

Armanno, Matias Nahuel

Automatización de análisis de accesos y verificación SOD en aplicativos

2021

Instituto: Ingeniería y Agronomía

Carrera: Ingeniería en Informática



Esta obra está bajo una Licencia Creative Commons Argentina.
Atribución – no comercial – sin obra derivada 4.0
<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Documento descargado de RID - UNAJ Repositorio Institucional Digital de la Universidad Nacional Arturo Jauretche

Cita recomendada:

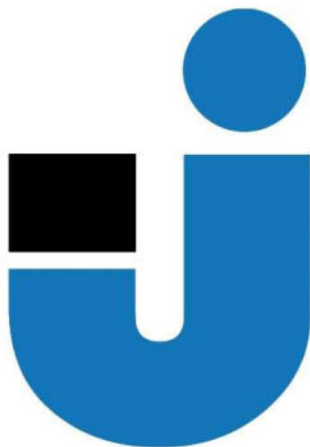
Armanno, M. N. (2021) *Automatización de análisis de accesos y verificación SOD en aplicativos* [Informe de la práctica Profesional Supervisada] Universidad Nacional Arturo Jauretche

Disponible en RID - UNAJ Repositorio Institucional Digital UNAJ <https://biblioteca.unaj.edu.ar/rid-unaj-repositorio-institucional-digital-unaj>

Universidad Nacional Arturo
Jauretche

Instituto de Ingeniería y
Agronomía

Ingeniería en Informática



**TRABAJO FINAL DE LA PRÁCTICA
PROFESIONAL SUPERVISADA**

*Automatización de análisis de accesos y
verificación SOD en aplicativos.*

Estudiante:

Matias Nahuel Armanno

Tutores:

Lic. Paula Bein

Dr. Ing. Martin Morales

Ing. Mariano Guastavino

Buenos Aires, 2021

PRÁCTICA PROFESIONAL SUPERVISADA (PPS)
Automatización de análisis de accesos y verificación SOD en aplicativos.
Informe Programa de Actividades

DATOS DEL ESTUDIANTE

Apellido y Nombres: Armanno Matías Nahuel

DNI: 37861674

Nº de Legajo: 7237

Correo electrónico: matiasnarmanno@gmail.com

Cantidad de materias aprobadas al comienzo de la PPS: 43

PPS enmarcada en artículo (4 ó 7) de la Resolución (CS) 103/16. (en caso de ser artículo 7 aclarar en cuál de las dos alternativas posibles se encuadra): 7 (B)

DOCENTE SUPERVISOR

Apellido y Nombres: Dr. Ing. Martín Morales

Correo electrónico: martin.morales@unaj.edu.ar

Apellido y Nombres: Ing. Mariano Guastavino

Correo electrónico: guastavino.mariano@gmail.com

**DOCENTE TUTOR DEL TALLER DE APOYO A LA PRODUCCIÓN DE TEXTOS ACADÉMICOS
DE LA UNAJ**

Apellido y Nombres: Lic. Paula Bein

Correo electrónico: paula.bein@gmail.com

DATOS DE LA ORGANIZACIÓN DONDE SE REALIZA LA PPS

Nombre o Razón Social: DESPEGAR COM AR SA

Dirección: DESPEGAR COM AR SA

Teléfono: 0810-810-1010

Sector: Turismo y Tecnología

TUTOR DE LA ORGANIZACIONAL

Apellido y Nombres: Ing. Luis Ríos

Correo electrónico: luis.rios@despegar.com

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

FIRMA DEL COORDINADOR DE LA CARRERA

Índice

2. Introducción

2.1.1 Marco Teórico

2.1.2 Justificación del trabajo

2.1.2.1 Necesidad de Ejecución de Proyecto

3. Desarrollo

3.1 Relevamiento del proyecto

3.2.1 Definición Funcional

3.2.2 Requerimiento Técnico

3.3 Herramientas

3.3.1 Python

3.3.2 MySQL

3.3.3 Oracle

3.3.4 Powershell

3.3.5 SQLite

3.4 Librerías

3.4.1 OS

3.4.2 CVS

3.4.3 Google

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

3.4.4 PANDAS

3.4.6 Sqlalchemy

3.4.7 Cx_Oracle

3.5 Metodología de Trabajo: Cascada

3.6 Arquitectura de la automatización

3.7 Generación de Entorno de Trabajo

3.7.1 Obtención de reportes de accesos a aplicativos

3.7.2 Obtención de nómina de empleado

3.7.3 Obtención de Políticas de Accesos y SOD

3.7.4 Carga de Reportes a Base de Datos

3.7.5 Ejecución de verificación de Accesos y SOD

3.7.6 Disponibilización de Reportes

3.7.7 Manejo de Errores

3.7.8 Orquestación de ejecución de Scripts

4. Conclusión

4.1 Resultado, posibles mejoras y ampliaciones

4.2 Reflexión sobre las prácticas profesionales como espacio de formación

5. ANEXO

5.1 Bibliografía

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

2. Introducción

Esta práctica profesional supervisada (PPS) tiene como propósito presentar el plan de trabajo para el desarrollo de automatismos que permitieron realizar la consolidación de los accesos a los aplicativos transaccionales con los que cuenta la empresa Despegar en un repositorio único. Sumado a la confección periódica de reportes que definan la correcta asignación de dichos permisos a los usuarios de la compañía. Por último, la disponibilización de un reporte que da información de todos los casos que generan incompatibilidades a nivel de segregación de funciones entre roles/permisos y aplicativos.

A grandes rasgos, se generaron diversas conexiones sobre aplicativos para obtener información sobre los roles y permisos que se encuentran configurados en los mismos, para de esta manera resguardar dicha información en una única base de datos que permita corroborar cuáles son todos los accesos que cuenta un empleado sobre todos los aplicativos de la compañía.

A partir de allí, teniendo como base una política que defina qué puesto en nómina debe tener un usuario para poder utilizar un permiso o rol específico, se realizaron diversos cruces entre los datos obtenidos y se generaron reportes donde se especifican los accesos brindados de manera correcta e incorrecta. A su vez, se dejó disponible un reporte periódico que especifica cuáles son los permisos que generan inconvenientes de segregación de funciones entre roles/permisos (es decir, la asignación de varios permisos/roles que le permita a un único usuario manipular todas las fases de una transacción sin ningún tipo de supervisión, como por ejemplo facturar y emitir una orden de compra), para que de esta manera puedan ser

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

detectados y reportados del mismo modo que se realiza con la verificación de accesos.

2.1.1 Marco Teórico:

En primer lugar, es necesario definir tres conceptos que se encuentran relacionados a la necesidad de surgimiento y desarrollo del proyecto, siendo estos “auditoría”, “Ley SOX” y “Automatización”

El concepto de Auditoría se encuentra asociado a una serie de métodos de investigación y análisis con el objetivo de producir la revisión y evaluación profunda de la gestión efectuada. Esta actividad está sujeta a relevamientos de procesos de Empresas o Compañías, los cuales son unidades económico-social integradas por elementos humanos, materiales y técnicos, que tienen el objetivo de obtener utilidades a través de su participación en el mercado de bienes y servicios.

Seguidamente, la compañía en la actualidad se encuentra en proceso de certificación SOX¹, lo cual lleva a la obligatoriedad de la implementación de diversos controles que permitan asegurar la integridad de la información financiera generada por la misma, teniendo como base las normas de contabilidad, control interno, gobierno corporativo, independencia de las auditorías y la implementación de sanciones por delitos financieros.

La forma de asegurar dicha integridad de información es mediante la eliminación de tareas manuales que conlleven a la manipulación de información. Para lograrlo es

¹ La ley Sabaranes-Oxley (SOX) regula las funciones financieras contables y de auditoría, además penaliza, en una forma severa, el crimen corporativo y de cuello blanco.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

necesario implementar procesos automáticos, los cuales permitan a las organizaciones diseñar, ejecutar, observar, supervisar y mejorar continuamente los la relación entre áreas de negocio. Entre las ventajas que produce la implementación de automatismos de procesos, se puede mencionar:

- Reducción en el ciclo de producción, mejorando la carga del equipo y disminuyendo los recursos de las acciones.
- Seguimiento y control de resultados en tiempo real, tanto en el aspecto general como de forma detallada, facilitando la toma de decisiones.
- Trazabilidad de los pasos ejecutados en el proceso.
- La automatización de procesos puede vincular las bases de datos de todos los sectores de la empresa (por ejemplo: contabilidad, recursos humanos, inventario, ventas y representantes regionales) en un sistema de información común. Esto reduce el número de errores, evita aquellos relacionados con la comunicación y aumenta la velocidad del procesamiento de datos.
- Implementación de indicadores de rendimiento de los negocios para su medición.
- Generación de informes, en cualquier momento, consolidados con indicadores (documentados) sumamente útiles para la gestión.
- Identificación y remediación los puntos de ineficiencia, desperdicios y mala aplicación de los recursos.
- Mejora en flujos de trabajo para eliminar errores e inconsistencias del proceso.
- Aumentar de manera significativa de la velocidad de ejecución de tareas
- Eliminar cantidad de horas insumidas entre las actividades.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

- Permitir la integración de diferentes sistemas de diferentes plataformas sin la necesidad de que el empleado tenga que especializarse para que los utilice.
- Estandarización de la operación

2.1.2 Justificación del trabajo

Las aplicaciones productivas -de las cuales depende la continuidad del negocio de las compañías- deben permanecer correctamente administradas y contar con controles transaccionales establecidos, ya que si no es posible generar riesgos de fraude contable que pueden desencadenar en pérdida millones de dólares y el prestigio para algunas compañías. La mejor opción a la hora de brindar seguridad a un sistema informático es prevenir que los riesgos innatos de las aplicaciones y sistemas operativos se transformen en una debilidad, y ésta en un riesgo materializado.

Las auditorías realizan análisis de los riesgos de las compañías e implementan controles que logran mitigar la materialidad de los mismos. Asimismo, los integrantes del equipo son los responsables de determinar nuevos tipos riesgos en el negocio y sus aplicaciones. Al automatizar un análisis de administración de aplicaciones, se optimizan los costos y la seguridad en los mismos reduciendo así las posibilidades de error humano. Adicionalmente es posible reducir la cantidad de horas consumidas por recurso humano y el esfuerzo necesario para detectar riesgos y mitigarlos.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

Un caso que puede ejemplificar lo mencionado anteriormente es un usuario con permisos acumulados que pueda ejecutar acciones en un aplicativo sin una debida segregación, donde se generaría el riesgo de acumulación de permisos, de esta manera sería posible saltar cualquier procesos de aprobaciones (tal es el caso de la ejecución de una compra) y se perdería la trazabilidad de una acción que pueda desencadenar en la erogación de millones de dólares para la compañía. Para este riesgo se pueden generar distintos tipos de controles o mitigantes que no permitan la materialización del mismo, como por ejemplo, la definición de la obligatoriedad de aprobadores por montos para la ejecución de una compra. Otro mitigante utilizado es definir un esquema de permisos segregados de un aplicativo y, a su vez, asociar un “dueño” para cada uno que de su consentimiento a la hora otorgamiento de los mismos a demás empleados de la entidad.

En la actualidad, debido a la naturaleza de la entidad, donde las aplicaciones en su mayoría son desarrolladas y mantenidas por los propios equipos de desarrollo, todas cuentan con una arquitectura de funcionamiento y definición de permisos independientes entre sí. Por lo tanto, sería muy costoso obtener una solución en el mercado que se adapte a las necesidades.

2.1.2.1 Necesidad de Ejecución de Proyecto

Como se mencionó anteriormente, la compañía se encuentra en pleno proceso de Certificación SOX, para lo cual se realizan Auditorias de procesos que permiten identificar y medir el grado de cumplimiento que ésta posee sobre la normativa mencionada anteriormente (de esta manera se definen las compañías como SOX

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

“compliance” o “non-compliance”). En el caso de identificar inconvenientes que generen riesgos de materialización de fraude financiero, se catalogan “issues” para los cuales deben generarse un plan de acción para que el mismo pueda ser resuelto o su riesgo resulte mitigado con controles.

