '\\r\\n'
86     ignore 1 rows;
87
88     #Cargamos para año_cursado_universidad
89     • Load data infile 'C:\\Users\\Asus\\Desktop\\year_universidad.csv'
90     into table año_cursado_universidad
91     CHARACTER SET UTF8
92     fields terminated by ';'
93     optionally enclosed by '"'
94     lines terminated by '\\r\\n'
95     ignore 1 rows;
96
```



```

97      #Cargamos para estudiante
98      • SET FOREIGN_KEY_CHECKS=0;
99      • Load data infile 'C:\\Users\\Asus\\Desktop\\estudiantes.csv'
100      into table estudiante
101      CHARACTER SET UTF8
102      fields terminated by ';'
103      optionally enclosed by '"'
104      lines terminated by '\\r\\n'
105      ignore 1 rows;
106
107      #Cargamos para año_cursado_estudiante
108      • SET FOREIGN_KEY_CHECKS=0;
109      • Load data infile 'C:\\Users\\Asus\\Desktop\\year_estudiantes.csv'
110      into table año_cursado_estudiante
111      CHARACTER SET UTF8
112      fields terminated by ';'
113      optionally enclosed by '"'
114      lines terminated by '\\r\\n'
115      ignore 1 rows;
116      • SET FOREIGN_KEY_CHECKS=1;
117

```

Una vez cargada toda nuestra data, crearemos nuestro modelo estrella con el cual realizaremos nuestros indicadores KPI'S.

Generación del DATAMART

Modelo Bidimensional

El modelo que se desarrollo fue el modelo estrella.
Primero crearemos la tabla de hechos.

Tabla_hechos:

Esta entidad tendrá las siguientes variables: DNI_Estudiante, Año, Nombre_Universidad, Matriculados_en_el_año, Promedio_Estudiante, Clase_Económica, Tipo_Modalidad, Gestión, Situación_Académica y Departamento

La creación también la haremos en R en base a los datos de las entidades anteriormente ya obtenidas. Para su creación usaremos el código merge() de R que nos permite unir varias datas en base a una variable idéntica en ambas tablas, usando de base a año_estudiante el cual contiene la mayor cantidad de variables deseadas para nuestra tabla de hechos comenzaremos añadiendo de uno en uno las variables deseadas de las demás entidades.

```

#Usaremos de base a year_estudiantes
tabla_hechos <- year_estudiantes

#Para obtener: Nombre_Universidad y Clase_Económica
tabla_hechos <- merge(tabla_hechos, dplyr::select(estudiantes, DNI_Estudiante, Nombre_Universidad, Clase_Económica), by = "DNI_Estudiante")

#Para obtener: Matriculados_en_el_año y Tipo_Modalidad
tabla_hechos <- merge(tabla_hechos, dplyr::select(año_cursado_universidad[año_cursado_universidad$Matriculados_en_el_año > 0,], Año, Nombre_Universidad, Matriculados_en_el_año, Tipo_Modalidad), by = c("Año", "Nombre_Universidad"))

#Para obtener: Gestión y Departamento
universidades <- read.csv2("universidades.csv")
tabla_hechos <- merge(tabla_hechos, dplyr::select(universidades, Nombre_Universidad, Gestión, Departamento), by = "Nombre_Universidad")

write.table(tabla_hechos, "tabla_hechos.csv", fileEncoding = "UTF-8",
            quote = F, row.names = F, dec = ".", sep = ";")

```

Creación del modelo estrella:

Para la creación del modelo estrella crearemos otra base de datos (la llamaremos "tf_bd2_me"), crearemos nuevas tablas que hagan referencia a las que se usarán para los KPI'S e importaremos la data de la otra base de datos a esta (En la tabla de hechos debe ir todo lo relacionado a los KPI'S).

