Rendimiento Escolar Año 2010-2019

Proyecto Analítico de Datos

Profesor Mario Ortiz

Alumnos Johan Órdenes
Leonel Villagra

Fecha 20/01/2021

Índice

Agradecimientos a Patricio Ávalos por habernos ayudado en este proceso.

1. Fé de erratas en la documentación

En la tabla “Código de Enseñanza” en el anexo 2 (página 9) falta agregar ésta fila:

Código de Enseñanza	
COD_ENSE	**Descripción**
861	Enseñanza Media T-P Marítima Adultos (Decreto N° 152/1989)

Tabla de documentación requerida en tipo de enseñanza
Fuente: Modificación propia sobre la documentación

Los archivos tienen encodings diferentes dependiendo del año. Las letras Ñ y los acentos no se muestran correctamente o directamente no se abren los archivos si no se utiliza el específico para cada año, por ende utilizamos los siguientes:

- 2014, 2015: ‘latin’
- 2016, 2018, 2019: ‘utf-8-sig’
- otros: ‘utf-8’

Los campos que dicen ‘EN BLANCO’ no menciona nada sobre el contenido de los datos, sin embargo, hemos encontrado los siguientes:

- En el campo ‘INT_ALU’ también hay strings vacíos.
- En el campo ‘SIT_FIN’ también hay strings vacíos.

No existe la tabla comuna ni sus datos asociados.

2. Visualización general de los archivos

En esta investigación se nos pide hacer el proceso de ETL, es decir exportar, transformar y cargar los datos de los archivo correspondientes a los años 2010 a 2019 en formato .csv en los que todos los datasets contienen un aproximado de 3.5 millones de tuplas. Esto lo hicimos realizando la lectura de los datos entregados, limpieza y eliminación de los datos inservibles, y por último el almacenamiento de los datos en una base de datos en Postgres.

En la siguiente tabla se muestra la cantidad de alumnos para los años faltantes, la codificación de los archivos y las columnas eliminadas para nuestro análisis de datos.

Archivo	Encoding	Estudiantes	Columnas eliminadas
datasets/Rendimiento por estudiante 2010.csv	Utf-8	3.335.825	['SIT_FIN_R']
datasets/Rendimiento por estudiante 2011.csv	Utf-8	3.326.746	['FEC_ING_ALU', 'SIT_FIN_R']
datasets/Rendimiento por estudiante 2012.csv	Utf-8	3.308.477	['SIT_FIN_R']
datasets/Rendimiento por estudiante 2013.csv	Utf-8	3.255.518	['GD_ALU', 'COD_TIP_CUR', 'COD_REG_ALU', 'COD_RAMA', 'SIT_FIN_R']
datasets/Rendimiento por estudiante 2014.csv	latin	3.227.534	['GD_ALU', 'COD_DEPE2', 'COD_TIP_CUR', 'COD_REG_ALU', 'COD_RAMA']
datasets/Rendimiento por estudiante 2015.csv	latin	3.238.586	['COD_DEPROV_RBD', 'NOM_DEPROV_RBD', 'GD_ALU', 'COD_DEPE2', 'ESTADO_ESTAB', 'COD_TIP_CUR', 'COD_REG_ALU', 'COD_RAMA', 'SIT_FIN_R']

datasets/Rendimiento por estudiante 2016.csv	Utf-8-sig	3.226.943	['COD_DEPROV_RBD', 'NOM_DEPROV_RBD', 'COD_DEPE2', 'ESTADO_ESTAB', 'COD_TIP_CUR', 'COD_DES_CUR', 'COD_REG_ALU', 'COD_RAMA', 'SIT_FIN_R']
datasets/Rendimiento por estudiante 2017.csv	Utf-8	3.246.824	['COD_DEPROV_RBD', 'NOM_DEPROV_RBD', 'COD_DEPE2', 'ESTADO_ESTAB', 'COD_TIP_CUR', 'COD_DES_CUR', 'COD_REG_ALU', 'COD_RAMA', 'SIT_FIN_R']
datasets/Rendimiento por estudiante 2018.csv	Utf-8-sig	3.293.750	['NOM_REG_RBD_A', 'COD_DEPROV_RBD', 'NOM_DEPROV_RBD', 'COD_DEPE2', 'ESTADO_ESTAB', 'COD_TIP_CUR', 'COD_DES_CUR', 'COD_REG_ALU', 'COD_RAMA', 'SIT_FIN_R']
datasets/Rendimiento por estudiante 2019.csv	Utf-8-sig	3.328.915	['NOM_REG_RBD_A', 'COD_DEPROV_RBD', 'NOM_DEPROV_RBD', 'COD_DEPE2', 'ESTADO_ESTAB', 'COD_TIP_CUR', 'COD_DES_CUR', 'COD_REG_ALU', 'COD_RAMA', 'COD_MEN', 'SIT_FIN_R']

Tabla de información general sobre los archivos
Fuente: Modificación propia sobre la documentación

3. Eliminación general de los datos

Para visualizar los datos a modo testing decidimos optar por exportar las primeras 10 filas de todos los datasets para ir utilizándolos uno a uno y ver las posibles columnas a descartar.