Con el resultado de años de ejecución de estas auditorías, se concluyó que particularmente la “Despegar” cuenta con una gran cantidad de deficiencias asociadas a controles de accesos a aplicaciones críticas e incompatibilidades de segregación de funciones. Esto está relacionado a la gran cantidad de aplicativos “inhouse” (desarrollado y administrado por equipos de trabajo interno) y “outhouse” (es administrado y desarrollado por una entidad externa a la compañía la cual esta debe de contar con licencias para su uso legal) con los que cuenta para llevar a cabo su operatoria, cuya administración se encuentra administrada en su mayoría por los propios equipos de desarrollo ya que no se cuenta con un área que centralice y monitoree las mismas.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

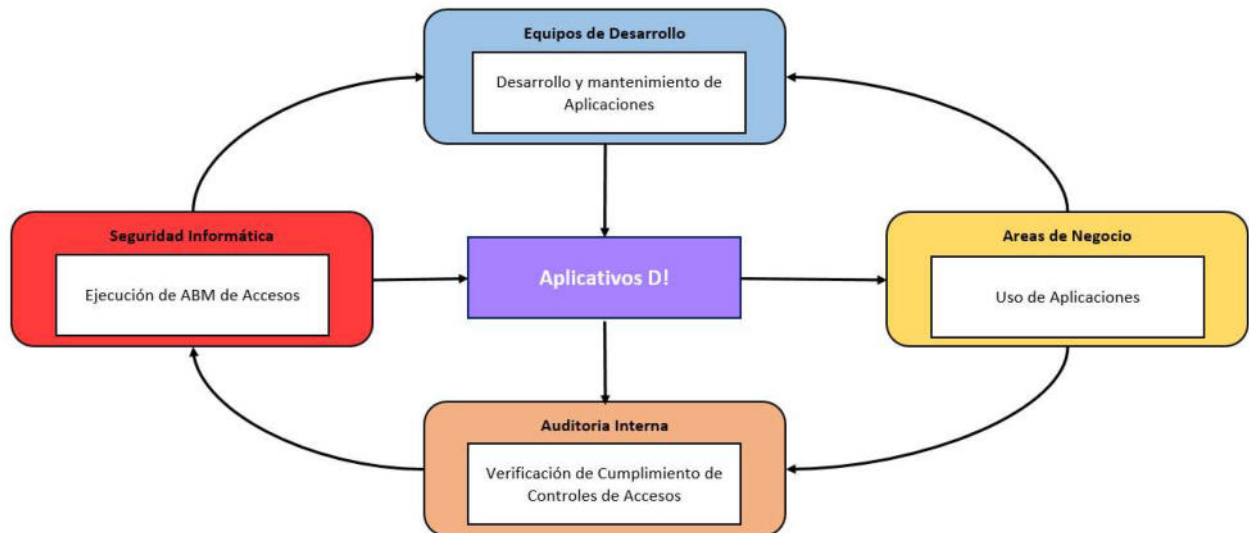


Ilustración 1- Interacción entre áreas para el uso de aplicativos en la compañía.
 Fuente: Elaboración propia basada en la práctica

3 DESARROLLO

3.1 Relevamiento del proyecto

En primer lugar, se realizó una indagación con los analistas del área de Auditoría Interna con el objetivo de identificar los pasos que generalmente pueden ser catalogados como repetitivos y con poco valor agregado durante el proceso de análisis de accesos aplicativos y la verificación de segregación de funciones (esto también permitió estimar el grado de impacto que puede generar la implementación de un automatismo).

Los pasos informados fueron los siguiente: analista debe comunicarse con el equipo que administra cada aplicativo en cuestión, solicitar el listado de permisos del mismo

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

y el reporte de usuarios que poseen asignados dichos permisos. Luego, debe obtener -desde el área de Recursos Humanos- la nómina de la entidad actualizada y verificar si el permiso otorgado a un usuario se encuentra efectivamente asignado para que este pueda ejecutar sus tareas diarias, según su rol dentro de la compañía, y de este modo chequear que no cuenta con un permiso que excede su categoría Jerárquica.

Seguidamente, para hacer un análisis de Segregación de funciones, se deben comparar cuáles son los roles dentro de un aplicativo que generen incompatibilidades, es decir, definir aquellos permisos que al estar asignados en conjunto a un usuario permita que el mismo no tenga la obligación de ejecutar un flujo de aprobación para la ejecución de una determinada acción (como por ejemplo la asignación a un usuario de un permiso que le permita solicitar una orden de compra en el aplicativo, y a su vez se le concede el permiso para emitir una orden de pago). En el caso de que esto ocurra, es necesario eliminar alguno de los permisos afectados. Nuevamente, por las características de la entidad, las acciones ejemplificadas anteriormente muchas veces pueden encontrarse segmentadas en diversos aplicativos, por lo tanto, para casos particulares esta revisión no estará completa si no se hace el análisis de los demás permisos configurados en todos los aplicativos de la compañía.

Una vez definidos los procesos, se llevó a cabo una reunión con la Gerencia y Coordinadores del área con el fin de estipular prioridades, definir etapas de trabajo y recursos requeridos para ejecutar los desarrollos.

Debido a la gran cantidad de aplicativos con los que cuenta la entidad, se consensuó definir el proyecto en dos etapas, con el mismo foco en común, pero segmentando

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

el alcance de aplicativos. De este modo, se generó una diferenciación entre aplicaciones cuya administración es llevada a cabo por la entidad (es decir, aquellas que permiten realizar acciones que intervengan directamente en la financiación de la compañía y generen un impacto reflejado en los balances contables de la compañía) y aquellas cuya administración es llevada a cabo por Compañías Externas (como por ejemplo, el cobro de las aerolíneas que ofrecen servicio de vuelo)

Con respecto a las aplicaciones propias y administradas por la Compañía se repartieron en tres grandes entornos, que se diferencian por sus características para la ejecución de ciertas tareas financieras definidas en la Entidad. En un primer grupo se especificaron los aplicativos relacionados a modificación de reglas para el cobro proveedores y cupos (llamado "ATP1"); el siguiente grupo se involucró los accesos al ERP de la compañía (Oracle EBS) y por último, un entorno de aplicativos donde se definen diversas tareas como por ejemplo: reservas de servicios, aplicativos de comunicación de interfaces tales como shopping, checkout, postventa, etc (Este subconjunto fue denominado como "ATP3")

3.2.1 Funcionalidad de la automatización:

En una primera instancia se hizo foco en el control de Accesos a los ambientes ATP1, ATP3, Oracle y aplicativos externos, donde el objetivo consistió en realizar una verificación sobre los accesos brindados en dichos ambientes y así validar los mismos fueron correctamente asignados.

De este modo, la intención fue simular de manera automática los cruces de datos que realizan los auditores entre los reportes de accesos asignados, el puesto de los

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

usuarios según la nómina de la entidad y una política definida por la propia área de Auditoría que especifica qué roles deben corresponder en los aplicativos según área/puesto del usuario.

La segunda parte del desarrollo se basó en la verificación de SOD en los ambientes mencionados anteriormente, donde se hizo foco en la verificación de aquellas incompatibilidades de asignación de permisos que posean funcionalidades conflictivas. Al igual que el caso anterior, se realizó una verificación de todos los accesos a los aplicativos en cuestión y se comprobó que un mismo usuario cuenta con accesos a realizar actividades conflictivas según una política administrada por el área de Auditoría, la cual se basa en la definición de los permisos que generan inconvenientes de segregación de funciones entre sí.

De esta manera, se obtuvieron los resultados de los análisis en forma de reportes (con periodicidad de generación semanal) a los cuales se pueden acceder por medio de ficheros de Google Drive. Los mismos, se encuentran diferenciados del siguiente modo: un informe por ambiente que lista los accesos incorrectos, un informe por ambiente que define los accesos correctos (para contar con la totalidad del universo analizado) y un último informe que da detalle de los usuarios que poseen accesos con inconvenientes definidos por SOD (Segregations of duties).

3.2.2 Definiciones Técnicas.

Las tareas que se realizaron la automatización corresponde en primer lugar a la obtención de los listados de usuarios con accesos y permisos definidos a los ambientes mencionados en párrafo anterior, ya sea por conexión a bases de datos, consumo de API's u obtención de información por medios de protocolos específicos.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

Luego, se continuó con la obtención de la nómina de empleados de la entidad para contar con datos laborales de los empleados, siendo estos su nombre y apellido, puesto, área, usuario AD, usuario de email, nombre de su superior y estado actual en la compañía (Activo o Inactivo). Esta nómina fue proporcionada por el equipo de Recursos Humanos por medio de un protocolo SFTP.

Por último, se establecieron las conexiones a las políticas mencionadas anteriormente por medio de la API proporcionada por Google, donde se generaron las credenciales requeridas para la comunicación entre el automatismo y dichos servicios.

Los resultados de los análisis permanecen disponibles para que cualquier miembro del equipo de Auditoría Interna pueda acceder por medio de un fichero compartido en Drive, haciendo uso de las credenciales de Google mencionadas anteriormente.

Tanto el comienzo de la ejecución de la automatización, la generación de errores en el proceso y la finalización del mismo, se notifican vía mail para su posterior revisión y control.

Ventajas:

A nivel funcional, la realización de este sistema permitió reducir de gran manera los tiempos de ejecución ya que no se requirió invertir tiempo en la solicitud de listados de accesos y en las tareas para la definición de los reportes (como por ejemplo el efectuar cruces manuales), y evitando también la posibilidad de errores manuales en la ejecución de los análisis. Además de su fácil soporte y métodos de uso y la corta periodicidad de generación que permite se pueda contar con información actualizada constantemente, permitió que los reportes sean utilizados como fuentes

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

para generación de KPI's que miden la mejora de los procesos en la asignación de permisos a aplicativos.

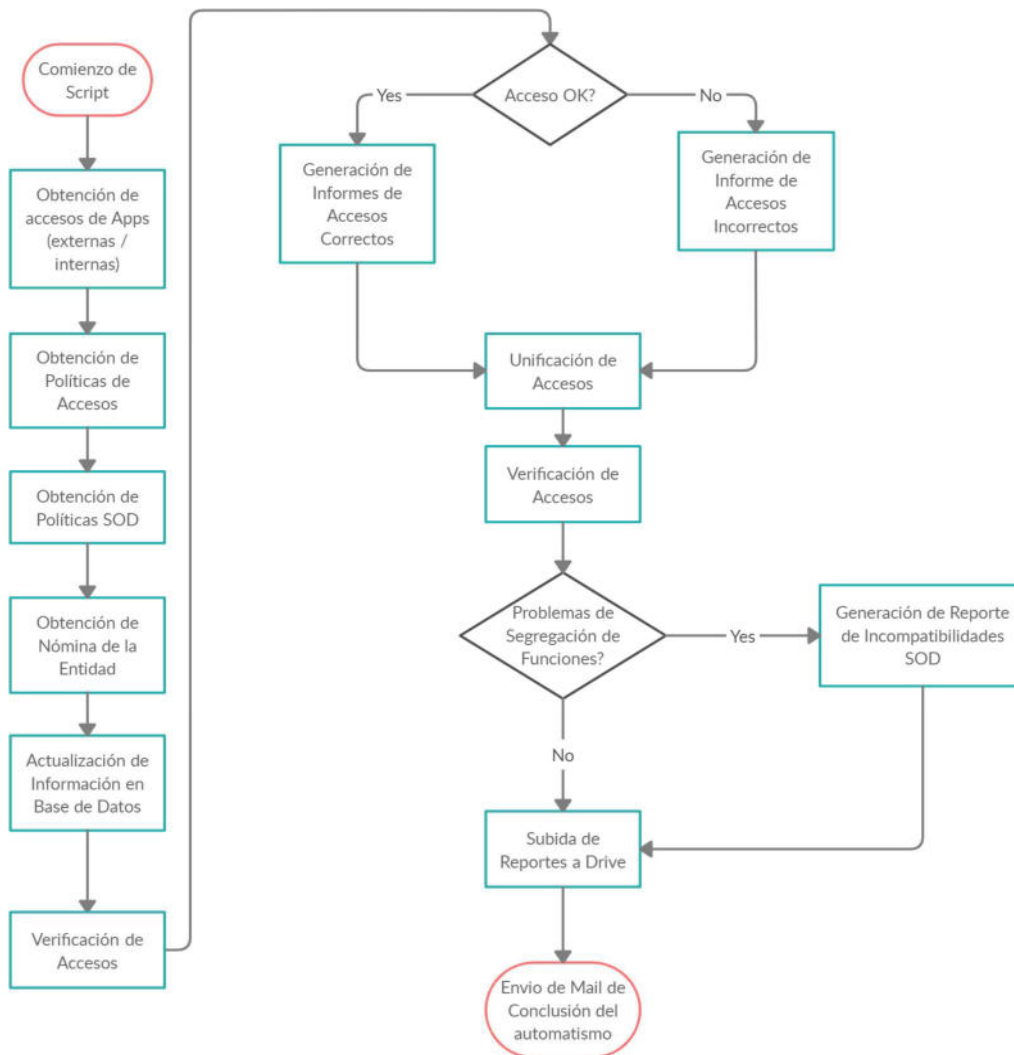


Ilustración 2- WorkFlow de la automatización

Fuente: Elaboración propia basada en la práctica

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

3.3 Herramientas

3.3.1 Python

Es un lenguaje de programación multiplataforma y multiparadigma, lo que quiere decir que permite hacer ejecutable su código fuente entre varios sistemas operativos. Es utilizado idealmente para la automatización de procesos y ejecución de tareas en tanto en entorno cliente como servidor.

Es un lenguaje interpretado, no compilado, fuertemente tipado y soporta varios paradigmas de programación como orientación a objetos, programación estructurada, imperativa y, en menor medida, funcional. A su vez, da gran facilidad a la hora de trabajar con grandes volúmenes de datos ya que, al ser multiplataforma, favorece su extracción y procesamiento.