```

120      #Creamos la base de datos para nuestro modelo estrella
121 •    CREATE DATABASE IF NOT EXISTS tf_bd2_me;
122
123      #Seleccionar la base de datos
124 •    USE tf_bd2_me;
125
126      #Creamos las tabla Universidad2
127 •    CREATE TABLE IF NOT EXISTS Universidad2(
128          Nombre_Universidad varchar(100) NOT NULL,
129          Distrito varchar(50),
130          Provincia varchar(50),
131          PRIMARY KEY(Nombre_Universidad)
132      )ENGINE=INNODB;
133
134      #Creamos las tabla Estudiante2
135 •    CREATE TABLE IF NOT EXISTS Estudiante2(
136          DNI_Estudiante INT NOT NULL,
137          Edad int,
138          Sexo_Estudiante ENUM("Masculino","Femenino"),
139          Situación_Laboral ENUM("Sólo estudia","Estudia y trabaja"),
140          PRIMARY KEY(DNI_Estudiante)
141      )ENGINE=INNODB;
142
143      #Creamos las tabla Año-cursado2
144 •    CREATE TABLE IF NOT EXISTS Año_cursado2(
145          Año INT NOT NULL,
146          PRIMARY KEY(Año)
147      )ENGINE=INNODB;
148

```

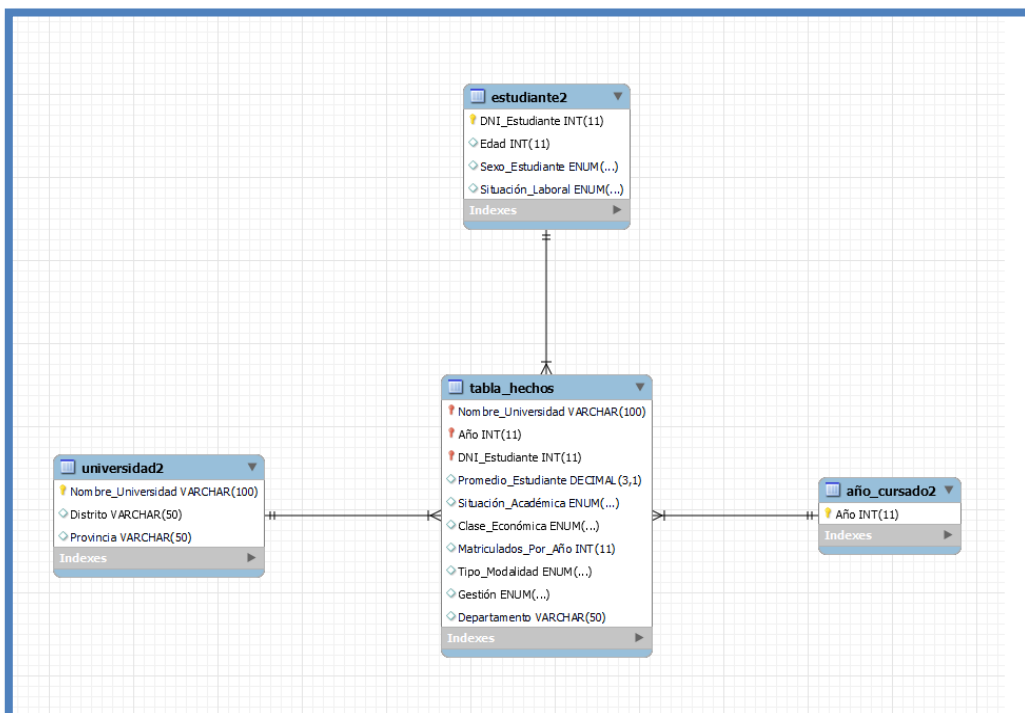
```