Primero que todo es necesario hacer un análisis exhaustivo de los datos de las tablas para agregar a nuestra propia base de datos y para limpiar los datos en el excel, por ende decidimos por un lado ver todos los datos de las columnas que contenía cada uno de los archivos por año en formato .csv, y por otro lado visualizar las columnas que contenía dicho archivo con la librería Pandas en Python.

Columnas funcionales que vamos a utilizar y que están disponibles en todos los años a analizar:

- AGNO
- RBD
- DGV_RBD
- NOM_RBD
- COD_REG_RBD
- COD_COM_RBD
- NOM_COM_RBD
- COD_DEPE
- RURAL_RBD
- COD_ENSE
- COD_GRADO
- LET_CUR
- MRUN
- GEN_ALU
- FEC_NAC_ALU
- COD_COM_ALU
- NOM_COM_ALU
- COD_SEC
- COD_ESPE
- PROM_GRAL
- ASISTENCIA
- SIT_FIN

Columnas agregadas por decisión de analizar datos:

Columna	Razón
COD_ENSE2	A diferencia de COD_ENSE, este nos permite saber de forma agrupada qué tipo de enseñanza recibe el alumno, básica o media y es mucho más fácil y rápido de visualizar en el análisis de datos, pero esta columna sólo está disponible desde el año 2014.
COD_JOR	Pensamos eliminarlo porque los datos sólo están disponibles desde el año 2013 en adelante, pero un requerimiento de análisis en las preguntas pide saber a qué jornada asisten los alumnos.
EDAD_ALU	Pensamos eliminarlo porque la verdad la mayoría de los alumnos tiene aproximadamente la misma edad y no es relevante para el análisis final.
INT_ALU	Quizás si se quiere hacer un análisis más específico se podría obtener los datos de los alumnos con necesidades especiales, pero desde el año 2014 en adelante no se utilizan estos datos.

Tabla de columnas agregadas
Fuente: Elaboración propia sobre la documentación

Columnas inexistentes en los archivos:

Columna	Razón
LET_RBD, NUM_RBD	Dado que no pertenecen a ningún año para el análisis, sólo existen desde el año 2009 hacia atrás.

Columnas no existentes en los años 2010-2019
Fuente: Documentación asociada

Columnas a eliminar por descarte:

Columna	Razón
NOM_REG_RBD_A	Sólo aparece en los 2 últimos años y es poco significativo para el análisis de datos final.
COD_DEPROV_RBD, NOM_DEPROV_RBD	Estos datos sólo aparecen desde el año 2015 y creemos que es innecesario saber a qué departamento provincial pertenece un establecimiento.
COD_DEPE2	Tenemos COD_DEPE que es básicamente lo mismo, la diferencia es que la anterior aparece en todos los años y contiene 1 atributo más. En este caso que los datos sean agrupados, no cambia el resultado final.
ESTADO_ESTAB	Sólo aparece desde el 2015. No es necesario saber en qué estado se encuentra un establecimiento para hacer el análisis de los datos.
COD_TIP_CUR	Sólo aparece desde el 2013. No es necesario saber si un curso es simple o combinado. Además en nuestro diagrama no aparece la entidad curso.
COD_DES_CUR	Dado que solamente aplica a enseñanza media TP, es demasiado específico para un análisis general de los datos. Además los atributos de esta columna están más ligados a un establecimiento que a un curso en sí.
GD_ALU	Sólo está disponible en los años 2013 a 2015, es muy poca cantidad de datos (menos de 1/3) considerando la totalidad general de los datos para un análisis.
COD_REG_ALU	Creemos que no es necesario saber a qué región pertenece un alumno si ya tenemos los datos a qué comuna pertenece un alumno, podemos obtener también los datos mediante el establecimiento.