El intérprete de Python estándar incluye un modo interactivo en el cual se escriben las instrucciones en una especie de intérprete de comandos: las expresiones pueden ser introducidas una a una, pudiendo verse el resultado de su evaluación inmediatamente, lo que da la posibilidad de probar porciones de código en el modo interactivo antes de integrarlo como parte de un programa.



Ilustración 3 - Lenguaje de Programación Python
Fuente: <https://es.wikipedia.org/wiki/Archivo:Python-logo-notext.svg>

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

3.3.2 MySQL

MySQL es un sistema gestor de bases de datos. Se trata de un sistema de libre distribución y de código abierto. Su base del funcionamiento es el sistema Linux, por eso MySQL se distribuye fundamentalmente para en este SO, aunque también hay versiones para Windows.

Existen cuatro versiones de MySQL:

- Estándar, la cual incluye el motor estándar y la posibilidad de usar bases de datos
- InnoDB, que ofrece todo el potencial de MySQL, pero sin soporte completo para utilizar transacciones.
- Max, es utilizado para usuarios que requieran MySQL con herramientas de prueba para realizar opciones avanzadas de base de datos
- Pro, siendo la versión comercial del MySQL estándar
- Classic, es igual que la estándar pero no dispone de soporte para InnoDB

El uso de MySQL (excepto en la versión Pro) está sujeto a licencia GNU public license (llamada GPL). Esta licencia admite el uso de MySQL para crear cualquier tipo de aplicación. Se pueden distribuir copias de los archivos de MySQL, salvo que esas copias se cobren a un tercer usuario. Se prohíbe cobrar por incluir MySQL. Se puede modificar el código fuente de MySQL, pero si se distribuye la aplicación con el código modificado, habrá que obtener una copia comercial y consultar sobre el cobro de la licencia. Al distribuir copias, se tiene que poder obtener información sobre las licencias GNU.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

La conexión al servidor MySQL para crear, modificar o realizar cualquier otra operación sobre bases de datos, se realiza mediante el programa mysql que se encuentra en la carpeta bin del programa. La ejecución de este programa redirige al llamado monitor de MySQL que es la línea de comandos desde la que es posible ejecutar instrucciones MySQL.



Ilustración 4 - MySQL

Fuente: <https://raiolanetworks.es/blog/optimizar-mysql-configurando-archivo-my-cnf/>

3.3.3 Oracle

Oracle es un sistema de gestión de base de datos relacional fabricado por Oracle Corporation. Es básicamente una herramienta cliente/servidor para la gestión de base de datos la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general. Permite que las organizaciones puedan controlar y gestionar grandes volúmenes de contenidos no estructurados en un único repositorio con el objetivo de reducir los costes y los riesgos asociados a la pérdida de información.

Una BD Oracle tiene una estructura física y una estructura lógica:

- La estructura física se corresponde a los ficheros del sistema operativo.
- La estructura lógica está formada por los tablespace y los objetos de un esquema de BD

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

A nivel lógico se divide en unidades de almacenamiento:

- Tablespaces: Cada BD estará formada por uno o mas tablespaces (al menos existe el tablespace SYSTEM catálogo del sistema) Cada tablespace se corresponde con uno o más ficheros de datos.
- Objetos: correspondiente a tablas, vistas, índices asociados a una tabla, clusters, etc.



Ilustración 5 - Oracle

Fuente: <http://www.linware.com.ar/blog/2013/07/02/oracle-y-su-primera-base-de-datos-en-la-nube-database-12c/>

3.3.4 Powershell

PowerShell es un marco de gestión de configuración y automatización de tareas multiplataforma, que consta de un shell de línea de comandos y un lenguaje de secuencias de comandos. A diferencia de la mayoría de los shells, que aceptan y devuelven texto, PowerShell se basa en .NET Common Language Runtime (CLR) y acepta y devuelve objetos .NET. Este cambio fundamental trae herramientas y métodos completamente nuevos para la automatización.

Los comandos de PowerShell se conocen como cmdlets. Es posible usar cada cmdlet por separado, pero su poder se obtiene cuando los combina para realizar tareas complejas. Como muchos shells, PowerShell da acceso al sistema de archivos en la computadora. Los proveedores de PowerShell permiten acceder a

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

otros almacenes de datos, como el registro y los almacenes de certificados, de manera tan sencilla como para se accede al sistema de archivos.

A su vez, es posible crear un cmdlet propio y módulos de funciones utilizando código compilado o scripts. Los módulos pueden agregar cmdlets y proveedores al shell. PowerShell también admite scripts que son análogos a los scripts de shell de UNIX y los archivos por lotes cmd.exe.

A diferencia de las interfaces de línea de comandos tradicionales, los cmdlets de PowerShell están diseñados para tratar con objetos. Un objeto es información estructurada que es más que la cadena de caracteres que aparecen en la pantalla. La salida del comando siempre lleva información adicional que puede usar si la necesita.

En la mayoría de los casos, no es necesario utilizar herramientas de procesamiento de texto para extraer información específica. Por lo tanto, se puede acceder directamente a partes de los datos mediante la sintaxis de objetos de PowerShell estándar.



Ilustración 6 - Poweshell

Fuente: <https://itblogsogeti.com/2014/10/15/powershell-interactuando-con-el-sistema-operativo-lluis-cruz-sogeti/>

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

3.3.5 SQLite

Es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kiB)² biblioteca escrita en C. SQLite es un proyecto de dominio público.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un solo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.



Ilustración 7 – SQLite

Fuente: <https://www.sqlite.org/index.html>

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

3.4 Librerías

3.4.1 OS

Librería manipulada para la interacción con el Sistema Operativo. Este módulo es utilizado en la automatización para operar las credenciales de las diversas plataformas, como por ejemplo en la definición de variables de entorno.

3.4.2 CVS

Utilizado para el maquetado de datos en formato CSV. Dicho formato es utilizado en la automatización por la sencillez que ofrece al manipular grandes volúmenes de información y facilidad en la estructura de cadenas de caracteres.

3.4.3 Google

Librería utilizada para la conexión con la API de Google, dicha librería permite la comunicación contra Gsheet y Drive que alimentan el automatismo y a su vez la exposición de los datos. Esta comunicación solamente puede ser llevado por medio de la configuración de credenciales generadas en la consola de desarrollo de Google.

3.4.4 PANDAS

Es una librería que permite establecer estructuras de datos rápidas, flexibles y expresivas. Fundamentalmente está desarrollada para trabajar con estructuras relacionales. Si bien puede ser utilizada en gran cantidad de lenguajes, es de mucha utilidad para ser implementada en Python. Entre sus ventajas se destaca:

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

- Fácil manejo de los datos faltantes (representados como NaN) tanto en coma flotante como en datos de coma no flotante.
- Mutabilidad de tamaño: las columnas se pueden insertar y eliminar de DataFrame y objetos de mayor dimensión.
- Alineación de datos automática y explícita: los objetos se pueden alinear explícitamente a un conjunto de etiquetas, o el usuario puede simplemente ignorar las etiquetas y dejar que Series, DataFrame, etc. alineen automáticamente los datos para usted en los cálculos.
- Grupo por funcionalidad potente y flexible para realizar operaciones de división, aplicación y combinación en conjuntos de datos, tanto para agregar como para transformar datos
- Facilita las conversiones de estructuras.

3.4.6 Sqlalchemy

Librería utilizada para la conexión con Bases de Datos SQL y ejecución de consultas. Es utilizado ya que se considera como un gran motor para la interacción con este tipo de Bases de Datos y permite la ejecución de tareas de forma muy performante y ágil. Además, cabe destacar su característica de combinar en una misma librería herramientas de Python SQL y Object Relational Mapper, generando así una gran facilidad de manipulación de datos al momento del desarrollo de tareas.

3.4.7 Cx_Oracle

Utilizado para la conexión contra Bases de Datos Oracle. Internamente, cx_Oracle carga dinámicamente las bibliotecas de cliente de Oracle para acceder a la base. Particularmente, la misma puede estar en la misma máquina que Python o puede alojarse remotamente.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

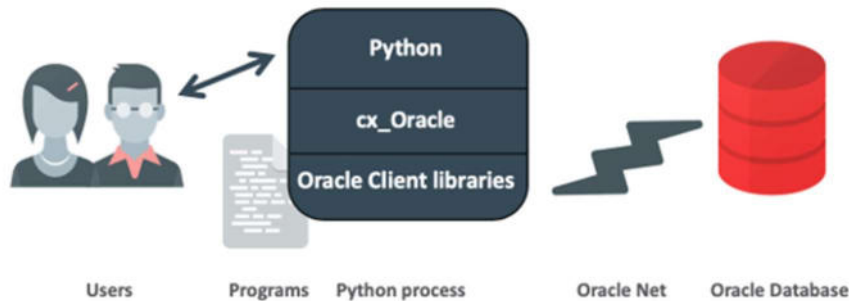


Ilustración 8 – Interacción Python y Base de Datos Oracle por Librería cx_Oracle
 Fuente: https://cx-oracle.readthedocs.io/en/latest/user_guide/introduction.html

3.5 Metodología de Trabajo: Cascada

Luego de verificar alternativas, se definió que la metodología de trabajo que mejor se adaptaba a la implementación del proyecto era la del desarrollo en cascada. Esta se caracteriza por ser del tipo secuencial en donde se especifican etapas de manera consecutiva y obligatorias, y no es posible saltar ninguna de las mismas (es decir, no puede comenzar una sin haber culminado la anterior).

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

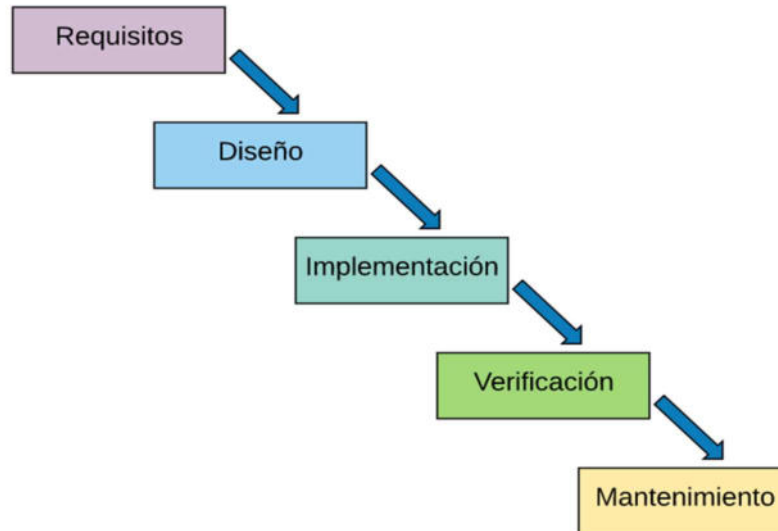


Ilustración 9 – Esquema metodología en cascada

Fuente: <https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada>

Para este proyecto en particular, los líderes del área definen las funcionalidades que debe tener la automatización. Luego, entre el jefe directo y el desarrollador generan el requerimiento, criterio y características que debe cumplir el software para ser funcional y tener éxito en su alcance. Una vez formalizado los requerimientos en un documento funcional, se especifica técnicamente los pasos a seguir y el desarrollador implementa el flujo definido en el gráfico para continuar con las tareas. Por último, los usuarios finales definen si la funcionalidad cumple con las expectativas.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

Esta metodología fue pensada por sobre otra ya que se necesitó contar con los resultados finales del desarrollo para corroborar si la funcionalidad cumplía con lo esperado. Esto es así ya que por características propias el automatismo se puede iterar e ir mostrando resultados parciales debido a que la lógica y la suma de módulos (aplicativos para analizar) para la amplitud del alcance, obliga a que se deba optar por corroborar resultados una vez ya finalizado e implementado el desarrollo.

Como toda metodología de trabajo presenta ventajas y desventajas.

Ventajas:

- Se sientan las bases del desarrollo en primer paso dejando todo documentado y registrado, se puede conocer la completitud del alcance del proyecto una vez que este comienza.
- Acorta tiempos de definición de objetivos y requerimientos, ya que estos se generan una sola vez. Además, permite que estos sean más claros desde un principio.
- Favorece la medición del progreso del proyecto, ya que el alcance completo se conoce de manera anticipada.
- Es ideal para proyectos que requieren una rápida solución.