149 #Creamos las tabla de hechos
150 • CREATE TABLE IF NOT EXISTS Tabla_Hechos(
151     Nombre_Universidad varchar(100) not null,
152     constraint Fk_Universidad_hecho
153     foreign key(Nombre_Universidad)
154     references Universidad2(Nombre_Universidad),
155     Año int not null,
156     constraint Fk_año_cursado_hecho
157     foreign key(Año)
158     references año_cursado2(Año),
159     DNI_Estudiante int not null,
160     constraint Fk_Estudiante_hecho
161     foreign key(DNI_Estudiante)
162     references estudiante2(DNI_Estudiante),
163     Promedio_Estudiante numeric (3,1),
164     Situación_Académica ENUM("Normal","Observado", "Suspendido"),
165     Clase_Económica ENUM("A", "B", "C", "D", "E"),
166     Matriculados_Por_Año int,
167     Tipo_Modalidad ENUM("Semi-presencial","Virtual"),
168     Gestión ENUM("Público","Privado"),
169     Departamento varchar(50),
170     PRIMARY KEY(DNI_Estudiante,Año,Nombre_Universidad)
171 )ENGINE=INNODB;
172
173 #Insertamos los datos de la base de datos tfbd2_er para las entidades estudiantes2, universidad2 y año_cursado2
174 #Insert para Estudiantes2
175 • Insert into tf_bd2_me.estudiante2
176     select DNI_Estudiante,
177           Edad,
178           Sexo_Estudiante,
179           Situación_Laboral
180     from tf_bd2_entidadr2.estudiante;

```

Para la tabla de hechos sí se ha necesitado importar desde un archivo csv, es el mismo que hemos visto como lo creamos anteriormente en R.

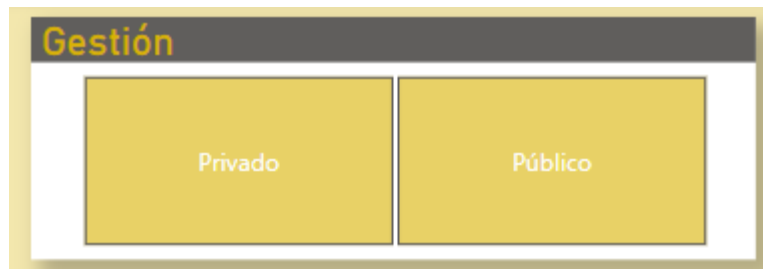
Ahora esto podríamos verlo graficado de la misma forma que hicimos con nuestro datamart anteriormente. El resultado sería el siguiente:



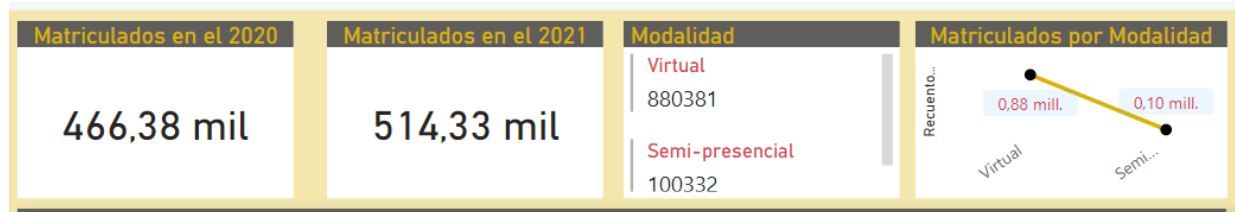