COD_RAMA	Los datos aparecen desde el 2013 y creemos que no es necesario saber que tipo de educación recibe un estudiante.
FEC_ING_ALU	Los datos sólo están disponibles en el año 2011 y como buenos diseñadores de bases de datos, sentimos que perjudicaría el agregarlo porque es redundante y el año de ingreso se puede repetir hasta mínimo 8 veces que es la cantidad de años que un estudiante se forma en un establecimiento educacional.
COD_MEN	Son datos muy recientes y sólo está disponible en el último año en los datos.
SIT_FIN_R	Los alumnos trasladados no tienen ponderaciones finales.

Tabla de columnas eliminadas
Fuente: elaboración propia sobre la documentación

Para nuestro análisis no hubo ninguna eliminación de filas ni datos asociados a estas, sólo columnas.

Luego de finalizar el análisis anterior, decidimos guardar los datos de todas las columnas, dado que nos sirven para hacer un análisis final a través de una serie de preguntas que nos podríamos plantear y sugeridas por el profesor, en el caso de “EDAD_ALU” podríamos ver por ejemplo la cantidad de alumnos que tenga cierta edad, digamos podríamos obtener los alumnos que no están dentro del promedio general y analizar el por qué, e “INT_ALU” y ver la cantidad de alumnos que hay, analizar y poder ayudarlos de alguna manera, dando respuestas más concretas a las preguntas solicitadas por la universidad.

4. Cambio de variables

Para la tabla “Situación Final” (Aprobado/Reprobado/Retirado/Sin Información) para el campo (Sin Información) los strings nulos y vacíos fueron cambiados por guión, ejemplo:

Código Antes	Código Después	Descripción
‘ ‘ / null	‘-’	Sin información

Cambio de variables en documentación “Situación Final”
Fuente: Elaboración propia sobre la documentación

Para la tabla “Alumno Integrado” (Sí / No / Sin Información), para el campo (Sin Información) los strings que contenían punto fueron reemplazados por 2, para conservar el valor numérico de las claves primarias, ejemplo:

Código Antes	Código Después	Descripción
‘.’ / ‘ ‘	2	Sin información

Cambio de variables en documentación“Alumno integrado”
Fuente: Elaboración propia sobre la documentación

Para las tablas “Sector Económico” y “Especialidad”, los strings vacíos fueron reemplazados por 0, para conservar el valor numérico de las claves primarias, ejemplo:

Código Antes	Código Después	Descripción
‘ ‘	0	Ciclo General / Sin Información

Cambio de variables en documentación “Sector económico” y “Especialidad”
Fuente: Elaboración propia sobre la documentación

Para el campo “Promedio General” los valores de coma fueron reemplazados por puntos para evitar errores al subir datos a PostgreSQL.

Para el campo “RUT del alumno” fueron tratados como String dado que el para año 2012, no contiene valores numéricos para esa columna.

Se creó la tabla “Comuna” que contiene los datos de todas las comunas existentes entre los años 2010-2019, obtuvimos estos datos de las columnas “COD_COM_RBD” (Comuna en donde se encuentra el establecimiento) y “COD_COM_ALU” (Comuna de donde proviene el alumno) uniéndose en una tabla en común ahorrando espacio en la base de datos existente.

No utilizamos los datos de la región del alumno porque se pueden extraer haciendo una relación cruzada entre alumno y establecimiento y obtener los datos directamente.

5. Ejecución, consumo de RAM y tiempo de compilación

Primero que todo, lo que hace el programa es generar un dataset único, lo hicimos de esta manera por las siguientes razones:

- Comunas: Los datos de las comunas no están en la documentación y por lo que pudimos rescatar, se fueron agregando comunas al pasar los años.

- Alumnos: No queremos tener los 35 millones de alumnos ingresados en nuestra base de datos, por ende se hace un descarte, porque un estudiante del 2010 puede estar también en los años siguientes, o puede que un alumno no esté en los primeros años y se ingresó después.

- Establecimientos: También hubieron establecimientos que se fueron agregando al paso de los años, es por eso que tuvimos que descartarlos desde el dataset.

Cada vez que importamos un año en particular, agregamos los datos de ese año limpiando la ram y vaciándola con cada archivo importado. Esto es altamente necesario para no exceder los límites de nuestra memoria RAM.

El programa ocupa un aproximado de 7-13 Gb de RAM libres como máximo, por ende se recomienda un mínimo de 14 Gb de RAM libres para ejecutar el código sin mayores problemas.

Sobre la complejidad algorítmica, los algoritmos que más tiempo demoran en ejecutarse es la lectura de archivos en promedio O(n) con pequeñas funciones de orden O(n^2) que afectan levemente el rendimiento general dado que son cantidades de datos mínimas, excepto para eliminar los duplicados, que se necesita realizar obligatoriamente para corroborar la información de que no exista alguno.

Intentamos primero realizar la subida de archivos con funciones predeterminadas de pandas, nos encontramos el problema que nuestro código se demoró aproximadamente 1 hora en insertar datos sólo en el año 2010, entonces estimamos que para el resto de años iba a incrementar por hasta más de 10 horas incluyendo toda la limpieza de datos para que la ram no colapsara. Por ende no realizamos testing en este caso.

Luego utilizando la misma función por defecto agregamos lo que se llama “bulk-insert” con “executemany”, que predeterminadamente por nuestras variables

tomamos 100.000 queries de inserciones y las agregando todas a la vez a modo de empaquetamiento, en vez de agregar una por una por separado.

Para esto decidimos crear una tabla que muestra los registros asociados paso a paso a continuación:

Fase	RAM	Acumulado	Real
Inicio del programa	85.99 mb	0 s	0 s
Vaciar base de datos	86.01mb	-	-
Crear tablas	86 mb	-	-
Insertar tablas estáticas	86 mb	0.36 s	36
Lectura dataset 2010	888.52 mb	11.57 s	1 s
Limpieza	1.22 gb	14.67 s	3 s
Lectura dataset 2011	2.01 gb	26.77 s	12 s
Limpieza	1.82 gb	32.61 s	6 s
Lectura dataset 2012	2.86 gb	45.13 s	13 s
Limpieza	3.45 gb	53.64 s	9 s
Lectura dataset 2013	4.43 gb	65.86 s	12 s
Limpieza	4.65 gb	76.60 s	11 s
Lectura dataset 2014	5.64 gb	88.82 s	12 s
Limpieza	5.95 gb	101.39 s	13 s
Lectura dataset 2015	7.05 gb	115.87 s	14 s
Limpieza	7.27 gb	131.15 s	15 s
Lectura dataset 2016	8.40 gb	146.34 s	15 s
Limpieza	8.58 gb	163.30 s	17 s
Lectura dataset 2017	9.72 gb	179.37 s	16 s
Limpieza	8.81 gb	216.76 s	37 s
Lectura dataset 2018	10 gb	233.24 s	16 s
Limpieza	8.01 gb	265.64 s	32 s

Lectura dataset 2019	9.26 gb	282.67 s	17 s
Limpieza	8.55 gb	324.42 s	42 s
Conversión string	10.22 gb	333.47 s	9 s
Inserción “comuna”	12.52 gb	338.91 s	5 s
Limpieza	8.48 gb	382.43 s	44 s
Inserción “establecimiento”	8.53 gb	-	-
Limpieza	6.5 gb	395.27 s	13 s
Inserción “alumno”	9.77 gb	-	0
Limpieza	7.38 gb	1209.42 s	814 s (13:56 minutos)
Inserción “notas”	5.26 gb	-	-
Limpieza final	99.25 mb	11058.32 s	9849 s (2:44 horas)

Tabla de tiempos de ejecución
Fuente: Elaboración de Equipo

En total el programa se demoró 3 horas con 3 minutos en ejecutarse.

Sabíamos de antemano que las formas de mejorar el tiempo de procesamiento del código no iban a ser tan buenas, una por la forma en que nosotros planeamos en analizar los datos, y otra porque python no es el lenguaje más rápido en ejecutarse.

Probamos varias librerías, pero no pudimos obtener los resultados esperados.

Uno de los intentos de mejora fue utilizar la librería Modin de pandas (Con el motor Dask), que de alguna manera hace que los dataframes reduzcan considerablemente el tamaño y en vez de realizar la lectura normal, la hace de forma paralela tomando al menos 4 núcleos del procesador.

- Ventajas: Esta librería cumplía con nuestros requisitos, y se notaba un ligero cambio en la carga de los archivos.

- Problemas: La función drop duplicates no funciona como debería, y cuando lo hace colapsa la memoria RAM dejando congelado completamente el programa.

Otro de los intentos fue StringIO para hacer los bulk insert, pero el problema que tuvimos es que en algunos casos toma valores enteros como float y aún así casteándolos no solucionaba el problema.

Luego decidimos dejar los datos como los teníamos y generar los gráficos correspondientes, ya que perdimos 2 días completos tratando de solucionar este problema.

El archivo SQL exportado tiene un peso final de 287mb, lo cual es excelente comparado con el peso de la suma de todos los datasets que equivale aproximadamente a 4.4gb en total. Es aquí en donde nos damos cuenta lo bueno de utilizar apropiadamente las bases de datos relacionales multidimensionales.

Por nuestra parte nos dimos cuenta que realizaron una separación cada año porque el costo de las consultas cuando son muchos datos es bastante demoroso y para sistemas gubernamentales, no puede existir este tipo de problemas.

6. Esquema de bases de datos final y documentación

Para transformar los datos recopilamos decidimos crear nuestro propio diagrama de entidad relacionamiento detallando la mayoría de los datos contenidos en el PDF:

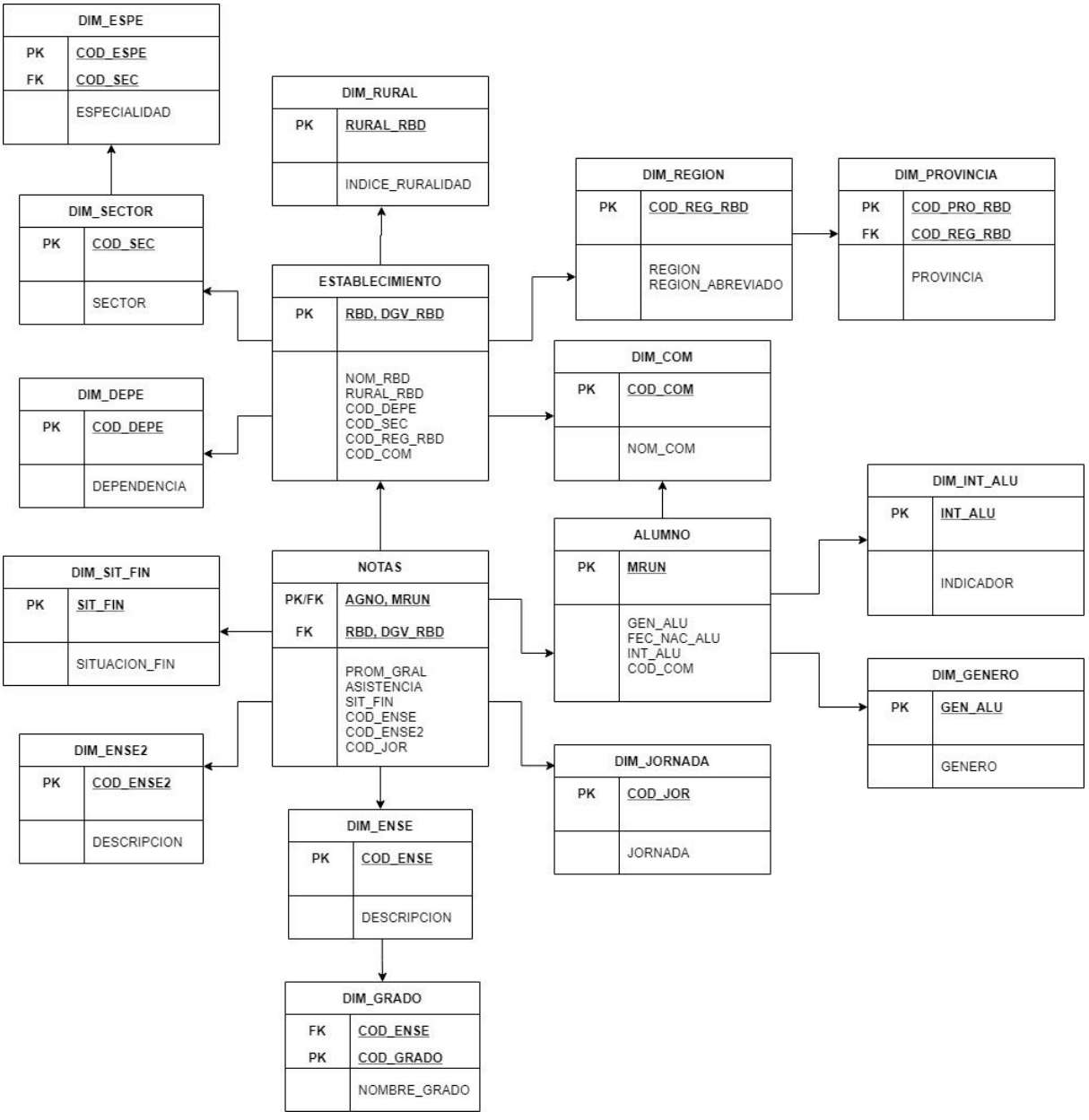


Diagrama de bases de datos multidimensional
Fuente: Elaboración de Equipo (draw.io)

Las siguientes tablas fueron agregadas estáticamente, tal cual como se muestra en la documentación y sin necesidad de filtrar el dataframe, dado que la inserción se hacía de manera mucho más rápida:

Nombre de la tabla	Cantidad de Datos	Descripción
DIM_RURAL	2	Contiene los datos de los establecimientos si es rural o no.
DIM_SECTOR	20	Contiene los datos del sector económico a los que pertenece un establecimiento.
DIM_ESPE	72	Contiene los datos de las especialidades de los sectores económicos de un establecimiento si es que existen.
DIM_DEPE	6	Contiene los datos de las dependencias de un establecimiento.
DIM_PROVINCIA	57	Contiene los datos de las provincias de donde se ubican todos los establecimientos, si es que existe la región.
DIM_SIT_FIN	4	Contiene los datos de la situación final de los alumnos.
DIM_ENSE	33	Contiene los datos de la enseñanza de los alumnos.
DIM_ENSE2	6	Contiene los datos de la enseñanza de los alumnos reducida.
DIM_GRADO	111	Contiene los datos del grado del alumno si es que existe la enseñanza.
DIM_JORNADA	4	Contiene los datos de la jornada en la que se asistió al curso.
DIM_INT_ALU	3	Contiene los datos de los alumnos que poseen alguna discapacidad.
DIM_GENERO	3	Contiene los datos del género de los alumnos.

Glosario de atributos y tablas estáticas

Fuente: Elaboración de equipo basada en documentación

Luego creamos una tabla obtenida por filtros luego de analizar los datasets, podemos deducir que no aparece en la documentación por la cantidad de datos que posee:

Nombre de la tabla	Cantidad de Datos	Descripción
DIM_COM	372	Comunas en donde se encuentran los establecimientos y donde provienen los alumnos

Glosario de tabla “Comuna”

Fuente: Elaboración de equipo basada en el dataset

Finalmente tenemos las tablas generales en donde están todas las referencias hacia las demás tablas y son las que contienen mayor cantidad de datos:

Nombre de la tabla	Cantidad de Datos	Descripción
ESTABLECIMIENTO	10.418	Contiene los datos de todos los establecimientos a lo largo del país.
ALUMNO	5.873.121	Contiene los datos de todos los alumnos registrados en los años 2010-2019.
NOTAS	30.877.781	Contiene los datos de todas las notas de los alumnos registrados en los años 2010-2019.

Glosario de tablas principales

Fuente: Elaboración de equipo basada en el dataset

Como podemos notar la tabla notas es la que contiene la mayor cantidad de datos, esto es de esperar porque son los datos que más nos interesan de la base de datos y la capacidad de analizarlos se hace muy sencilla.

7. Análisis final de los datos

Luego de ya tener nuestra base de datos completa, primero que todo comenzamos a plantearnos preguntas acerca de nuestro objetivo. ¿De qué nos sirven estos datos? ¿Qué conclusiones podemos obtener de estos? Entre otras.

Es por eso que decidimos implementar esta solución utilizando Power BI, herramienta indispensable para analizar todo de manera eficiente, ordenada y generar reportes sin mayores problemas, tratando que los reportes se expliquen por sí solos.

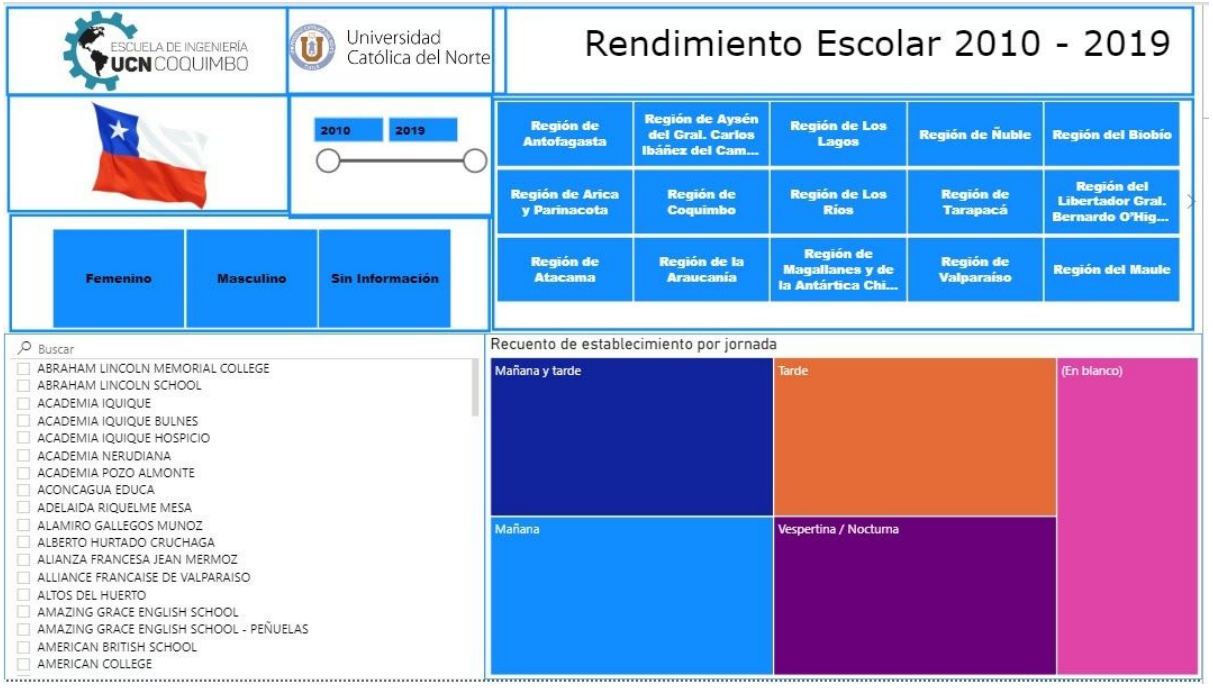
Promedio de notas en establecimientos Municipales y Particulares
Fuente: Elaboración de Equipo (Power BI)

Partiremos analizando el promedio de notas en todos los establecimientos a lo largo de los años. Estos datos nos sirven para saber si es que ha habido un aumento o disminución en el transcurso de los años, en los que podemos ver que efectivamente ha habido una disminución en sobre la “Generación Millennial” y “ Generación Z”.

Podemos deducir que quizás sucede esto por el aumento de distracciones como dispositivos móviles, computadoras, entre otros, aunque en realidad no existen datos certeros como para realizar una medición concreta.

Estos datos les sirven bastante al gobierno, para saber en qué situaciones aumentar el promedio nacional, en qué regiones ocurren e tratando de alguna manera ir mejorando la calidad de educación de los establecimientos educacionales.

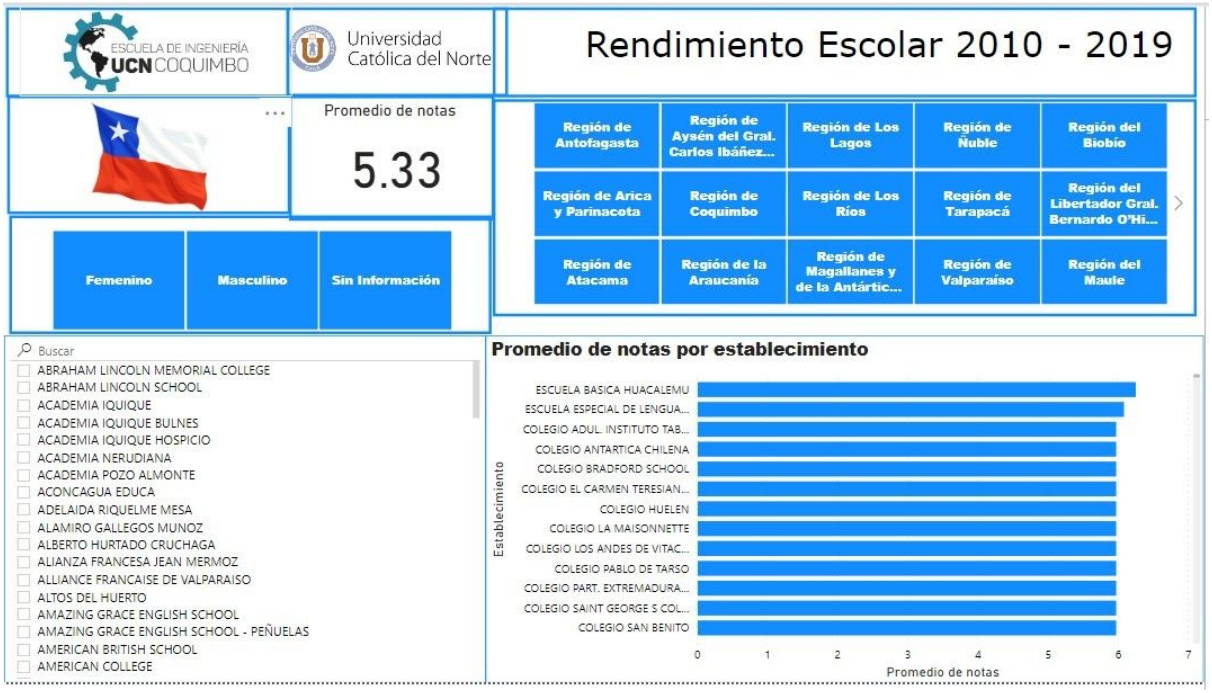
En general hemos detectado que la baja asistencia a clases empeora el rendimiento de los alumnos y que los acontecimientos históricos en el caso de la crisis social que se vivió en Chile a finales del 2019 refleja un poco el cambio en el rendimiento también.