Desventajas:

- No hay interacción de la parte usuario mientras se está desarrollando, así no se puede plantear un cambio de alcance o requisitos hasta que se termine todo el proyecto.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

- No pueden paralelizarse etapas, por lo cual si una queda estancada por alguna complicación técnica, retrasará el comienzo de la siguiente y los tiempos de entrega del proyecto
- No es recomendable para proyectos muy largos y que requieran de entregas intermedias.
- No hay oportunidad de detectar fallos mientras se está desarrollando y corregirlos en el momento. Entonces, luego de que se termine el desarrollo, habrá que corregirlos todos juntos. (Etapa de verificación)

3.6 Arquitectura de la automatización

Esta automatización se centra en un Cluster (“audit-aut”) alojado en Cloudia (plataforma de gestión que aloja todos los servicios a nivel software utilizados por la entidad en la nube) que actúa como orquestador ya que cuenta con los scripts programados para la conexión y el intercambio de credenciales con Google Drive (por medio de la API de Google), y por último la conexión con una Virtual Machine (“Reg-Auditoría”) que almacena la Base de Datos Local creada con SQLite. A su vez, se comunica a las bases de datos de Oracle, ATP3 y ATP1 para la obtención de los accesos, y luego al script (alojado en la VM mencionada anteriormente) que se conecta a un servidor administrado por el equipo de RRHH que almacena dentro de sus Files las nóminas de la compañía. Por último, tiene seteadas las configuraciones para las reglas de Firewall que permiten la comunicación contra Aplicaciones Externas.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

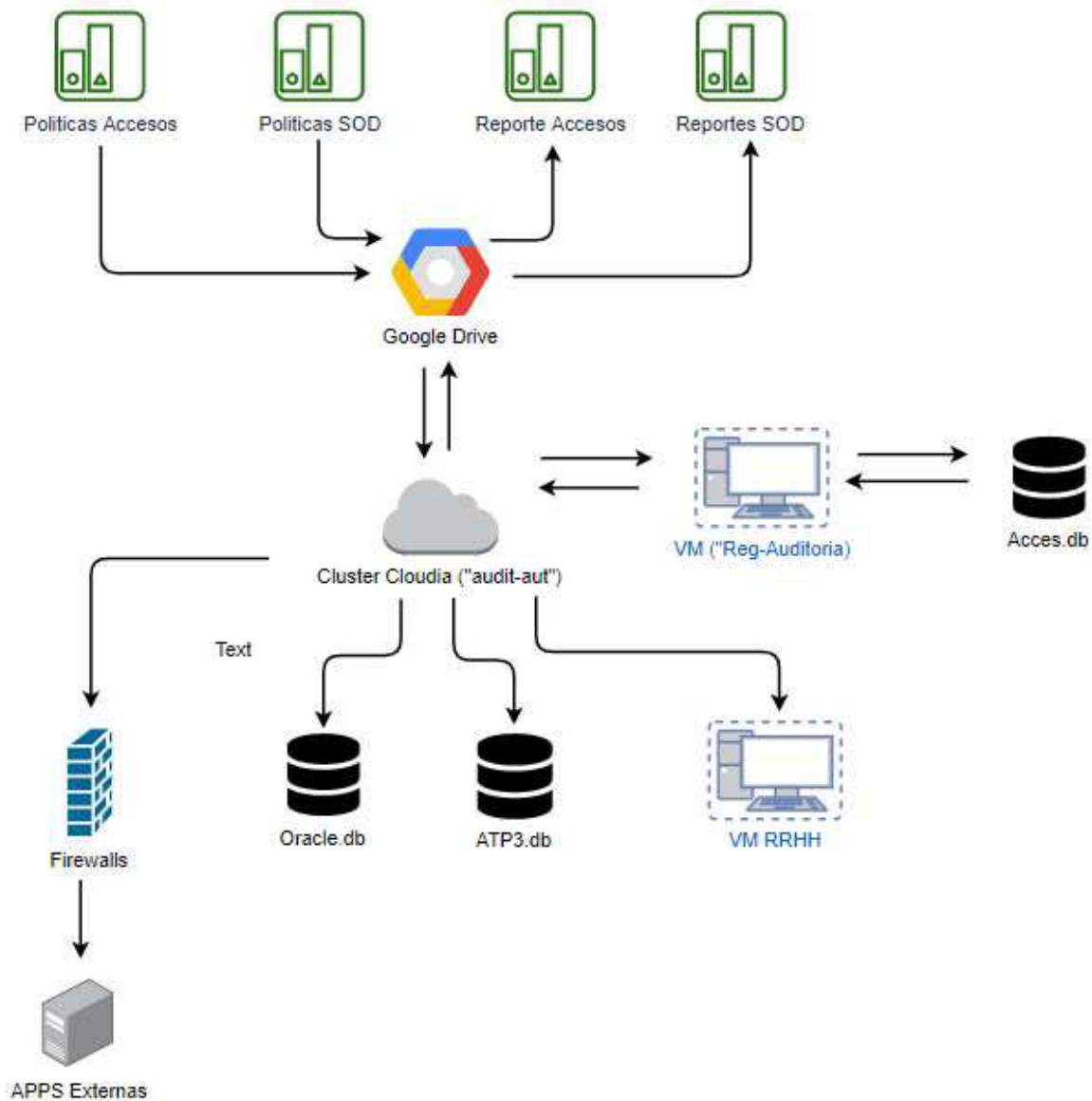


Ilustración 10 - Arquitectura la Automatización
 Fuente: Elaboración propia basada en la práctica

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

3.7.1 Obtención de reportes de accesos a aplicativos

Se comenzó por la conexión a las bases de Datos ATP1, ATP3 y Oracle. Para ello se generaron tres scripts independientes que se ejecutan de manera consecutiva. Para ATP1 y ATP3 se emplearon las mismas funciones ya que ambas son bases de datos MySQL.

En primer lugar, se importó la librería 'os' ya que por medio de ella se utilizaron las funciones de variables de entornos para obtener el usuario y contraseña para conexión a la base de datos. Por medio de la librería 'sqlalchemy' se estableció dicha conexión y luego se generó una query para obtener un listado de usuarios, aplicaciones que tienen acceso y roles definidos en cada uno en el ambiente. Dicha consulta fue ejecutada gracias a las funciones de lectura de queries que provee la librería 'pandas'. Finalmente se generó una función que disponibiliza un reporte en formato 'csv' para ser exportado.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------


```

queryATP3.py
import pandas as pd
import sqlalchemy
import os

from ManejadorError import mailError
def queryATP3():
    usuario = os.environ.get('UsuarioATP')
    contrasenia = os.environ.get('PassATP')

    engine = sqlalchemy.create_engine('mysql+pymysql://'+usuario+'@'+contrasenia+'@:3306/atp')

    query = '''
SELECT distinct atp.application.id ID, atp.application.name APPLICATION, atp.application.description DESCRIPTION, atp.role.name ROL,
atp.role.description ROLDESCRIPTION, atp.user.username USERNAME, atp.user.email MAIL
FROM atp.user
JOIN atp.user_assigned_role ON atp.user.id = atp.user_assigned_role.user_id
JOIN atp.assigned_role ON atp.user_assigned_role.assigned_role_id = atp.assigned_role.id
JOIN atp.role ON atp.assigned_role.role_id = atp.role.id
JOIN atp.application ON atp.role.application_id = atp.application.id
RIGHT JOIN atp.user_application ON atp.user.id = atp.user_application.user_id
WHERE (atp.user.deleted = 0) and (atp.user.active = 1) and (atp.user.source = 'AD')

...

df = pd.read_sql_query(query,engine)

export_csv = df.to_csv ("E:/3- Reportes SOD/accesATP3.csv", index = None, header=True)
    
```

Ilustración 12 - Conexión a base de datos ATP3 y obtención de accesos.

Fuente: Elaboración propia basada en la práctica

Para la conexión a la base de datos de Oracle, se optó por la misma lógica que la descrita en el párrafo anterior, salvo que fue necesario utilizar el módulo 'xc_Oracle' para la implementación de la autenticación a la base y se generó una función con el módulo 'csv' para crear el reporte de accesos y responsabilidades asignadas por usuario.

Por último, la obtención de los accesos a aplicativos externos quedó sujeta a la forma en la que cada proveedor envía la información. Esta puede ser tanto de manera manual (como por ejemplo, enviado un listado de accesos por mail) o

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

automática (dejando disponible un reporte en un directorio específico alojado en un host para acceder al mismo mediante un SFTP).

3.7.2 Obtención de nómina de empleado

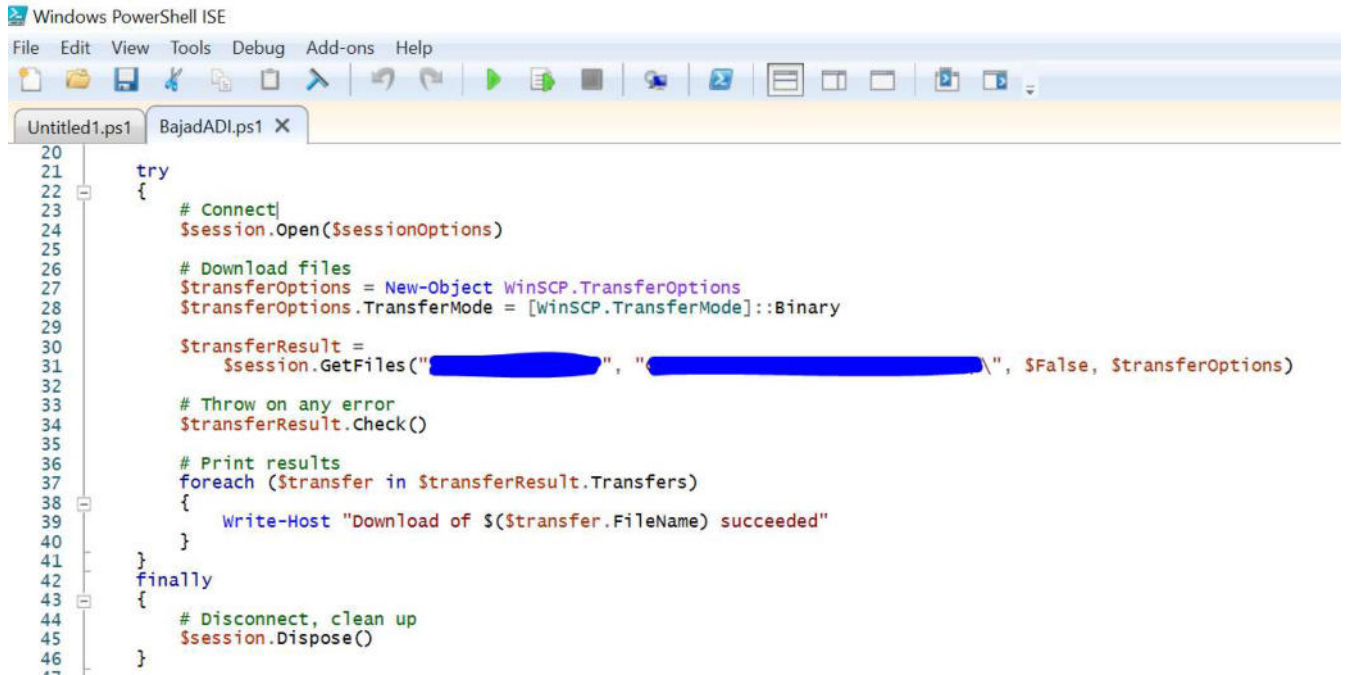
Seguidamente, se continuó con la obtención de nómina de empleados de la entidad para verificar el puesto y área de los usuarios con accesos a los aplicativos. Para ello, el equipo de RRHH creó un reporte que se actualiza diariamente de manera automática por medio del aplicativo Peoplesoft², el cual queda disponible en formato .csv en un cluster administrado por el propio equipo.

De este modo, se estableció un script por medio de Powershell para obtener el reporte en cuestión por medio de un usuario proporcionado por el equipo de Seguridad Informática, que posee las credenciales necesarias iniciar una sesión al cluster de RRHH por medio del protocolo SFTP.

En dicho script, se constituyó una función que busca el reporte en una ruta específica y luego se deja disponible localmente para alimentar la automatización.