Dashboard

Con los datos ya simulados y previamente procesados procedemos a la creación de nuestro dashboard haciendo uso del software Power BI, este será de carácter informativo sobre la situación de los estudiantes universitarios, en los siguientes pasos procederemos a trabajar en base a nuestras variables simuladas:

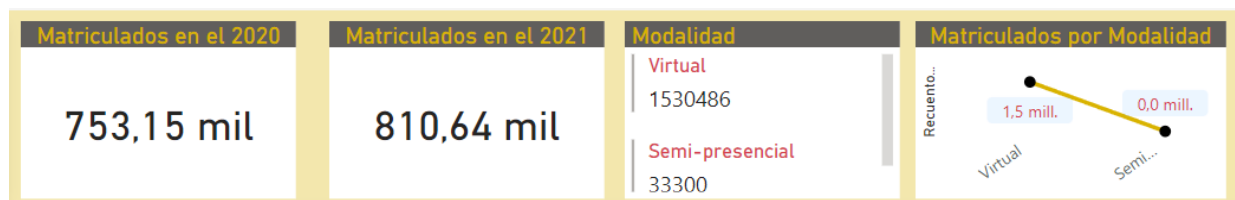
- **Filtro o llave de interacción con el total de alumnos según el tipo de gestión**



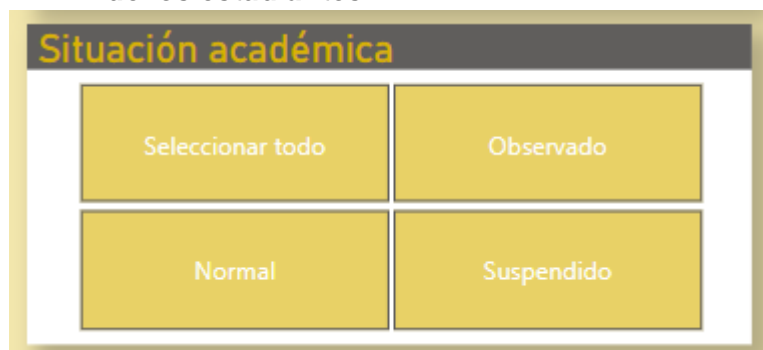
- **Total, de matriculados en universidades de gestión pública durante el periodo 2020 - 2021**



- **Total, de matriculados en universidades de gestión privada durante el periodo 2020 - 2021**



- **Filtro o llave de interacción con el total de alumnos según la situación académica de los estudiantes**



- **Filtro o llave de interacción con el total de alumnos según departamentos del Perú**

Departamento	2020	2021	Total
AMAZONAS	✗ 20173	✗ 22709	42882
ANCASH	✗ 13117	✗ 14848	27965
APURIMAC	✗ 20001	✗ 22340	42341
AREQUIPA	✗ 43180	✗ 44643	87823
AYACUCHO	✗ 26651	✗ 29737	56388
CAJAMARCA	✗ 46503	✗ 55456	101959
CALLAO	✗ 19800	✗ 22106	41906
CUSCO	✗ 46593	✗ 51687	98280
HUANCAVELICA	✗ 20011	✗ 22435	42446
HUANUCO	✗ 20108	✗ 22290	42398
ICA	✗ 36562	✗ 37087	73649
JUNIN	✗ 59657	! 66652	126309
LA LIBERTAD	✗ 16631	✗ 14860	31491
LAMBAYEQUE	✗ 26779	✗ 29784	56563
LIMA	✓ 259063	✓ 281220	540283
LORETO	✗ 26771	✗ 22200	48971
MADRE DE DIOS	✗ 6612	✗ 7435	14047
MOQUEGUA	✗ 6568	✗ 7271	13839
PIURA	✗ 13182	✗ 14793	27975
PUNO	✗ 26332	✗ 29161	55493
SAN MARTIN	✗ 6608	✗ 7405	14013
TACNA	✗ 19972	✗ 22233	42205
TUMBES	✗ 6598	✗ 7340	13938
UCAYALI	✗ 19781	✗ 22190	41971