Jornada de asistencia de alumnos
Fuente: Elaboración de Equipo (Power BI)

Luego tenemos la jornada de asistencia de los alumnos a estos establecimientos. Esto predice un estimativo a las horas en que un alumno ingresa y egresa de clases. Por ejemplo para colegios que se encuentran en la zona sur del país, las clases suelen ser durante el mediodía por temas climáticos en los cuales puede suceder que llueva durante la mañana o por la noche que es lo más común que suceda.

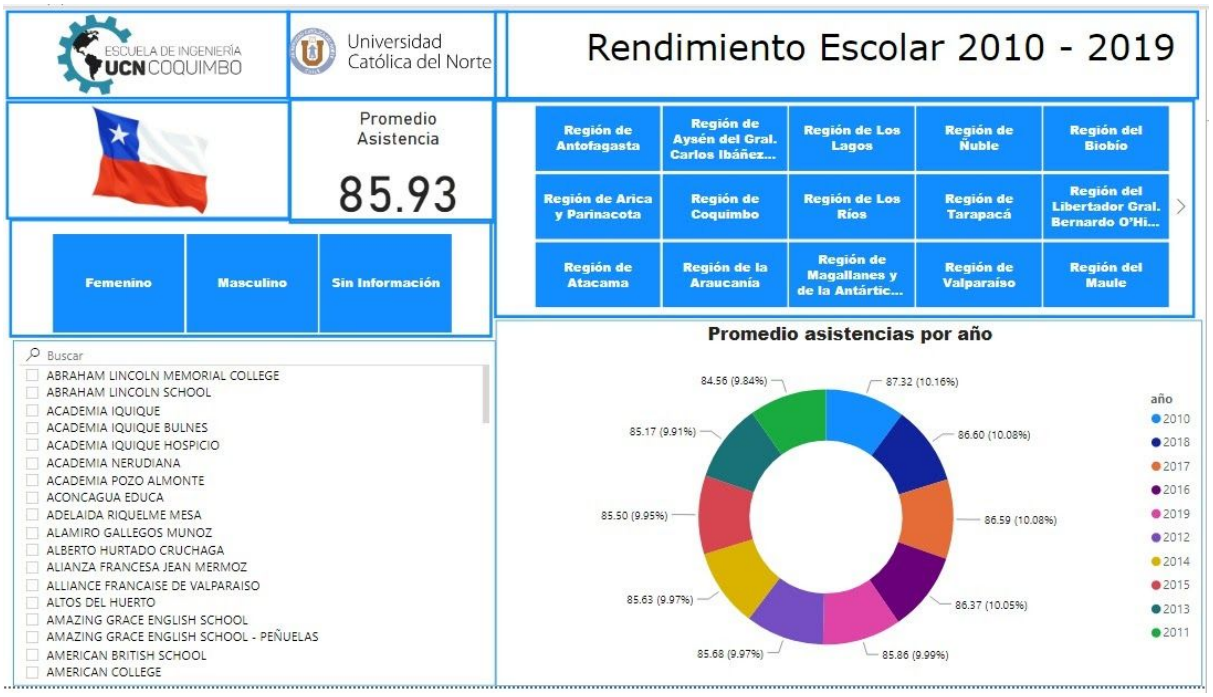
También podemos ver como cambian las jornadas con el paso de los años y si se han ido adaptando a nuestra teoría.



Promedio de notas general por establecimiento

Fuente: Elaboración de Equipo (Power BI)

Como vimos anteriormente más a detalle aquí es básicamente lo mismo pero enfocado a cualquier tipo de establecimiento sin importar la dependencia a las cuales pertenecen.



Promedio de asistencia por establecimiento
Fuente: Elaboración de Equipo (Power BI)

Como vimos anteriormente más a detalle aquí es básicamente lo mismo pero enfocado a cualquier tipo de establecimiento sin importar la dependencia a las cuales pertenecen.