² Peoplesoft: CRM de Oracle utilizado para la gestión y administración de Recursos Humanos de una entidad.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------



```

20
21
22     try
23     {
24         # Connect
25         $session.Open($sessionOptions)
26
27         # Download files
28         $transferOptions = New-Object WinSCP.TransferOptions
29         $transferOptions.TransferMode = [WinSCP.TransferMode]::Binary
30
31         $transferResult =
32             $session.GetFiles("██████████", "██████████", $false, $transferOptions)
33
34         # Throw on any error
35         $transferResult.Check()
36
37         # Print results
38         foreach ($transfer in $transferResult.Transfers)
39         {
40             Write-Host "Download of $($transfer.FileName) succeeded"
41         }
42     } finally
43     {
44         # Disconnect, clean up
45         $session.Dispose()
46     }

```

Ilustración 13 – Obtención de Reporte de Nómina de RRHH.
Fuente: Elaboración propia basada en la práctica

Un error que se descubrió es que en ocasiones el reporte se generaba con un número de líneas muy reducido, por lo que se implementó una función dentro del propio script que corrobora la cantidad de líneas que posee dicho reporte, para posteriormente realizar una validación la cual especifica que en el caso de contar con menos de 4500 líneas (número aproximado de empleados que se encuentra actualmente activos en la compañía) se ejecute nuevamente el proceso de obtención del archivo.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

```
47 | ~
48 | }
49 |
50 | function validacion {
51 | [int]$LinesInFile = 0
52 | $reader = New-Object IO.StreamReader [REDACTED]
53 | while($reader.ReadLine() -ne $null){ $LinesInFile++ }
54 | return $LinesInFile
55 |
56 | }
57 |
58 |
59 | while ($semaforo){
60 |     ejecucion
61 |     $val = validacion
62 |     if($val -ge 4500){
63 |         $semaforo = $false
64 |         Write-Output OK
65 |     }else {
66 |         $semaforo = $true
67 |         Write-Output BAD
68 |     }
69 | }
70 |
71 |     exit 0
72 |
73 | }
74 | catch
75 | {
76 |     Write-Host "Error: $($_.Exception.Message)"
77 |     exit 1
78 | }
```

Ilustración 14 – Validación de cantidad de líneas.
Fuente: Elaboración propia basada en la práctica

3.7.3 Obtención de Políticas de Accesos y SOD

Las “Políticas de Accesos a Aplicativos” que definen el deber ser entre los permisos a aplicativos según el puesto en nómina de los usuarios y la “Política SOD” que establece las incompatibilidades entre permisos, se encuentran definidas en un GSheet administrado por el equipo de Auditoría Interna.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

Plan de Automatización de Controles ☆ ☰ ☰

Archivo Editar Ver Insertar Formato Datos Herramientas Complementos Ayuda Última modificación hace 2 minutos

100% € % .00 .00 123 Arial 10 B I S A

Control_SOX

	A	B	C	F
1	INFORMACION PARA HACER EL CRUCE			
2	Responsabilidad	AREA_ADI	PUESTO_ADI	Control_SOX
3	DSP AR FA ANALISTA	Administración Ar	Analista Ssr de Contabilidad	ARActivos Fijos01
4	DSP AR FA CONFIGURADOR	Soporte ERP	Analista Ssr Funcional ERP	ARActivos Fijos01
5	DSP AR FA CONFIGURADOR	Soporte ERP	Analista Ssr Funcional ERP	ARActivos Fijos01
6	DSP AR FA ANALISTA	Administración Ar	Analista Sr de Finanzas	ARActivos Fijos01
7	DSP AR FA ANALISTA	Administración Ar	Gerente de Administración Ar	ARActivos Fijos01
8	DSP AR FA CONFIGURADOR	Soporte ERP	Líder Funcional ERP	ARActivos Fijos01
9	DSP AR FA CONFIGURADOR	Soporte ERP	Analista Sr Funcional ERP	ARActivos Fijos01
10	DSP AR FA CONFIGURADOR	Soporte ERP	Coordinador Funcional ERP	ARActivos Fijos01
11	DSP AR FA CONFIGURADOR	ERP	Analista Ssr Funcional ERP	ARActivos Fijos01
12	DSP AR FA CONFIGURADOR	Soporte ERP	Analista Sr Funcional ERP	ARActivos Fijos01
13	DSP AR FA CONFIGURADOR	ERP	Analista Jr	ARActivos Fijos01
14	DSP AR FA CONFIGURADOR	ERP	Analista Sr Funcional ERP	ARActivos Fijos01
15	DSP AR FA CONFIGURADOR	Soporte ERP	Analista Ssr Funcional ERP	ARActivos Fijos01
16	DSP AR FA CONFIGURADOR	ERP Br	Coordinador Funcional ERP	ARActivos Fijos01

Ilustración 15 – Definición de “Política de Accesos”.
Fuente: Elaboración propia basada en la práctica

Plan de Automatización de Controles ☆ ☰ ☰

Archivo Editar Ver Insertar Formato Datos Herramientas Complementos Ayuda Última modificación hace 5 minutos

120% € % .00 .00 123 Calibri 11 B I S A

Administrador

	A	B	C	D	E	F	G
1	ID_Argumento	G1	APP1	ROL1	G2	APP2	ROL2
2	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 ds-bo	Comercial_Revision
3	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 dee-dee	Administrador
4	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 dee-dee	ABM
5	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 dee-dee	ROLE_BR_ADMIN_SUPERVISOR
6	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 dee-dee	ROLE_EQUIPO_DE_DATA_ENTRIES
7	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 dee-dee	ROLE_IT_CONTENIDO
8	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 dee-dee	ROLE_CONTENT_LEADERS
9	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 dee-dee	ROLE_NEGOCIO_HOTELES
10	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 dee-dee	Sourcing - Area Manager
11	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 dee-dee	ROLE_UY_GEEE_PAD_SUPERVISOR
12	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 dee-dee	ROLE_UY_PAD_TEAM_LEADER
13	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 cars-contracts	commercialManager
14	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 sara	AccountExecutive
15	1	1	dee-dee	ROLE_HOTEL_PRODUCT_TEAM		2 sara	AMPublicacion
16	1	1	dee-dee	ROLE_ABM_PROVEEDORES_AND_CLIENTES		2 ds-bo	Comercial_Revision

Ilustración 15 – Definición de “Política SOD”.
Fuente: Elaboración propia basada en la práctica

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

Es por eso que para obtenerlas, se confeccionó un script que permita llevar a cabo una conexión a dichas políticas, para de esta manera poder manipular su información de forma local.

Para ello fue necesario establecer un proyecto en la Consola Cloud de Google³ y definir un proyecto.

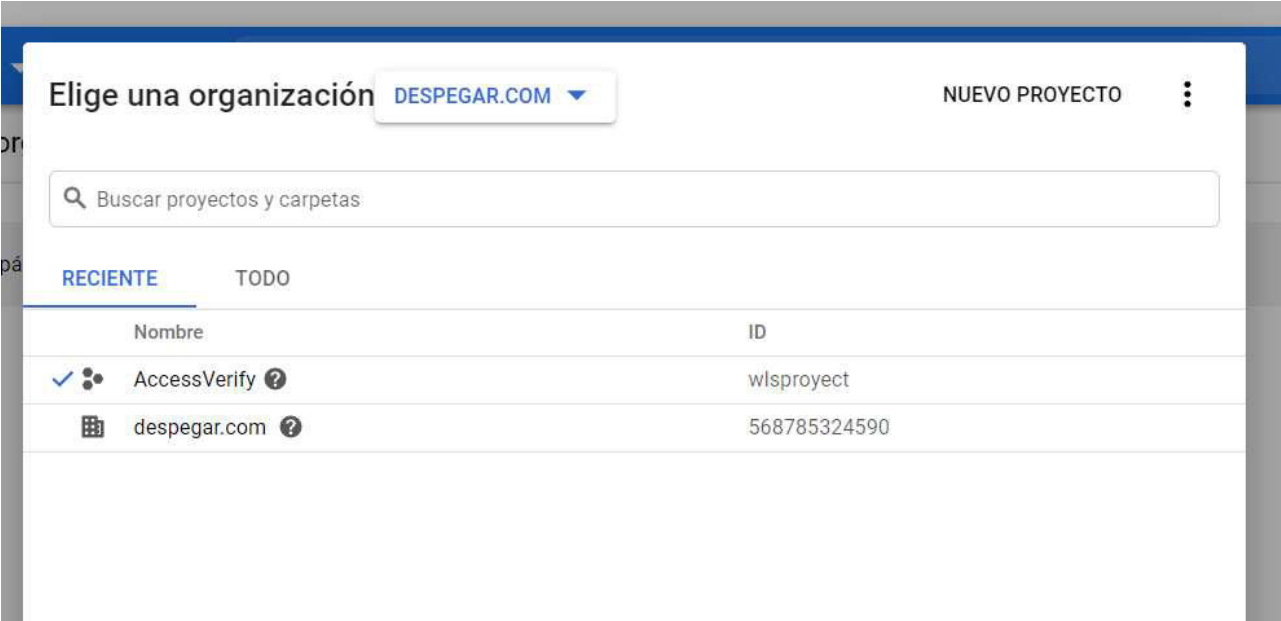


Ilustración 16 - Generación de Proyecto en Consola Cloud de Google.
Fuente: Elaboración propia basada en la práctica

Seguidamente, se seleccionaron las API's de Google a utilizar (las cuales correspondían a "Google Drive" y "Google Sheet") y se vincularon al proyecto en cuestión.

³ Consola Cloud de Google: entorno que ofrece Google para la generación de credenciales y utilización de sus API's para desarrollo de software de código libre.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

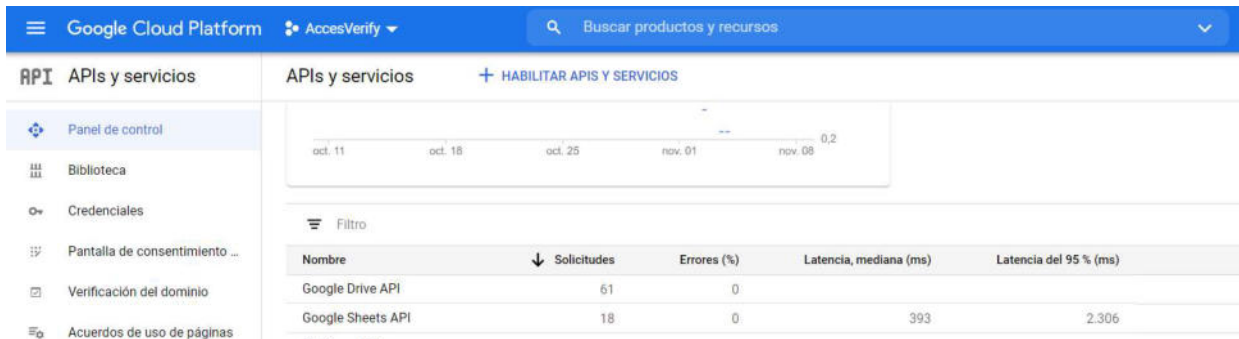
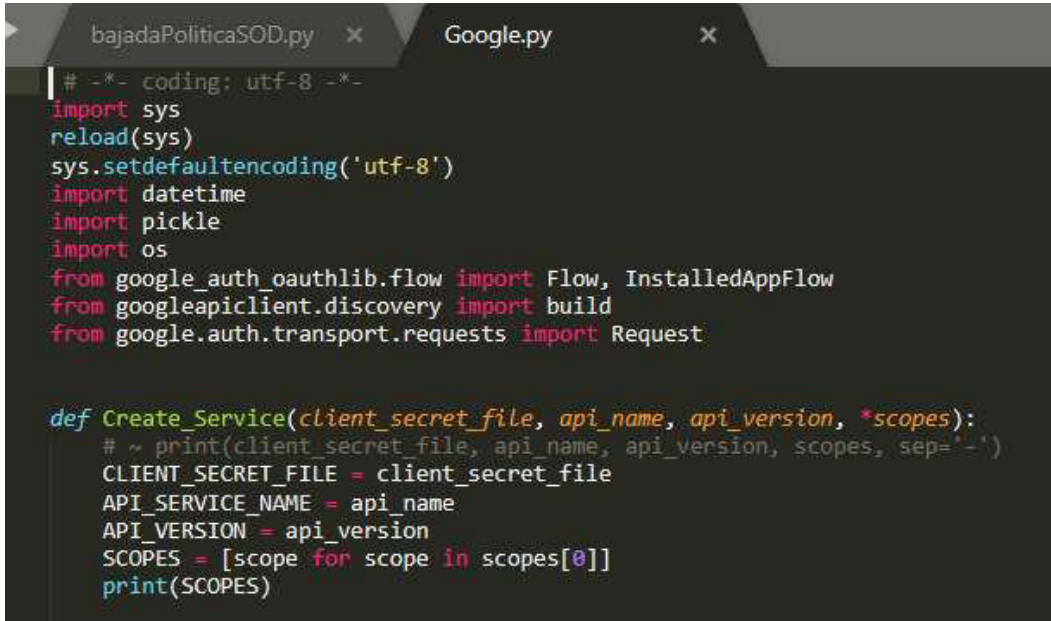


Ilustración 17- Selección de API's de Google.
Fuente: Elaboración propia basada en la práctica

Por último, desde dicha consola se generó un archivo .json llamado "client_secret" el cuál cuenta con las credenciales requeridas para la comunicación entre un script de Python y el proyecto de Google junto a sus API's vinculadas.