Total	807253	877882	1685135

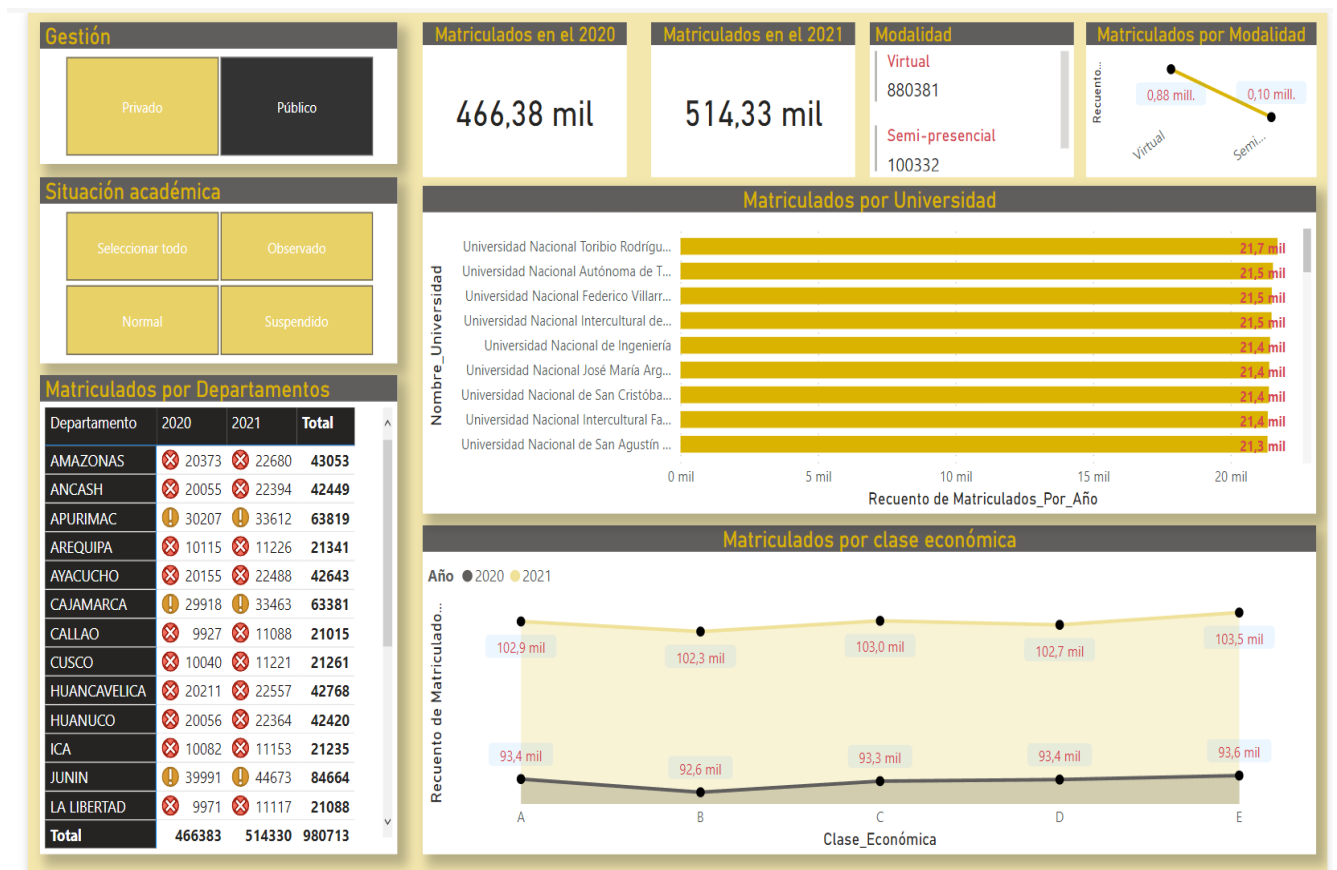
- **Filtro o llave de interacción con el total de alumnos según universidades a nivel nacional**



- **Filtro o llave de interacción con el total de alumnos según la clase económica**



- **Dashboard final**



CONCLUSIONES

- La decisión sobre el retorno a clases presenciales debe ser una decisión reflexiva como lo muestra el elevado interés mundial y nacional que se ha despertado sobre el tema. En este trabajo proponemos que en base a nuestro análisis de datos con Power BI y MySQL, podemos tener una mejor visión sobre el impacto que ha tenido sobre la población en retorno a clases presenciales.
- Se puede observar referente al nivel académico, en el año 2020, cuando la modalidad de estudio era de manera virtual al 100%, había 1596618 estudiantes en situación normal, mientras que, en el año 2021, cuando la modalidad de estudio se hizo híbrida, este número descendió a 88517 estudiantes. Es decir, el tipo de modalidad de estudio ha afectado considerablemente el rendimiento académico de los estudiantes universitarios.
- Se observa un ligero incremento de alumnos matriculados en universidades nacionales en comparación con las universidades particulares, esto respondería a la migración de estudiantes de universidades privadas a públicas que se dio debido a la crisis económica por la COVID-19. Este resultado es significativo cuando se compara el total de matriculados en universidades particulares con universidades nacionales.
- De manera positiva, se puede observar un incremento de alumnos matriculados en todas las regiones del Perú en el año 2021 en comparación con el año 2020.