# Tarea

**Entrega:** 15 de septiembre del 2021, 23:59 hrs.

## Contexto del problema

El Centro de Perfeccionamiento, Experimentación e Investigaciones Pedagógicas (en adelante CPEIP), parte del Ministerio de Educación, busca obtener de forma automatizada los docentes a los que les corresponde recibir una asignación por ejercer en establecimientos con un porcentaje de estudiantes prioritarios mayor o igual al 60%[1]. Para ello, cuentan con dos bases de datos primordiales:

- **Cargos docentes (2020):** Esta base cuenta con la información sobre los contratos de los docentes en distintos establecimientos educacionales en el año 2020. Esta cuenta tanto con información del docente en sí (título, especialidad, entre otras características) como del establecimiento en el que es contratado (región, ciudad, tipo de establecimiento, entre otras características).

- **Concentración de estudiantes prioritarios por establecimiento (2020):** Esta base cuenta con el porcentaje de estudiantes prioritarios por establecimiento en el año 2020. Solo cuenta con la información del RBD del establecimiento (número entero que lo identifica de forma única en el sistema educacional chileno) y el porcentaje de estudiantes prioritarios **(número entero en el rango 0-100)**.

Desde el CPEIP trataron de llevar a cabo la creación de una vista que resumiera toda esta información, no obstante, sus servidores no pudieron procesar tal volumen de datos, por lo que decidieron optar por un proveedor de la nube para poder desagregar la base y crear la vista con la que podrán determinar a qué profesores les corresponde una asignación y a cuáles no. Para llevar a cabo esto, necesitan a un grupo de expertos en herramientas de la nube, por lo cuál les envían la solicitud directamente a ustedes para llevar a cabo esta tarea.

## Desarrollo

Su equipo debe realizar este proyecto utilizando los servicios de la nube de Amazon, Amazon Web Services (en adelante AWS). Para esto, deberá cumplir con los siguientes criterios.

[1] Fuente: https://www.cpeip.cl/carrera-docente-asignaciones/

## 1. Almacenamiento

De partida, debe almacenar las dos bases de datos provistas por el CPEIP en uno o más *buckets* en S3. Si bien el carácter de los datos es público, el CPEIP les indica que necesitan que nadie más que ustedes tengan acceso a estos archivos dentro de los servicios del proveedor, por lo que los *buckets* deben ser de carácter **privado**.

Por otra parte, al CPEIP le gustaría que crearan una base de datos **privada** en la que puedan almacenar sus datos de forma desagregada con las siguientes entidades y relaciones:

- **Entidades**[2]
  - **Docente**: Docentes que realizan clases en el año escolar 2020. Poseen un MRUN único que los identifica, *i.e.* su RUT enmascarado, de forma que no se filtre su identidad. Al CPEIP le gustaría que esta entidad, además del MRUN, contenga en **campos de texto** los siguientes datos: título, tipo de título y especialidad. También desean que posea en un campo de número entero su año de titulación.
  - **Establecimiento:** Establecimientos en los que los docentes realizan clases en el año escolar 2020. Poseen un RBD único que los identifica. Además de este identificador, el CPEIP quiere tener en distintos **campos de texto** los siguientes datos: Región, departamento provincial, comuna y tipo de dependencia. Además, desean agregar a esta tabla un campo de tipo decimal que incluya el porcentaje de alumnos prioritarios para el año escolar 2020.
- **Relaciones**
  - **HaceClasesEn:** Relación entre las entidades “Docente” y “Establecimiento”. Simplemente posee una tabla con una llave foránea de la entidad “Docente” y una llave foránea de la entidad “Establecimiento”.

Tiene completa libertad con respecto al nombre de las columnas para los atributos solicitados, no obstante, el CPEIP le solicita que sean lo suficientemente claros para que lo puedan entender. Por otra parte, puede crear cómodamente la base de datos dentro de una instancia RDS. No obstante, el CPEIP da un pago extra si la creación de la vista se hace rápidamente, por lo que puede considerar el uso de un ***cluster*** **Redshift para un premio al final del proyecto**.

[2] **Importante:** existen algunas columnas de atributos repetidas con un sufijo de número. Por ejemplo, COD_DEPE y COD_DEPE2. En estos casos, siempre optaremos por la primera columna.

## 2. Procesamiento

Si bien al CPEIP no le interesa la forma en la que lleve a cabo el procesamiento de sus datos, el jefe de su equipo, *Hernán Canteras*, indica que deben llevar a cabo este procesamiento a través de **AWS Lambda**. En particular, le gustaría lo siguiente:

- Al subir el archivo CSV de cargos docentes, que se generen las tablas de entidad y relación automáticamente, desagregando los datos. Dado que no se tienen porcentajes de estudiantes prioritarios por estudiante en ese momento, que se establezca como valor por defecto 0 para este valor en los atributos de la entidad "Establecimiento".

- Al subir el archivo CSV de porcentajes de alumnos prioritarios, que se agregue en la tabla de entidad "Establecimiento" el atributo de porcentaje de alumnos prioritarios en formato decimal.

Lo anterior se puede llevar a cabo en dos funciones Lambda distintas, pero es importante que se ejecuten con la subida del archivo. Una vez ingresados los datos, debe generar una vista que contenga la siguiente información:

- MRUN del docente.
- Especialidad del docente.
- Atributo *booleano* (valor de verdad) que indique si el docente recibe o no asignaciones por porcentaje de alumnos prioritarios mayor a 60% en **alguno** de los establecimientos en los que imparte clases.

Esto puede llevarse a cabo en una función Lambda separada gatillada manualmente por su equipo, o bien en la función Lambda encargada de subir los datos de porcentajes de alumnos prioritarios. Su jefe le guiña el ojo y le indica que si hace esto último, tendrá una **bonificación al final del proyecto**.

**3. Disposición del resultado**

Al CPEIP le gustaría tener acceso directo a la vista final generada por usted. No obstante, quieren resguardar esto al máximo, por lo que le solicita a su equipo que le de acceso a través de un **servidor**. Un miembro del equipo de CPEIP conoce lo básico de SSH para conectarse con una llave privada, así como también ejecutar un *script* en Python que imprima los resultados. No obstante, no tiene el conocimiento suficiente para hacer su propio *script* con la consulta a la vista de la base de datos, por lo que su equipo debe facilitárselo.

**4. Seguridad**

El jefe *Canteras*, antes de darle el visto bueno para partir, le dice que un aspecto fundamental en el proyecto es que todo se mantenga **seguro y privado**. Particularmente, le dice que debe asegurar lo siguiente:

- Que no exista **ningún** recurso que tenga acceso público.
- Que **todos** los recursos se encuentren dentro de una VPC.
- Que cada recurso al que puedan asociarse grupos de seguridad tengan un **grupo de seguridad único, distinto al *default***.
- Que los grupos de seguridad entreguen el **mínimo de accesos necesario** para la ejecución de su proyecto.

**Formalidades**

Los equipos deben ser de tres a cuatro integrantes, donde **al menos uno** posea una cuenta AWS de capa gratuita. Deben llevar a cabo el desarrollo de toda la tarea en esta cuenta. Para la entrega, se habilitará un buzón con plazo hasta el **miércoles 15 de septiembre a las 23:59**. En este buzón, debe adjuntar tres archivos:

- Archivo .pem para acceder al servidor EC2 desde el que se consultará la vista generada.
- Archivo .csv con clave y secreto de acceso de usuario para revisión de sus recursos creados. **Es importante que este usuario tenga los permisos mínimos necesarios para ver los recursos, no para manipularlos**.
- Archivo “README.txt” con la siguiente información:
  - Nombre de los/as integrantes.
  - *Endpoint* del servidor EC2 para realizar la conexión.
  - Nombre del *script* en Python para obtener las filas de la vista.

**Distribución de puntaje**

La tarea cuenta con un total de 6 puntos y la fórmula para obtener la nota es la siguiente:

$$N = ptje + 1$$

Para transparentar la evaluación y facilitar la priorización de los objetivos a cumplir, a continuación les entregamos la distribución de puntaje.

- **Almacenamiento**
  - Creación de *buckets* S3: **0.5 ptos.**
  - Creación de instancia RDS: **0.5 ptos.**
    - Creación y uso de *cluster* Redshift en reemplazo de RDS: **+0.3 ptos.**
- **Procesamiento**
  - Creación de funciones Lambda: **0.3 ptos.**
  - Adición de *triggers* con archivos subidos a los *buckets* creados: **0.3 ptos.**
  - Creación de tablas de entidad y relación a partir del archivo CSV de cargos docentes: **1 pto.**
  - Adición de campo de porcentajes de alumnos prioritarios a partir de archivo CSV: **0.5 ptos.**
  - Creación de vista final de pago de asignaciones a docentes: **0.5 ptos.**
    - Creación de la vista automática posterior a la inserción de porcentajes de alumnos prioritarios: **+0.3 ptos.**
- **Disposición del resultado**
  - Creación de instancia EC2: **0.3 ptos.**
  - Creación de *script* que imprime resultados de la vista: **0.3 ptos.**
- **Seguridad**
  - Ningún recurso con acceso público: **0.4 ptos.**
  - Todos los recursos dentro de la misma VPC (salvo por los *buckets*): **0.4 ptos.**
  - Grupo de seguridad único por recurso: **0.4 ptos.**
  - Conectividad asegurada entre recursos a partir de la mínima cantidad de reglas de entrada necesarias por grupo: **0.6 ptos.**

Los puntajes que incluyan un + en la numeración son **bonificaciones**. Si los llevan a cabo, esta bonificación se aplicará a su nota. Si tienen el puntaje máximo sin bonificaciones, el puntaje restante irá a complementar las notas de sus controles de **forma proporcional**. Por ejemplo, si obtienen 0.2 puntos de bonificación (2 décimas), esto equivale a 0.6 puntos (6 décimas) en los controles.

**Bases y archivos a descargar**

En los siguientes enlaces podrá descargar todas las bases involucradas.

- Cargos docentes 2020
- Descripción de la base de datos de cargos docentes
- Concentración de prioritarios por establecimiento 2020

Por otra parte, como recordarás de las clases, no podemos ejecutar la librería **psycopg2** en Lambda sin incluirla en un archivo .zip. Por lo tanto, aquí les facilitamos la carpeta con la versión de la librería que funciona correctamente en Lambda con Python 3.7. Debes comprimir en un único archivo .zip la carpeta "package" y el *script* que ejecutará la función.

**Ayuda adicional**

En el siguiente Google Colab encontrarás algunas casillas con *scripts* que pueden ser de utilidad para el desarrollo de la tarea. Se irán agregando más *scripts* a partir de las dudas que vayan surgiendo.

**Consultas**

Las dudas que tengan de la tarea las deben plantear en el foro correspondiente. Para consultas que involucren compartir información confidencial de su cuenta o desarrollo, contactar directamente a glcontreras@uc.cl con copia a jabecerra@uc.cl y varojas@uc.cl.

Por otra parte, en la sesión del miércoles 1 de septiembre se realizará una revisión profunda del enunciado. Si además de lo anterior requieren de más sesiones virtuales de consultas, deben escribir un correo electrónico al cuerpo docente en nombre del grupo para formalizar la petición. Se buscará un día y horario a convenir para todos/as, además de compartir la grabación de la sesión.