Una vez finalizada las configuraciones dentro de la consola, se llevó a cabo un script de Python denominado "Google" en el cuál se estableció una función que permite crear la conexión por medio dichas credenciales, especificando el tipo de API a utilizar.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------



```

bajadaPoliticaSOD.py x Google.py x
# -*- coding: utf-8 -*-
import sys
reload(sys)
sys.setdefaultencoding('utf-8')
import datetime
import pickle
import os
from google_auth_oauthlib.flow import Flow, InstalledAppFlow
from googleapiclient.discovery import build
from google.auth.transport.requests import Request

def Create_Service(client_secret_file, api_name, api_version, *scopes):
    # ~ print(client_secret_file, api_name, api_version, scopes, sep='-')
    CLIENT_SECRET_FILE = client_secret_file
    API_SERVICE_NAME = api_name
    API_VERSION = api_version
    SCOPES = [scope for scope in scopes[0]]
    print(SCOPES)

```

Ilustración 18 - Generación de script Google.
Fuente: Elaboración propia basada en la práctica

Seguidamente se definieron scripts independientes con el objetivo de obtener cada política en un formato .csv. En cada uno de ellos se especificó el uso de credenciales por medio del “client_secret.json” y se estableció el ID del GSheet en el que se cargan las políticas, junto con el nombre de la solapa a utilizar.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:


```
bajadaPoliticaSOD.py x Google.py x
# -*- coding: utf-8 -*-
import sys
reload(sys)
sys.setdefaultencoding('utf-8')

from Google import Create_Service
import pandas as pd
from ManejadorError import mailError

def bajadaPoliticaSOD():
    CLIENT_SECRET_FILE = 'client_secret.json'
    API_SERVICE_NAME = 'sheets'
    API_VERSION = 'v4'
    SCOPES = ['https://www.googleapis.com/auth/spreadsheets.readonly']
    gsheetId = '...'

    s = Create_Service(CLIENT_SECRET_FILE, API_SERVICE_NAME, API_VERSION, SCOPES)
    gs = s.spreadsheets()
    rows = gs.values().get(spreadsheetId=gsheetId, range='SOD').execute()
    data = rows.get('values')
    df = pd.DataFrame(data)

    export_csv = df.to_csv(r"..." 'SOD_POLITICA.csv', index = None, header=True)

try:
    bajadaPoliticaSOD()
except:
    mailError("Se genero un error en la bajada de la politica de SOD")
```

Ilustración 19 - Ejemplo de script para obtención de Política SOD.
Fuente: Elaboración propia basada en la práctica

3.7.4 Carga de Reportes a Base de Datos

Como se mencionó anteriormente, el tipo de base utilizada para almacenar los datos generados por la automatización corresponde a un SQLite. Ésta fue utilizada debido a su rápida y fácil implementación (sumado a que es una tecnología que no poseía conocimiento).

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

Para llevar a cabo dicha implementación, nuevamente se desarrolló un script importando la librería “pandas” para el manejo de los reportes .csv y la librería “sqlite3” para la conexión a la base “access.db” (base local utilizada como motor de la automatización)

```

File Edit Selection Find View Goto Tools Project Preferences Help
SubidaDatosBD.py
1 # -*- coding: utf-8 -*-
2 import sys
3 reload(sys)
4 sys.setdefaultencoding('utf-8')
5
6 import pandas as pd
7 import sqlite3
8
9 from ManejadorError import mailError
10
11 def SubidaDatosBD():
12     #Cargo csv
13
14     dfORACLE = pd.read_csv("XXXXXXXXXXXXXXXXXXXX/accesORACLE.csv")
15     dfRRHH = pd.read_csv("XXXXXXXXXXXXXXXXXXXX/RRHH.csv")
16     dfATP3 = pd.read_csv("XXXXXXXXXXXXXXXXXXXX/accesATP3.csv")

```

Ilustración 20 - Función para cargar reportes .csv recopilados.
Fuente: Elaboración propia basada en la práctica

Seguidamente, se establecieron los métodos implementados para la conexión a la base de datos.

```

31
32 # load data
33 con = sqlite3.connect("XXXXXXXXXXXXXXXXXXXX/access.db")
34 con.text_factory = str
35 cursor = con.cursor()
36

```

Ilustración 21 - Conexión a la Base de Datos por sqlite3
Fuente: Elaboración propia basada en la práctica

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

De esta manera, simplemente utilizando la función “to_sql” se creó una tabla dentro de la base de datos con el reporte en cuestión. Como se puede apreciar en la imagen a continuación, se ejecuta la función por cada tabla a crear.

```
54 dfPoliticaATP3.to_sql("POLITICA_ATP3", con)
55 dfPoliticaSOD.to_sql("SOD_POLITICA", con)
56 dfRRHH.to_sql("RRHH", con)
57
58 con.close()
59
60
61 try:
62     SubidaDatosBD()
63 except:
64     mailError("Se genero un ERROR en la Actualizacion de la Base de Datos Local")
65
66
```

Ilustración 22 - Creación de tablas en BD.
Fuente: Elaboración propia basada en la práctica

Por último, para verificar cómo se visualizan los reportes, se ingresó a la base de datos por medio del Browser de SQLite. De esta manera, cada uno de los reportes pueden ser accedidos y manipulados por medio de la ejecución de una consulta SQL.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

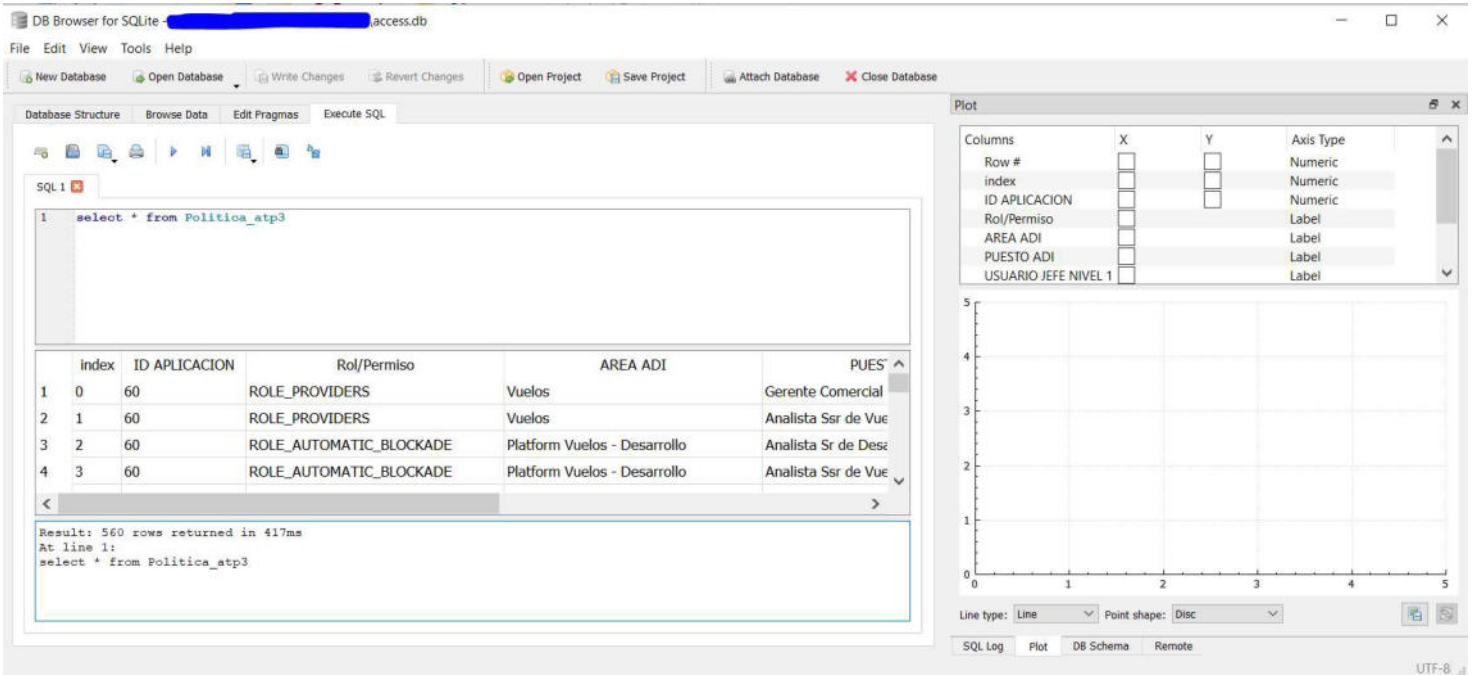


Ilustración 23 - Ejemplo de vista de reporte en Browser de SQLite.
Fuente: Elaboración propia basada en la práctica

3.7.5 Ejecución de verificación de Accesos y SOD

Ya que fue posible contar con la información de las fuentes unificadas en una base de datos local, se llegó a la conclusión de que la mejor opción era aprovechar el motor que ofrece la bd relacional para llevar a cabo el mapeo de cada revisión.

En primer lugar, para la verificación de acceso correctos e incorrectos según el puesto de los usuarios en la compañía, se confeccionó una consulta SQL por cada ambiente a examinar, las cuales se basaron en el cruce entre la nómina de la entidad, la política de acceso establecida para cada caso particular y el listado de

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

accesos adquirido de la fuente a analizar. Una vez que se determinaron cada una de las queries, se creó un script de Python por cada consulta a ejecutar, donde se estableció la conexión a la base de datos, la definición de la query, la ejecución de la misma y la confección de un reporte en formato .csv.

```

accessVerifyORACLE.py x
1 # -*- coding: utf-8 -*-
2 import sys
3 reload(sys)
4 sys.setdefaultencoding('utf-8')
5
6 import sqlite3
7 import os
8 import csv
9 from ManejadorError import mailError
10
11 def accessVerifyORACLE():
12     lista = []
13     conn = sqlite3.connect("XXXXXXXXXXXXXXXXXXXXXXX.access.db")
14     cur = conn.cursor()
15     coords = cur.execute("""
16
17     SELECT DISTINCT AO.Usuario, AO.Mail, AO.Respo, RH.[PRIMER_NOMBRE] Nombre,
18     RH.APELLIDO Apellido, RH.DEPARTAMENTO, RH.PUESTO, RH.MANAGER_NAME,
19     RH.TERMINATION_DATE [Fecha de Baja], PO.[Control_50X]
20     FROM ACCESS_ORACLE AO,
21     (SELECT r1.*
22     FROM RRHH r1
23     where r1.TERMINATION_DATE is not null and
24     r1.EMAIL_ADDRESS in (SELECT a1.EMAIL_ADDRESS from rrhh a1 where a1.TERMINATION_DATE is not null
25     EXCEPT
26     SELECT b1.EMAIL_ADDRESS from rrhh b1 where b1.TERMINATION_DATE is null )
27     UNION
28     SELECT r2.* FROM RRHH r2
29     WHERE r2.TERMINATION_DATE is null) RH,
30     POLITICA_ORACLE PO
31     WHERE AO.Mail = RH.[EMAIL_ADDRESS]
32     AND AO.Respo = PO.[Responsabilidad_Oracle]
33     AND ((RH.DEPARTAMENTO not in ( SELECT PO.[AREA_ADI] from POLITICA_ORACLE PO WHERE AO.Respo = PO.[Responsabilidad_Oracle])
34     AND RH.PUESTO not in ( SELECT PO.[PUESTO_ADI] from POLITICA_ORACLE PO WHERE AO.Respo = PO.[Responsabilidad_Oracle])
35     OR (RH.DEPARTAMENTO in ( SELECT PO.[AREA_ADI] from POLITICA_ORACLE PO WHERE AO.Respo = PO.[Responsabilidad_Oracle])
36     AND RH.PUESTO not in ( SELECT PO.[PUESTO_ADI] from POLITICA_ORACLE PO WHERE AO.Respo = PO.[Responsabilidad_Oracle])
37     OR (RH.DEPARTAMENTO not in ( SELECT PO.[AREA_ADI] from POLITICA_ORACLE PO WHERE AO.Respo = PO.[Responsabilidad_Oracle])
38     AND RH.PUESTO in ( SELECT PO.[PUESTO_ADI] from POLITICA_ORACLE PO WHERE AO.Respo = PO.[Responsabilidad_Oracle])
39     OR RH.TERMINATION_DATE IS NOT NULL)
40
41     """).fetchall()
42     results = coords
43
44     for row in results:

```

Ilustración 23 - Ejemplo de ejecución de Query para verificación de Accesos
Fuente: Elaboración propia basada en la práctica

Seguidamente, para la verificación de Segregación de Funciones, se ejecutaron los mismos pasos. Sin embargo fue necesario contar con una tabla intermedia que

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

unifique los accesos sin importar el aplicativo en cuestión. Por lo tanto, se creó un listado con la siguiente estructura de datos: Usuario-Permiso-Applicativo. Entonces, nuevamente se confeccionó un script de Python con las mismas funciones mencionadas anteriormente.

3.7.6 Disponibilización de Reportes

Una vez generados los reportes con los análisis correspondientes, se tuvo que definir el método más óptimo para que puedan ser utilizados por los usuarios.

Aprovechando las credenciales creadas en el proyecto de Google para obtener las políticas, se realizó un script para consumir el archivo .json “secret” y conectarse planillas de cálculo de Google (previamente creadas) por medio de sus respectivos “id’s”. A partir de allí, se crearon una serie de funciones para obtener los reportes generados (Análisis de accesos a aplicativos y verificación SOD) en formato .csv, insertar las cabeceras requeridas en cada uno, y actualizar los datos en todos Gsheets afectados.

```
def driveUploadSOD():
    now = datetime.now()
    fecha = str(now)[0:7]

    SCOPES = 'https://www.googleapis.com/auth/drive'
    store = file.Storage('storage.json')
    creds = store.get()
    if not creds or creds.invalid:
        flow = client.flow_from_clientsecrets('client_secret.json', SCOPES)
        creds = tools.run_flow(flow, store)
    DRIVE = discovery.build('drive', 'v2', http=creds.authorize(Http()))

    FILES = [
        ('[redacted]BadAccesSOD.csv', True),
    ]
    body = {'title': 'SOD ORACLE - ATP3 - ATP1'}

    body['parents'] = [{'id': '[redacted]'}]

    for filename, convert in FILES:
        res = DRIVE.files().update(fileId='[redacted]', body=body, media_body=filename).execute()
        if res:
            print('Uploaded "%s" (%s)' % (filename, res['mimeType']))
```

Ilustración 24 – Ejemplo de Script de subida de reporte SOD a GSheet.

Fuente: Elaboración propia basada en la práctica

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

Para definir la organización de los Gsheets, se estableció una estructura de directorios en Drive con el fin de garantizar un fácil acceso a todos los documentos.

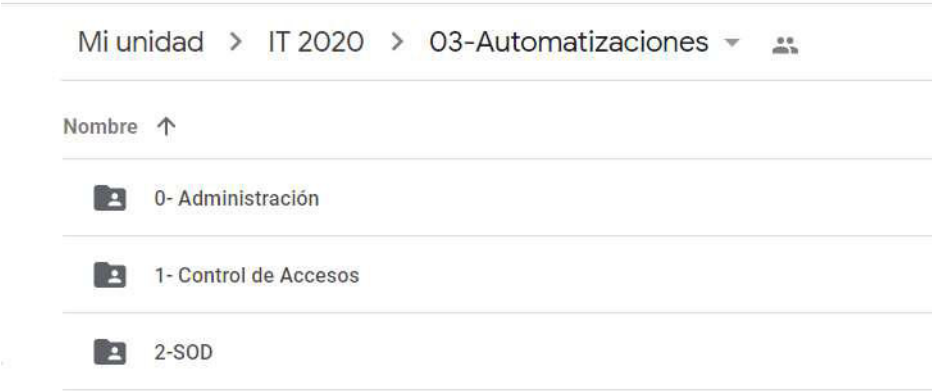


Ilustración 25 – Estructura de Directorios en Drive.
Fuente: Elaboración propia basada en la práctica

Se creó una carpeta raíz nombrada “03-Automatización” con tres subdirectorios:

- “0- Administración”: Contiene dos carpetas, siendo estas: “Políticas”- que contiene todas las políticas utilizadas en la automatización- y “Documentación” - que almacena bibliografía que manifiesta la orquestación y el funcionamiento de la automatización.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------



Ilustración 26 – Contenido de la carpeta “0-Administración”.
Fuente: Elaboración propia basada en la práctica

- “01- Control de Accesos”: está conformada por los reportes de análisis de accesos resultantes de la automatización. Estos ficheros se encuentran divididos por los reportes de asignaciones correctas e incorrectas.

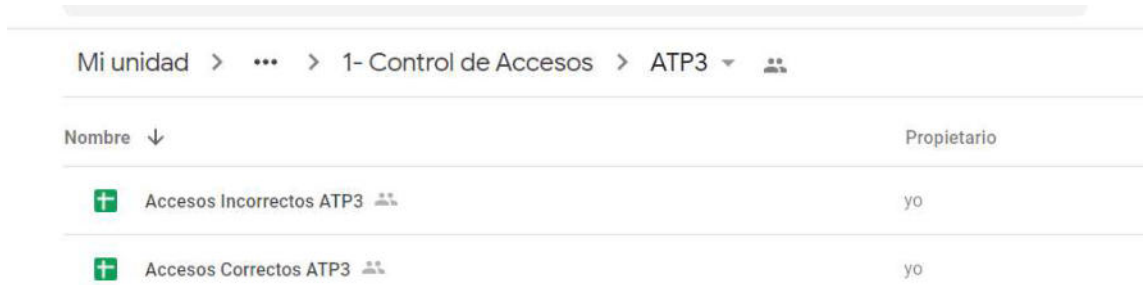


Ilustración 27 – Ejemplo de estructuras de control de accesos según entorno.
Fuente: Elaboración propia basada en la práctica

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

- “02-SOD”: Almacena el reporte semanal de resultado de incompatibilidades de Segregación de Funciones.

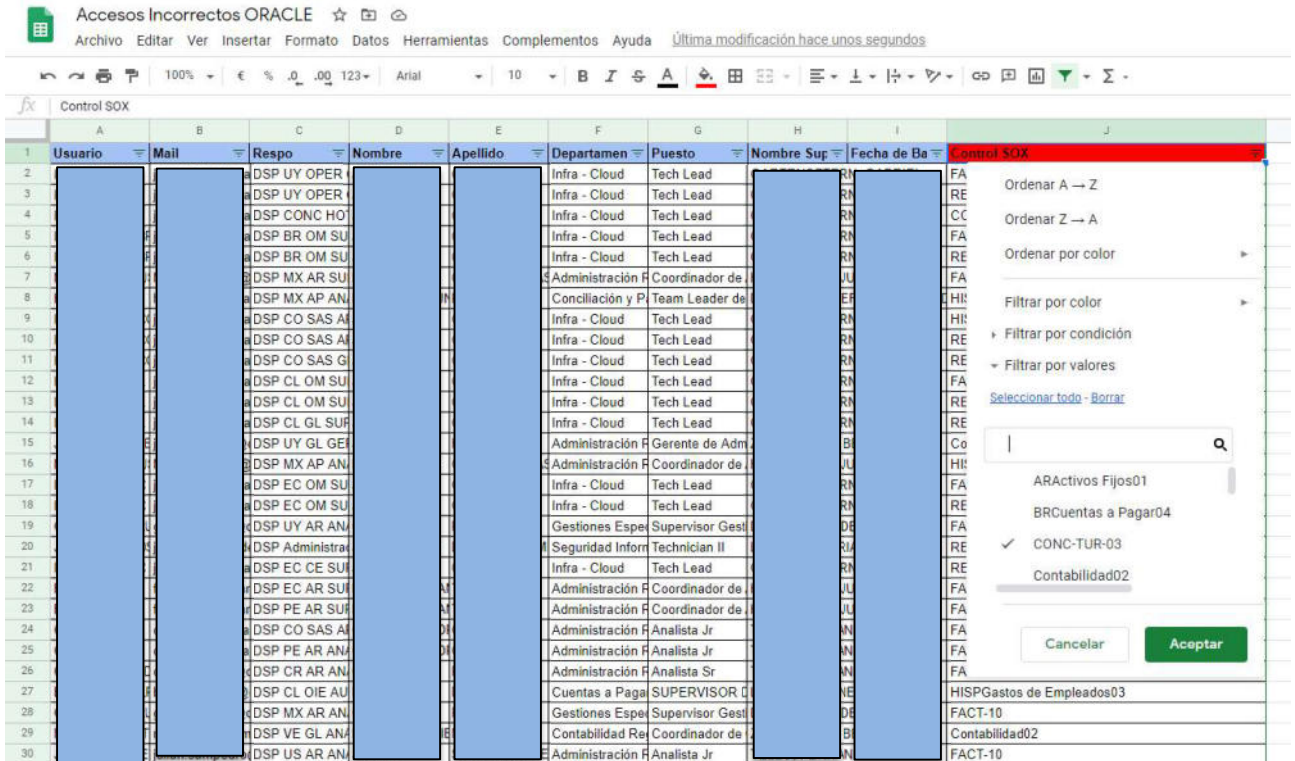


Ilustración 28 – Ejemplo de estructuras de control de incompatibilidades SOD.
Fuente: Elaboración propia basada en la práctica

Cabe aclarar que se estableció que todos los usuarios que puedan acceder a los directorios y a sus reportes cuenten con permisos de lectura sobre los mismos, exceptuando a Gerentes y Coordinadores del área de Auditoría. ya que se les atribuyó permisos de edición sobre todos los archivos de la carpeta “Políticas”.

De esta manera, si un analista requiere realizar un control de accesos en la herramienta Oracle, debe dirigirse a la ruta “1- Control de Accesos” -> “Oracle” y abrir el reporte “Accesos Incorrectos Oracle”. Luego, a la columna “Control SOX” y la filtrar por el control que se encuentre testeando. De este modo, contará con la lista de los usuarios que no cumplen con las condiciones definidas en la Política de accesos para dicho control.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------



Accesos Incorrectos ORACLE ☆ 📄 🌐

Archivo Editar Ver Insertar Formato Datos Herramientas Complementos Ayuda Última modificación hace unos segundos

Control SOX

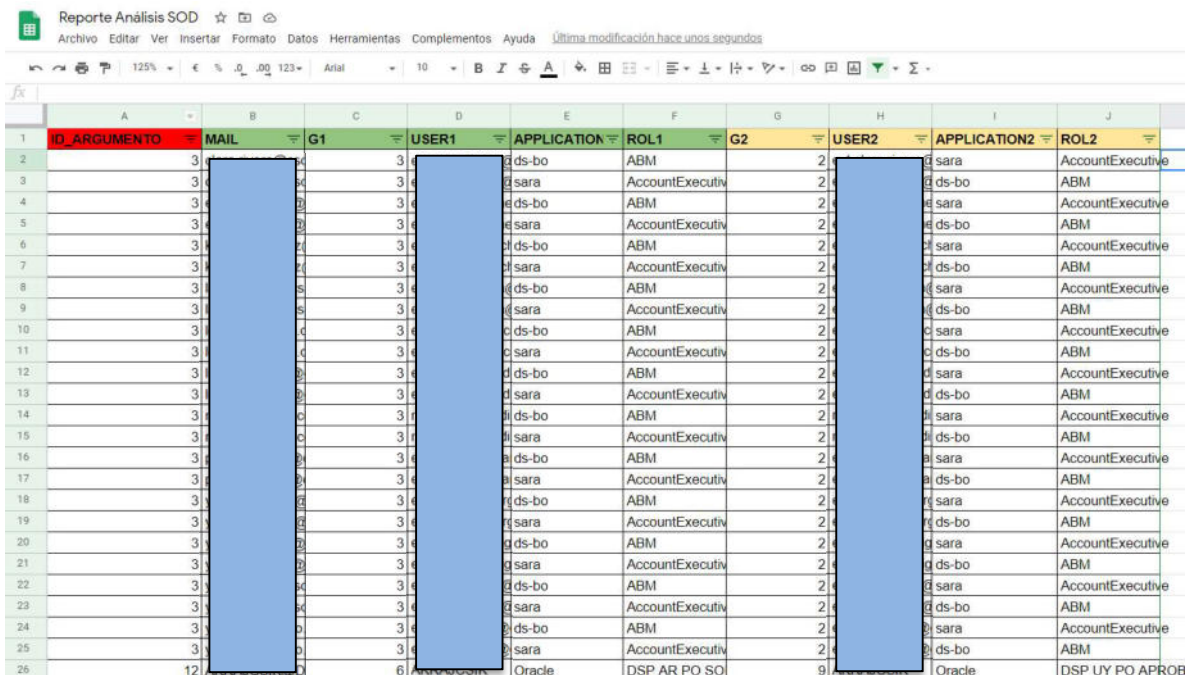
1	Usuario	Mail	Respo	Nombre	Apellido	Departamen	Puesto	Nombre Sup	Fecha de Ba	Control SOX
2			DSP UY OPER			Infra - Cloud	Tech Lead			FA
3			DSP UY OPER			Infra - Cloud	Tech Lead			RE
4			DSP CONC HO			Infra - Cloud	Tech Lead			CC
5			DSP BR OM SU			Infra - Cloud	Tech Lead			FA
6			DSP BR OM SU			Infra - Cloud	Tech Lead			RE
7			DSP MX AR SU			Administración	Coordinador de			FA
8			DSP MX AP AN			Conciliación y P	Team Leader de			HR
9			DSP CO SAS AI			Infra - Cloud	Tech Lead			RE
10			DSP CO SAS AI			Infra - Cloud	Tech Lead			RE
11			DSP CO SAS GI			Infra - Cloud	Tech Lead			RE
12			DSP CL OM SU			Infra - Cloud	Tech Lead			FA
13			DSP CL OM SU			Infra - Cloud	Tech Lead			RE
14			DSP CL GL SUR			Infra - Cloud	Tech Lead			RE
15			DSP UY GL GE			Administración	Gerente de Adm			Co
16			DSP MX AP AN			Administración	Coordinador de			UL
17			DSP EC OM SU			Infra - Cloud	Tech Lead			FA
18			DSP EC OM SU			Infra - Cloud	Tech Lead			RE
19			DSP UY AR AN			Gestiones Espe	Supervisor Gest			DE
20			DSP Administr			Seguridad Infor	Tecnician II			RE
21			DSP EC CE SU			Infra - Cloud	Tech Lead			RE
22			DSP EC AR SU			Administración	Coordinador de			FA
23			DSP PE AR SU			Administración	Coordinador de			FA
24			DSP CO SAS AI			Administración	Analista Jr			FA
25			DSP PE AR AN			Administración	Analista Jr			FA
26			DSP CR AR AN			Administración	Analista Sr			FA
27			DSP CL OIE AU			Cuentas a Paga	SUPERVISOR D			NE
28			DSP MX AR AN			Gestiones Espe	Supervisor Gest			DE
29			DSP VE GL AN			Contabilidad Re	Coordinador de			BI
30			DSP US AR AN			Administración	Analista Jr			IN

Ilustración 29 – Ejemplo de uso del reporte de Accesos Incorrectos.
Fuente: Elaboración propia basada en la práctica

De igual manera, si un analista requiere relevar un control de incompatibilidades de permisos asignados a un proceso específico, debe dirigirse a la carpeta “02-SOD” e ingresar al Reporte de Análisis de SOD. Allí debe seleccionar la columna “ID_ARGUMENTO” y filtrar por el ID del proceso definido en la Política de SOD (como por ejemplo “Compras”). De esta manera, se puede obtener el mail del usuario que genera el conflicto, junto a la funcionalidad que se le asignó en un grupo “G1” (como por ejemplo “Gestión Solicitudes de Compra”), asociado a los aplicativos y roles que permiten ejecutar dicha funcionalidad.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

A su vez, estos datos se encuentran acompañados de la funcionalidad que genera la incompatibilidad establecida en un Grupo “G2” (como es el caso de “Aprobación de Orden de Compra”), donde de igual manera se da detalle de los aplicativos y roles vinculados.



ID_ARGUMENTO	MAIL	G1	USER1	APPLICATION	ROL1	G2	USER2	APPLICATION2	ROL2
2				ds-bo	ABM			sara	AccountExecutive
3				sara	AccountExecutive			ds-bo	ABM
4				ds-bo	ABM			sara	AccountExecutive
5				sara	AccountExecutive			ds-bo	ABM
6				ds-bo	ABM			sara	AccountExecutive
7				sara	AccountExecutive			ds-bo	ABM
8				ds-bo	ABM			sara	AccountExecutive
9				sara	AccountExecutive			ds-bo	ABM
10				ds-bo	ABM			sara	AccountExecutive
11				sara	AccountExecutive			ds-bo	ABM
12				ds-bo	ABM			sara	AccountExecutive
13				sara	AccountExecutive			ds-bo	ABM
14				ds-bo	ABM			sara	AccountExecutive
15				sara	AccountExecutive			ds-bo	ABM
16				ds-bo	ABM			sara	AccountExecutive
17				sara	AccountExecutive			ds-bo	ABM
18				ds-bo	ABM			sara	AccountExecutive
19				sara	AccountExecutive			ds-bo	ABM
20				ds-bo	ABM			sara	AccountExecutive
21				sara	AccountExecutive			ds-bo	ABM
22				ds-bo	ABM			sara	AccountExecutive
23				sara	AccountExecutive			ds-bo	ABM
24				ds-bo	ABM			sara	AccountExecutive
25				sara	AccountExecutive			ds-bo	ABM
26	12		6	Oracle	DSP AR PO SO		9	Oracle	DSP UY PO APROB

Ilustración 30 – Ejemplo de uso del reporte de Análisis de Incompatibilidades SOD.
Fuente: Elaboración propia basada en la práctica

En el caso de que el analista requiera realizar una modificación en algunas de las políticas, debe informar al gerente/coordinador responsable del proceso y definir una propuesta de cambio. De este modo, el responsable analizará dicha propuesta y en el caso de considerarla correcta, debe efectuar la modificación que se verá impactada en la siguiente ejecución de la automatización.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

3.7.7 Manejo de Errores

Los errores pueden producirse por varios motivos, como por ejemplo: si una base de datos cambia de host, entonces la automatización intentará conectarse sin éxito apuntando a la dirección seteada en su configuración.

Por ese motivo, se implementó una serie de alertas que notifiquen las generadas en la ejecución de alguna de las aristas de la automatización. Para ello, se creó un script con una funcionalidad que permite realizar envíos de mails. La misma es invocada en cada uno de los scripts que conforman la automatización a modo de excepción. Esto quiere decir que cada script intentará ejecutar las tareas que posee configuradas, pero en el caso de no poder hacerlo, entrará en excepción y enviará un mail a la casilla de correo del equipo de Auditoría IT para notificar la falla puntual, con el fin de que posteriormente este error pueda ser analizado y remediado por el mismo equipo.

A su vez, también configuró la llamada de dicha funcionalidad para alertar el comienzo y el fin de la ejecución de la automatización.

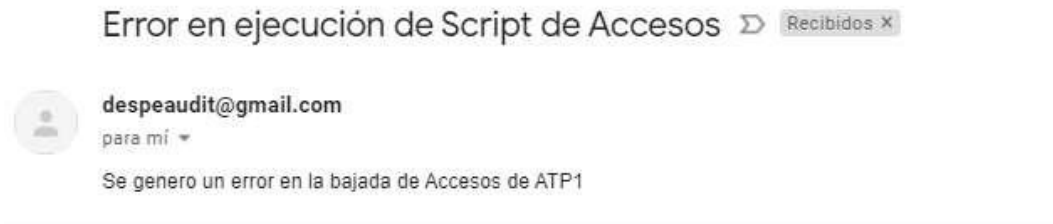


Ilustración 31 – Ejemplo de notificación de error en ejecución.
Fuente: Elaboración propia basada en la práctica

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------



Ilustración 32 – Ejemplo de notificación de comienzo y fin de ejecución.
Fuente: Elaboración propia basada en la práctica

3.7.8 Orquestación de ejecución de Scripts

Finalmente, como se mencionó a lo largo del informe, se definió que la ejecución de la automatización tenga con una periodicidad semanal.

De esta manera, ya que la infraestructura que almacena los scripts corresponde a un cluster que cuenta con un Linux como sistema operativo, y no fue posible utilizar aplicativo que orqueste las funciones (como por ejemplo “Azkaban”), se unificaron las llamadas de todos los desarrollos en único script, al cual se configuró su scheduling utilizando el Crontab del sistema para su ejecución.

Por lo tanto, se especificó que la automatización se ejecute todos los días lunes a las 8 am.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
-------------------	---------------------------	----------------------------	-----------------------------

4 Conclusión

4.1 Resultado, posibles mejoras y ampliaciones:

Luego de varias semanas de uso, el equipo comenzó a familiarizarse con los resultados obtenidos y se ajustaron puntos que surgieron a medida que las corridas se ejecutaron, como por ejemplo: ajustes en la información definida en la política, o validaciones que debían tenerse en consideración en las queries establecidas (como por ejemplo la necesidad de una verificación de usuarios que ya no pertenezcan a la compañía y se encuentren activos en aplicativos bajo scope). Gracias al diseño amigable uso del automatismo, los analistas pudieron realizar un estudio mas concreto de sus pruebas y eliminar falsos positivos que se obtenían constantemente en cada revisión. A su vez, como la ejecución del automatismo es semanal, se pudo verificar como fue mejorando gradualmente la tasa de errores de accesos reportados a los responsables por área. Esto no sólo generó impacto a nivel de tiempos de revisiones, sino también en calidad e integridad de pruebas llevadas a cabo.

Sin embargo, existen algunas modificaciones que pueden mejorar la experiencia de uso de la automatización y deberían ser implementadas en un futuro:

- Generar una interfaz visual con diferentes tipos de permisos (como por ejemplo: editor de matrices, ejecutor de corridas, vista completa, etc) que permita hacer la experiencia de usuario más amigable para el mismo.
- Generar informes que comuniquen automáticamente a los responsables de remediaciones.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

- Generar tableros que permitan verificar visualmente las tasas de errores semanales, el avance de remediaciones, acumulado de desvíos por responsable, etc.
- Generar logs de resultados que permitan ser almacenados durante periodos de revisión (12 meses) para contar pistas de auditorías en los periodos de revisión requeridos.

4.2 Reflexión sobre las prácticas profesionales como espacio de formación:

La Práctica Profesional Supervisada fue una posibilidad que me permitió poner en práctica los conocimientos técnicos aprendidos en la carrera universitaria para la aplicación de una solución que permita el ahorro de costos, tiempos de procesamiento de trabajo, ganar madurez y promocionar al área de Auditoría dentro de Despegar.

Tuve la oportunidad de desarrollarla en el puesto de trabajo en el que me desempeño y pude experimentar en un escenario real situaciones como la definición de requerimientos y de etapas de desarrollo, comprobación de la evolución de las mismas, investigación de herramientas a aplicar, ejecución de pruebas, análisis de resultados junto a clientes internos, etc. Todo esto me hubiese sido imposible llevar a cabo sin la PPS como intermediaria ya que el área en donde me desarrollo no tiene como fin ejecutar proyectos semejantes. Esto es así porque el core de trabajo es de consultoría de control al cliente interno de la Entidad (como áreas de desarrollo, dba's, áreas de seguridad informática, etc.) y las tareas diarias

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

corresponden a ejecución de proyectos donde se pone en práctica otras habilidades (como por ejemplo las blandas).

Gracias a la implementación del proyecto, me fue posible demostrar al área que la aplicación de automatizaciones y el cambio a un paradigma de trabajo más esquematizado permite aplicar soluciones a largo plazo. Si bien es necesario invertir horas de trabajo en la implementación de las mismas, una vez concluidas las tareas de desarrollo, en el futuro solamente es requerido invertir horas en la corrección de errores en procesos de la Entidad (como por ejemplo los procesos correspondientes a bajas de usuarios o asignación de permisos). Es decir, se puede hacer solamente énfasis en la verificación de la causa de los errores y la solución de los mismos.

En cuanto al desarrollo, al comienzo se me presentaron algunas dificultades para esquematizar el rumbo del proyecto, ya que durante la pandemia fue necesario cubrir otras necesidades del área requeridas, pero con la ayuda de mis tutores asignados pude re direccionar el rumbo de la PPS y adaptarla para que el proyecto fuera viable y digno de presentarse y, a su vez, brinde una solución real al área de Auditoría. Durante el transcurso del mismo, también pude incrementar mi conocimiento técnico de desarrollo y utilizar un razonamiento lógico para la aplicación de soluciones, los cuales muchas veces fueron logrados gracias al feedback obtenido por otras áreas de desarrollo.

En conclusión, la PPS aportó un gran valor agregado a mi desarrollo personal y profesional. Logré hacerme cargo de un desarrollo de punta a punta, es decir, desde el relevamiento de la necesidad hasta la implementación de las soluciones mencionadas a lo largo del proyecto.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

5. ANEXO

5.1 Bibliografía

- Allende, Sandra (2019) SISTEMA OPERATIVO LINUX Teoría y Práctica 2da. Edición, Ciudad Autónoma de Buenos Aires, edUTecNe.
- Blue Lance, Inc (2012) Sarbanes-Oxley Control Transformation Through Automation. Recuperado en junio de 2020, de https://bluelance.com/wp-content/uploads/2015/02/SOX_ControlTransformation.pdf
- Buturla, Ewelina (06/2020) How To Become A Task Automation Hero Using Python [With Examples]. Recuperado el 03 de junio de 2020, de <https://www.monterail.com/blog/python-task-automation-examples.html>
- Deloitte (2004) Sarbanes-Oxley Section 404: 10Threats to Compliance. Recuperado en junio de 2020, de <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/audit/us-aers-assur-ten-threats-sep2004.pdf>
- Desai, Suhas (04/2019) Database Programming in Python. Recuperado el 10 de junio de 2020, de <https://www.opensourceforu.com/2019/04/database-programming-python.html>
- Domínguez, Pablo (06/2019) En qué consiste el modelo en cascada. Recuperado el 18 de abril de 2020, de <https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada.html>
- Gentoo Foundation Inc. (01/2019) Cron. Recuperado el 19 de junio de 2020, de <https://wiki.gentoo.org/wiki/Cron/es.html>
- Gilfillan, Ian (01/2003) La Biblia MySQL, España, Anaya Multimedia-Anaya

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:

- Google Docs for Developers (2020) Google Python API. Recuperado el 05 de mayo de 2020, de <https://developers.google.com/docs/api/quickstart/python.html>
- Knowlton, Jim (01/2009) Python Crear-Modificar-Reutilizar, Estados Unidos, Anaya Multimedia
- PandasTeam (04/2020) Pandas Documentation User Guide. Recuperado el 05 mayo de 2020, de <https://pandas.pydata.org/pandas-docs/stable/#pandas-documentation.html>
- Sommerville, Ian (1988) Software Engineering | 9th Edition, Inglaterra, Pearson.
- SQLite Org (2020) SQLite Documentation. Recuperado en abril de 2020, de <https://www.sqlite.org/docs.html>

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional: