

Informe Practica Acad émica en IBM Colombia

Orlando Fabio Correa Vecino

ID: U00072107

Universidad Autónoma De Bucaramanga

Practica Académica

IBM

Bogotá

El documento acá redactado, fue leído, analizado y evaluado con el fin de dar autorización a la publicación de información que aquí se contenga.

Firma Docente encargado practica académica

---

M.Sc. HERNANDO GONZALEZ ACEVEDO

## TABLA DE CONTENIDO

Tabla de contenido .....	3
Tabla de Figuras .....	4
1. INTRODUCCION.....	6
2. OBJETIVOS .....	7
2.1 Objetivos Generales.....	7
2.2 Objetivos Específicos .....	7
3. MARCO CONCEPTUAL.....	8
3.1 Módulos de Comunicación Remota a Bases de Datos – Python.....	8
3.2 Análisis de Datos - Python .....	8
3.3 MySQL .....	9
3.4 Automatización en Bash – Lenguaje de Consola.....	9
3.5 SAP .....	9
4. ACTIVIDADES.....	10
4.1 Administración Sistemas SAP .....	10
4.2 DB-Monitoring .....	11
4.3 Automatización Update-Kernel.....	16
4.4 Automatización Monitoreo Replicas: .....	18
5. CONCLUSIONES.....	21
ANEXOS.....	22

## TABLA DE FIGURAS

Figura 1. Sentencia SQL - Lectura DB Oracle .....	12
Figura 2. Función de Propagación de la Sentencia SQL con todas las DB Oracle administradas.....	12
Figura 3. Función de Propagación, extracción y almacenamiento de datos en DB Db2 .....	13
Figura 4. Comandos de Consola para extracción de datos en DB Sybase .....	14
Figura 5. Función de Proyección para el tiempo de vida en cada de las DB .....	15
Figura 6. Web-Page de monitoreo interno en el área de SAP .....	15
Figura 7. Seccion interna del monitoreo de espacios dentro del Global-Monitoring .....	16
Figura 8. Diagrama de flujo de la lógica implementada en el Monitoreo de replicas .....	19

## TABLA DE ANEXOS

Anexo A.Manual Automatizacion DB-Monitoring .....	23
Anexo B.Manual Automatizacion Monitoreo Replicas.....	29
Anexo C.Manual Automatizacion Actualizacion Kernel.....	34

## 1. INTRODUCCION

IBM es una compañía estadounidense privada que provee soluciones de hardware, entre los que se incluyen computadoras portátiles y de escritorio, y sistemas de mainframes, servidores y almacenamiento, así como software, servicios financieros y una amplia gama de servicios de tecnología de información. Las principales operaciones de la compañía comprenden cinco segmentos: servicios globales de tecnología, servicios de negocios mundiales, software, sistemas y tecnología, y financiamiento global. Además de los servicios de TI, provee servicios de consultoría de negocios, aplicaciones y outsourcing, entre otros. IBM tiene presencia global y opera en más de 170 países, incluidas oficinas en Latinoamérica y el Caribe.

IBM-Colombia hoy en día es mucho más que una compañía de hardware, software y servicios, en la actualidad se enfoca en soluciones cognitivas en plataforma Cloud para las industrias.

Las prácticas académicas otorgadas por la empresa, tienen como fin en primera instancia dar a conocer a la población estudiantil las bases y avances de la nueva era tecnológica y como segunda medida, aprovechar las capacidades de los estudiantes para generar soluciones inteligentes y cognitivas a partir de herramientas tecnológicas y conocimiento.

## 2. OBJETIVOS

### 2.1 Objetivos Generales

- Analizar, crear e innovar desde diferentes perspectivas y de una manera proactiva, técnicas o procedimientos que permitan la mejora continua de los servicios prestados por IBM Colombia.

### 2.2 Objetivos Específicos

- Participar, aprender y apoyar de manera activa las labores administrativas de un especialista SAP Basis.
- Crear, integrar e implementar monitoreo bases de datos Oracle, Sybase y Db2.
- Crear e implementar desarrollo para la automatización de actualización Kernel en servidores SAP.
- Crear, integrar e implementar monitoreo de Replicas en servidores productivos SAP.

## 3. MARCO CONCEPTUAL

### 3.1 Módulos de Comunicación Remota a Bases de Datos – Python

Para el sistema de monitoreo de bases de datos Db2, Oracle y Sybase fue necesario realizar una investigación para determinar de qué manera era eficiente extraer la información necesaria en el monitoreo de cada una de las bases de datos. Se optó por utilizar los módulos de comunicación independientes de cada tipo, desarrollados para ser usados en lenguaje Python.

Módulos de Comunicación:

Ibm\_db2 = Librería desarrollada por IBM para permitir el desarrollo de diversas aplicaciones donde fuese necesario la integración con un tipo de bases de datos DB2-

Mysql\_connect = Librería desarrollada por Oracle para permitir el desarrollo de diversas aplicaciones donde fuese necesario la integración con un tipo de bases de datos Oracle.

Sybase.sh = Librería desarrollada en lenguaje de consola por mi persona, cuyo objetivo es realizar la consulta de bases de datos Sybase e insertar dicha información en la base de datos local para su posterior análisis.

### 3.2 Análisis de Datos - Python

Toda la información almacenada en el monitoreo de cada una de las bases de datos de servidores SAP, es alojada en una base de datos MySQL local. El objetivo del monitoreo de bases de datos es darle una herramienta al especialista que le ayudara a optimizar su rendimiento y atención al cliente. Dicha herramienta emite una alerta anticipada cuando detecta que alguna base de datos esta prontamente a llegar a su límite de capacidad, esto con el fin de evitar incidentes por caída de bases de datos debido a capacidades. Un solo minuto que pase caída una base de datos de un sistema productivo, significa para IBM millones de pesos en pérdidas.

Para lograr establecer una estimación de llenado de acuerdo con la información que se extrae diariamente por el script a todas las bases de datos administradas por IBM en sistemas SAP, se realizó un desarrollo en Python que analiza toda la información contenida.

### **3.3 MySQL**

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

Debido a que toda la información que era recogida por los diferentes tipos de monitoreo estaba siendo almacenada en una base de datos MySQL local en IBM, fue implícitamente necesario aprender a manejar consultas, inserciones, actualizaciones y demás tipos de operaciones que pueden realizarse en la base de datos, para así mismo poder realizar el posterior análisis en Python a esa información.

### **3.4 Automatización en Bash – Lenguaje de Consola**

Bash (Bourne-again-shell) es un programa informático, cuya función consiste en interpretar órdenes, y un lenguaje de consola. Es una Shell de Unix compatible con POSIX y el intérprete de comandos por defecto en la mayoría de las distribuciones GNU/Linux, además de macOS.

Este punto fue vital, dado que la mayoría de las automatizaciones que se lograron realizar en el tiempo de pasantía, fueron hechas en lenguaje de consola utilizando el intérprete de comandos Bash. Es decir, todas las operaciones que se pueden realizar a nivel de sistema operativo por medio de comandos en lenguaje de consola pueden ser automatizadas y ejecutadas por una Shell generada en el intérprete de comandos Bash.

### **3.5 SAP**

Al Sistema SAP se le relaciona con los sistemas ERP (Planificación de Recursos Empresariales), por tratarse de un sistema de información que permite gestionar las diferentes acciones de una empresa, sobre todo las que tienen que ver con la producción, la logística, el inventario, los envíos y la contabilidad. Sus servicios destacan por cubrir, además, las necesidades tanto administrativas como las de sus clientes y proporcionar las herramientas que se necesitan.

SAP es un sistema informático, sirve para brindar información. Se alimenta de los datos que se cargan y procesan dentro de un entorno, y el sistema se encargará (de acuerdo a la configuración realizada por el usuario -consultores SAP) de producir con esos datos información útil para la toma de decisiones y la exposición de esos datos de forma tal que puedan ser interpretados por los interlocutores interesados.

## 4. ACTIVIDADES

### 4.1 Administración Sistemas SAP

#### 4.1.1 Actualización Línea Base SAP:

Actualmente IBM-Colombia cuenta con 40 clientes a escala con sistemas SAP, sumando los sistemas y aplicativos SAP que tienen implementados esos clientes, son alrededor de 1200, es decir, IBM tiene que prestar servicio y disponibilidad del 100% a 1200 sistemas. Por ello se hace indispensable una base de datos que contenga la información técnica y general de todos esos sistemas, para que así todo el personal técnico y especialista pueda acceder y tener claridad del status actual del sistema.

Entre los factores que se tuvieron en cuenta para el análisis y documentación de toda esa información, están los siguientes:

- I. Cliente: Nombre del cliente al que pertenece ese sistema SAP.
- II. Producto: Descripción del producto SAP.
- III. Ambiente: Todo producto SAP viene en tres ambientes Calidad-Desarrollo-Productivo, se implementa el necesario.
- IV. Arquitectura: Arquitectura lógica del funcionamiento del sistema.
- V. Hostnames: Nombre de los servidores donde están implementados los sistemas.
- VI. IP's Asociadas: Direcciones IP asociadas al sistema para su posterior conexión remota en casos de ser necesaria
- VII. Release Versión: Versión del sistema SAP implementado
- VIII. Release Kernel: Versión del Kernel donde funciona en el sistema SAP
- IX. DB\_Version: Versión base de datos asociada al sistema SAP
- X. Puertos de comunicación: Puerto disponible para conexión remota
- XI. Release\_SO: Versión del sistema operativo instalado en el servidor

Para acceder a dicha información, era necesario conectarse al sistema remotamente a nivel de sistema operativo, para que así por medio de una serie de "queries" o códigos de programación dependiendo del sistema operativo, base de datos y sistema SAP implementado, se capturara dicha información. Al ser tantos sistemas, el proceso fue tedioso y requirió de bastante tiempo.

#### **4.1.2 Servicios CQC FOR EWA:**

El control de calidad continuo SAP EarlyWatch Check analiza los componentes de la solución SAP, el sistema operativo y la base de datos para determinar cómo optimizar el rendimiento y mantener su costo total de propiedad al mínimo. El EarlyWatch Alert de SAP puede activarse mediante SAP EarlyWatch Alert, que supervisa de manera automática su solución SAP. Si las alertas vitales se informan a través de SAP EarlyWatch Alert, SAP programa un control de calidad continuo. EarlyWatch Check de SAP para un análisis más detallado del sistema. Los resultados del análisis y las acciones necesarias para optimizar el rendimiento de la solución se enumeran en un informe final entregado por SAP. Para poder solicitar el servicio, es necesario preparar el sistema con 5 semanas de anticipación y hacer seguimiento a todos los requerimientos exigidos por SAP para la liberación de el servicio.

En el transcurso de esta preparación, se hace vital entrar y supervisar constantemente el sistema el cual será analizado por SAP para asegurar una óptima comprensión de datos y una óptima conexión remota de SAP hacia el sistema, ya que se hace desde Alemania. Los pre-requisitos necesarios son labores internas del sistema, tales como:

- I. Configurar Variables de entorno del sistema
- II. Habilitar, configurar o añadir conexiones con puertos seguros de conexión remota autorizados en IBM.
- III. Generar y activar el envío de alertas del sistema SAP hacia SAP, para que, a partir de ello, analicen y generen un informe de mejora de rendimiento.
- IV. Comunicación constante con especialista directo de SAP encargado de liberarle a IBM el servicio.

#### **4.2 DB-Monitoring**

El desarrollo e implementación del monitoreo se dividió en 5 fases:

- I. Crear Modulo Python de lectura bases de datos Oracle e integrar en el script principal.
- II. Crear Modulo Python de lectura bases de datos Sybase e integrar en el script principal.
- III. Crear Modulo Python de lectura bases de datos Db2 e integrar en el script principal.
- IV. Método Proyección
- V. Implementar monitoreo enlazado con GlobalMonitoring

## Fase I:

Para el desarrollo del módulo de lectura de bases de datos Oracle se utilizó como eje principal la librería Cx\_Oracle para Python, proporcionada directamente por Oracle. Ya teniendo la librería para realizar las conexiones remotas, se procede a diseñar la estructura de comandos SQL para la extracción coherente de datos que serán usados por un método de proyección.

Comando SQL:

```

BEGIN
v_crec_ocupado_diario_kbytes := 0;
v_crec_diario_extents := 0;
v_max_creci_ultimos_dias := 0;
FOR tablespaces IN data_files LOOP
v_tablespace_name := tablespaces.tablespace_name;
v_tan_kbytes := tablespaces.tan_kbytes;

SELECT SUM(bytes)/1024, MAX(bytes)/1024
INTO v_libre_kbytes, v_max_bloque_libre
FROM dba_free_space
WHERE tablespace name=v_tablespace_name;
IF v_libre_kbytes IS NULL THEN
v_libre_kbytes :=0;
END IF;
IF v_max_bloque_libre IS NULL THEN
v_max_bloque_libre :=0;
END IF;
SELECT SUM(bytes)/1024, SUM(extents)
INTO v_ocupado_kbytes, v_extents
FROM dba_segments
WHERE tablespace_name=v_tablespace_name;
IF v_ocupado_kbytes IS NULL THEN
v_ocupado_kbytes :=0;
END IF;
IF v_extents IS NULL THEN
v_extents :=0;
END IF;

```

Figura 1. Sentencia SQL - Lectura DB Oracle

En la figura 1 se denota el Comando SQL que extrae directamente de la base de datos Oracle los diferentes tipos de “tablespaces” o por decirlo de alguna manera, las diferentes particiones en las que está dividida la base de datos junto con sus valores de espacio usado y espacio libre respectivamente.

Función de Lectura Base de Datos Oracle:

```

def connect_Oracle(datype, customer, dbname, dbalias, dbuser, dbpass, sid, hostname, dport, usertunn, hostunn, portunn, umbralert):
inserts=[]
if portunn[1]==1:
a=os.system('sh /home/coecogni/espacios/tunnel.sh '+ 'umbralert'+ 'portunn'+ 'hostname'+ 'dport'+ 'usertunn'+ 'hostunn')
dsn=cx_Oracle.makedsn(host='localhost', sid=sid, port=portunn)
else:
dsn=cx_Oracle.makedsn(host=hostname, sid=sid, port=dport)
con=cx_Oracle.connect(user=dbuser, password=dbpass, dsn=dsn)
cur=con.cursor()
sql=open('/home/coecogni/espacios/trae_data_limpio.sql','r')
data=sql.read()
cur_out=cur.var(cx_Oracle.STRING)
cur.execute(data,cur_out)
data_out=cur_out.getvalue()
data_split=data_out.split('\n')
data_split2 = [x for x in data_split if x]
for i in data_split2:
data_split=i.split(',')
c_tablespace=data_split[0]
c_tamano=data_split[1]
c_libre=data_split[2]
c_ocupado=data_split[3]
c_maxbloqlibre=data_split[4]
c_crecdia=data_split[5]
c_extends=data_split[6]
c_crecdiaext=data_split[7]
c_crecultimosdia=data_split[8]
sql = "INSERT INTO espacios_data (CLIENTE, TIPO DB, FECHA, SID, DESCRIPCION, TABLESPACE, TAN_KBYTES, LIBRE_KBYTES, OCUPADO_KBYTES, MAX_BLOQUE_LIBRE, CREC_OCUPADO_DIAR
cur_sql_1.execute(sql,(customer,datype,time,sid,dbname,c_tablespace,c_tamano,c_libre,c_ocupado,c_maxbloqlibre,c_crecdia,c_extends,c_crecdiaext,c_crecultim
con.close()
return

```

Figura 2. Función de Propagación de la Sentencia SQL con todas las DB Oracle administradas

En la figura 2 se puede apreciar la función de propagación, la cual es la encargada de conectarse, ejecutar la sentencia SQL y almacenar los datos extraídos en una base de datos local, todo esto se hace individualmente a cada una de las base de datos Oracle administradas por IBM-Colombia.

## Fase II:

Para el desarrollo del módulo de lectura de bases de datos DB2 se usó la librería para Python IBM\_DB2 proporcionada directamente por IBM, generando así un alcance fácil y dinámico en la extracción de la información requerida de manera remota.

La librería ya viene con una serie de comandos de uso que facilitan aún más la extracción de la información:

```
def connect_db2(dbtype, customer, dbname, dbuser, dbpass, dbalias, sid, hostname, dport):
    try:
        # Create connection to remote DB2
        con = ibm_db.connect("DATABASE="+sid+";HOSTNAME="+hostname+";PROTOCOL=TCPIP;UID="+dbuser+";PWD="+dbpass+";PORT="+dport+";")
    except:
        # Connection error handling
        print(ibm_db.conn_errormsg())
        # TODO make error handling for connection
    else:
        try:
            # Execute query to get space
            stmt = ibm_db.exec_immediate(con, "select DB_STORAGE_PATH,FS_TOTAL_SIZE,STO_PATH_FREE_SIZE,FS_USED_SIZE from sysibmado.SNAPSTORAGE_PATHS")
        except:
            # Query error handling
            print(ibm_db.stmt_errormsg())
            # TODO make error handling for query
        else:
            # Fetch first result
            result = ibm_db.fetch_tuple(stmt)
            inserts = []
            while (result):
                # TODO whatever needed with the result
                inserts.append(("'" + customer + "'", "' + dbtype + "'", "' + time + "'", "' + sid + "'", "' + dbname + "'", "' + str(result[0]) + "'", "' + str(float(result[1])/1000.0) + "'",
                                str(float(result[2])/1000.0) + "'", "' + str(float(result[3])/1000.0) + "'", "0,0,0,0,0)")
                result = ibm_db.fetch_tuple(stmt)
            stmt = "INSERT INTO espacios_stats values " + ', '.join(inserts) + ";"
            cur_sql.execute(stmt)
            con.commit()
    return
```

Figura 3. Función de Propagación, extracción y almacenamiento de datos en DB Db2

En la figura 3 se aprecia la función encargada de conectarse a las bases de datos Db2, ejecutar la sentencia de extracción de datos y almacenar esos datos en una base de datos local para posteriormente realizarle su análisis.

## Fase III:

Para el desarrollo del módulo de lectura de bases de datos Sybase, se optó por realizarlo mediante el lenguaje de consola con el intérprete de comandos Bash, dado que las librerías proporcionadas para la conexión remota a bases de datos Sybase son muy inestables en python.

Mediante un script Shell se realizó el desarrollo que conecta a la base de datos, extrae e inserta la información necesaria en el Backus de monitoreo.

```

DATABASE_SIZE_sid=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_sid | awk ' FNR == 3 { print $(2*1000) } ' ")
DATABASE_USED_sid=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_sid | awk ' FNR == 8 { print $(1) } ' ")
DATABASE_FREE_sid=$(awk '{print $(1-$2)}' <<<"DATABASE_SIZE_sid $DATABASE_USED_sid");

DATABASE_SIZE_master=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_master | awk ' FNR == 3 { print $(2*1000) } ' ")
DATABASE_USED_master=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_master | awk ' FNR == 8 { print $(1) } ' ")
DATABASE_FREE_master=$(awk '{print $(1-$2)}' <<<"DATABASE_SIZE_master $DATABASE_USED_master");

DATABASE_SIZE_model=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_model | awk ' FNR == 3 { print $(2*1000) } ' ")
DATABASE_USED_model=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_model | awk ' FNR == 8 { print $(1) } ' ")
DATABASE_FREE_model=$(awk '{print $(1-$2)}' <<<"DATABASE_SIZE_model $DATABASE_USED_model");

DATABASE_SIZE_saptools=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_saptools | awk ' FNR == 3 { print $(2*1000) } ' ")
DATABASE_USED_saptools=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_saptools | awk ' FNR == 8 { print $(1) } ' ")
DATABASE_FREE_saptools=$(awk '{print $(1-$2)}' <<<"DATABASE_SIZE_saptools $DATABASE_USED_saptools");

DATABASE_SIZE_sybmgtdb=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_sybmgtdb | awk ' FNR == 3 { print $(2*1000) } ' ")
DATABASE_USED_sybmgtdb=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_sybmgtdb | awk ' FNR == 8 { print $(1) } ' ")
DATABASE_FREE_sybmgtdb=$(awk '{print $(1-$2)}' <<<"DATABASE_SIZE_sybmgtdb $DATABASE_USED_sybmgtdb");

DATABASE_SIZE_sybssystemdb=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_sybssystemdb | awk ' FNR == 3 { print $(2*1000) } ' ")
DATABASE_USED_sybssystemdb=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_sybssystemdb | awk ' FNR == 8 { print $(1) } ' ")
DATABASE_FREE_sybssystemdb=$(awk '{print $(1-$2)}' <<<"DATABASE_SIZE_sybssystemdb $DATABASE_USED_sybssystemdb");

DATABASE_SIZE_sybssystemprocs=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_sybssystemprocs | awk ' FNR == 3 { print $(2*1000) } ' ")
DATABASE_USED_sybssystemprocs=$(echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " cat $RUTE/output_file_sybssystemprocs | awk ' FNR == 8 { print $(1) } ' ")
DATABASE_FREE_sybssystemprocs=$(awk '{print $(1-$2)}' <<<"DATABASE_SIZE_sybssystemprocs $DATABASE_USED_sybssystemprocs");

echo $line | SSHPASS=$pass unix sshpass -e $pass ssh $user unixqslip " rm $RUTE/output_file_sid $RUTE/output_file_master $RUTE/output_file_model $RUTE/output_file_saptools $RUTE/autp
echo $cliente,"$typondb","$FECHA","$sid","$descripcion","$master","$DATABASE_SIZE_master","$DATABASE_FREE_master","$DATABASE_USED_master" >> todos_db2
echo $cliente,"$typondb","$FECHA","$sid","$descripcion","$model","$DATABASE_SIZE_model","$DATABASE_FREE_model","$DATABASE_USED_model" >> todos_db2
echo $cliente,"$typondb","$FECHA","$sid","$descripcion","$saptools","$DATABASE_SIZE_saptools","$DATABASE_FREE_saptools","$DATABASE_USED_saptools" >> todos_db2
echo $cliente,"$typondb","$FECHA","$sid","$descripcion","$sybmgtdb","$DATABASE_SIZE_sybmgtdb","$DATABASE_FREE_sybmgtdb","$DATABASE_USED_sybmgtdb" >> todos_db2
echo $cliente,"$typondb","$FECHA","$sid","$descripcion","$sybssystemdb","$DATABASE_SIZE_sybssystemdb","$DATABASE_FREE_sybssystemdb","$DATABASE_USED_sybssystemdb" >> todos_db2
echo $cliente,"$typondb","$FECHA","$sid","$descripcion","$sybssystemprocs","$DATABASE_SIZE_sybssystemprocs","$DATABASE_FREE_sybssystemprocs","$DATABASE_USED_sybssystemprocs" >> todos_db2

```

Figura 4. Comandos de Consola para extracción de datos en DB Sybase

En la figura 4 se denota el script que ejecuta una serie de comandos que proporcionan la información requerida dentro de cada sistema al que está conectado. Desde python se realiza el llamado al script con la función creada para la lectura de bases de datos Sybase.

#### Fase IV:

La lectura de bases de datos corre todos los días a las 4:30 a.m por lo que diariamente se va almacenando los datos que se extraen y así mismo generando un histórico de uso. Al tener un histórico de uso, se permite realizar análisis estadísticos de tal manera que se genere una proyección donde se pueda estimar el tiempo de vida o de llenado que le queda a cada una de las bases de datos monitoreadas.

El método de proyección captura los datos de cada sistema para los últimos 7 días, teniendo los datos se procede a realizar el cálculo del delta de uso:

$$\text{Delta\_uso\_1} = \text{uso\_hoy} - \text{uso\_ayer}$$

$$\text{Delta\_uso\_2} = \text{uso\_ayer} - \text{uso\_antes\_de\_ayer}$$

Así secuencialmente se calculan todos los deltas de uso, para que posteriormente se estime el máximo de los deltas capturados en esos últimos 7 días, hallando de tal manera el tiempo de vida:

$$\text{Tiempo\_de\_vida} = \text{Espacio\_libre\_database} / \text{Maximo\_delta\_uso}$$

```

for q in range(len(cliente_ready)):
    if list_max_delta[q]==0:
        cliente_ready_used_null.append(cliente_ready[q])
        descripcion_ready_used_null.append(descripcion_ready[q])
        tablespace_ready_used_null.append(tablespace_ready[q])
        list_time_for_fullspace_null.append("NULL")
        list_time_for_fullspace_complete.append(cliente_ready[q]+"*tipo_db_ready[q]+"*descripcion_ready[q]+"*list_name_alias[q]+"*list_name_sid[q]+"*tablespace_ready[q]+"
list_name_hostname[q]+"*list_name_ip[q]+"*str(tamaño_ready[q])"+"*str(ocupado_ready[q])"+"*str(libre_ready[q])"+"*list_name_user[q]+"*list_name_pass[q]+"*9999.0")
    if (list_max_delta[q]>0):
        cliente_ready_used.append(cliente_ready[q])
        descripcion_ready_used.append(descripcion_ready[q])
        tablespace_ready_used.append(tablespace_ready[q])
        time_for_fullspace=float(libre_ready[q]/list_max_delta[q])
        list_time_for_fullspace.append(time_for_fullspace)
        list_time_for_fullspace_complete.append(cliente_ready[q]+"*tipo_db_ready[q]+"*descripcion_ready[q]+"*list_name_alias[q]+"*list_name_sid[q]+"*tablespace_ready[q]+"
list_name_hostname[q]+"*list_name_ip[q]+"*str(tamaño_ready[q])"+"*str(ocupado_ready[q])"+"*str(libre_ready[q])"+"*list_name_user[q]+"*list_name_pass[q]+"*str(time_for_fullspace))
    if (list_max_delta[q]<0):
        cliente_ready_used_negativo.append(cliente_ready[q])
        descripcion_ready_used_negativo.append(descripcion_ready[q])
        tablespace_ready_used_negativo.append(tablespace_ready[q])
        time_for_fullspace_negativo=float(libre_ready[q]/(list_max_delta[q]*-1))
        list_time_for_fullspace_negativo.append(time_for_fullspace)
        list_time_for_fullspace_complete.append(cliente_ready[q]+"*tipo_db_ready[q]+"*descripcion_ready[q]+"*list_name_alias[q]+"*list_name_sid[q]+"*tablespace_ready[q]+"
list_name_hostname[q]+"*list_name_ip[q]+"*str(tamaño_ready[q])"+"*str(ocupado_ready[q])"+"*str(libre_ready[q])"+"*list_name_user[q]+"*list_name_pass[q]+"*str(time_for_fullspace
negativo))

```

Figura 5. Función de Proyección para el tiempo de vida en cada de las DB

En la figura 5 se aprecia la función creada mediante Python, encargada de analizar el histórico de datos almacenados de los diferentes clientes, con el fin de estimar una proyección en tiempo de vida útil.

### Fase V:

El GlobalMonitoring es una Web-Page dinámica, diseñada internamente en IBM como una herramienta útil para los especialistas de las diferentes áreas de servicio. En dicha Web-Page se presenta de manera gráfica el monitoreo de todos los sistemas SAP en la cobertura de IBM, de tal manera que se pueda generar una óptima administración previniendo caídas o rupturas del servicio por algún error generado por factores específicos incluidos ya en el monitoreo de GlobalMonitoring.

### Global Monitoring



Figura 6. Web-Page de monitoreo interno en el área de SAP

## Sección Monitoreo Espacios

DB SI	DB Type	SID	System Environment	TableSpace	INFO - Consolidated [ 2018-06-13 04:30:05 ]		
					Used MB	Free MB	Days
ORA	PRD	ERP PRD PRODUCTIVO	SYSAUX	2021.1200	1458.4320	112	
DB2	SMP	SMP Produccion	/db2/SMP/sapdata1	35455.2140	37961.8840	1508	
DB2	SMP	SMP Produccion	/db2/SMP/sapdata2	35455.2500	37961.8470	1508	

Figura 7. Sección interna del monitoreo de espacios dentro del Global-Monitoring

En la figura 7 se aprecia la interfaz gráfica que se tiene para la exposición de datos recolectados de cada una de las diferentes bases de datos administradas, dependiendo del cliente.

### 4.3 Automatización Update-Kernel

Update-Kernel es un proyecto que nace junto con la necesidad diaria de los administradores de sistemas SAP en minimizar al máximo el tiempo que se requiere en ciertas labores operacionales que implican indisponibilidad del sistema, entre esas tareas esta la actualización de binarios Kernel, cuyo promedio de tiempo en implementación es de 60Minutos. Con este autómata se garantizará una reducción significativa en el promedio de tiempo implicado para la implementación de dicha actualización, así mismo generará fiabilidad por parte del cliente, dado que el tiempo en indisponibilidad del sistema para este tipo de cambios será mínimo.

Debido a que la actualización Kernel se hace a nivel de Sistema Operativo, era necesario usar un lenguaje o interfaz de programación que permitiera de manera más dinámica y efectiva ejecutar comandos de sistema operativo, es decir, lenguaje de consola. Dicha interfaz o Shell escogida fue "Bash", cuya función consiste en interpretar órdenes y un lenguaje de consola.

El código de desarrollo se realizó de manera progresiva y paralela al nivel de aprendizaje que se iba teniendo de Bash, generando así un resultado efectivo para la actualización de Kernel en bases de datos Oracle ,Db2 y Sybase. Recordando que el Kernel es el puente de enlace entre las aplicaciones y las bases de datos, por ello es de vital importancia saber qué tipo de datos tiene el sistema, pues dependiendo de ello se procede a ejecutar una serie distinta de comandos.

El script necesita como entrada 13 parámetros:

- I. Tipo de base de datos
- II. Usuario de instancia central
- III. Clave de usuario de instancia central
- IV. IP de instancia central
- V. Usuario de base de datos
- VI. Clave de usuario de base de datos
- VII. IP de base de datos
- VIII. SID
- IX. Numero de instancia
- X. Nombre completo de la instancia
- XI. Ruta donde están alojados los nuevos binarios dentro de SAP-RHEL
- XII. Usuario con permisos de superadministrador
- XIII. Clave de usuario superadministrador

El script no se permitirá ejecutar hasta no tener los 13 parámetros incluidos.

Ya ejecutándose la lógica del script es la siguiente:

- I. Baja el listener y la Base de datos
- II. Limpia semáforos y procesos activos de la base de datos
- III. Baja SAP
- IV. Limpia semáforos y procesos activos de SAP
- V. Busca y extrae la ruta de alojamiento del Kernel a partir del log file "sapcpe"
- VI. Copia medios del nuevo Kernel en la ruta extraída
- VII. Sube Listener y Base de datos
- VIII. Sube SAP
- IX. Corroboración de la óptima actualización
- X. El script recorre a nivel de comandos de consola cada uno de los pasos lógicos mencionados anteriormente y realiza de manera óptima la transferencia y la actualización de medios Kernel.

Catalogación de errores:

Dado que el procedimiento de actualización Kernel es delicado a nivel de sistema operativo, se hace de vital importancia realizar un manejo de errores posibles a medida que se realiza la ejecución lógica de comandos en consola. Dicho esto, se catalogaron los siguientes errores:

- I. Error en credenciales de conexión: En caso de obtener este error, por favor revise y valide las credenciales suministradas para la conexión al servidor de instancia centra y servidor de base de datos.
- II. No hay espacio disponible para la transferencia de binarios: En caso de obtener este error, por favor revise y agregue más espacio al “filesystem” donde se encuentran alojados los binarios Kernel del servidor destino para poder realizar la transferencia satisfactoria de los nuevos binarios.
- III. Error bajando instancia SAP: En caso de obtener este error, por favor revise la óptima ejecución del script “stopsap”.
- IV. Error limpieza de semáforos: En caso de obtener este error, por favor revise la óptima ejecución del comando “cleanipc” para dicha instancia.
- V. Error bajando base de datos: En caso de obtener este error y dependiendo del tipo de base de datos, por favor revise y valide la bajada de esta manualmente.
- VI. Error en transferencia de nuevos binarios: En caso de obtener este error, ejecute el script nuevamente y si el error persiste por favor contacte a personal de soporte técnico de I+D.
- VII. Error subiendo base de datos: En caso de obtener este error y dependiendo del tipo de base de datos, por favor revise y valide la subida de esta manualmente.
- VIII. Error subiendo instancia SAP: En caso de obtener este error, por favor revise la óptima ejecución del script “startsap”.
- IX. Error ejecución “sapcpe”: En caso de obtener este error, por favor revise la ejecución del “sapcpe”, dado que no fue detectada y por tanto no se realizó la copia automática de los binarios en los aplicativos anidados al servidor.

#### **4.4 Automatización Monitoreo Replicas:**

El script del monitoreo de réplicas nace junto con la necesidad diaria de los administradores de sistemas SAP de tener una evaluación frecuente acerca del funcionamiento óptimo en el proceso de sincronización de un sistema con replica. El script fue elaborado en lenguaje de consola haciendo uso del intérprete de comandos BASH.

Diagrama de Flujo:

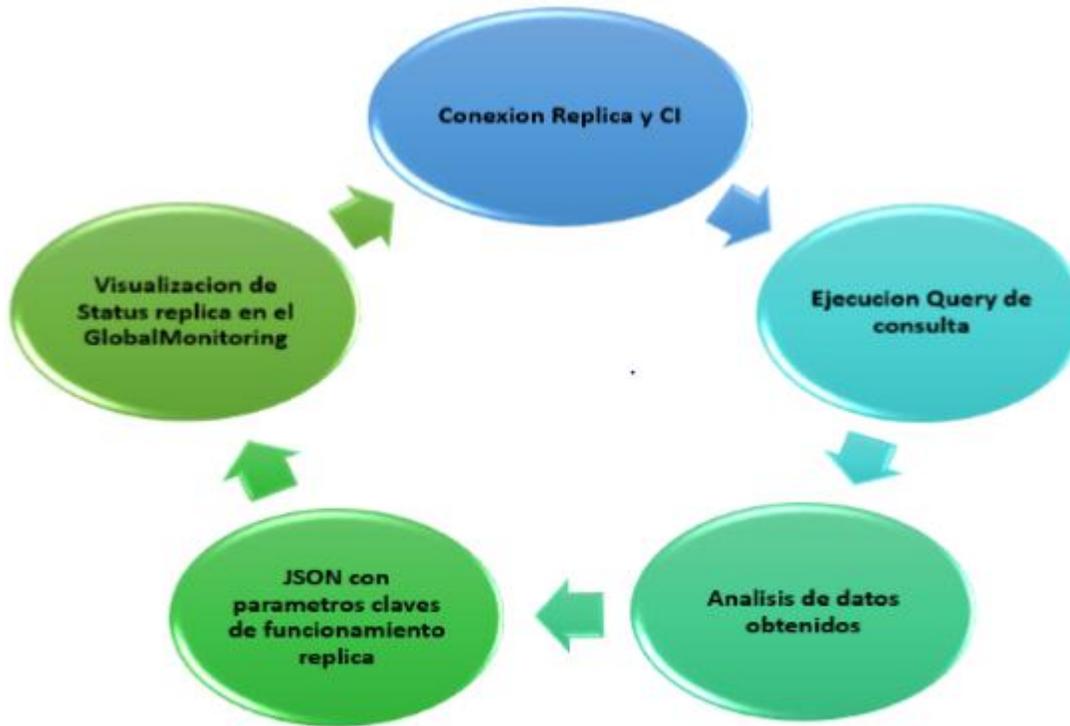


Figura 8. Diagrama de flujo de la lógica implementada en el Monitoreo de réplicas

En la figura 8 se aprecia la secuencia lógica utilizada para la implementación del monitoreo en réplicas de bases de datos administradas por IBM.

Catalogación de errores:

Se realizó un análisis de los respectivos errores que pudieran suceder en la extracción de la información y en el análisis de esta. Dependiendo del tipo de error, el autómata verifica si continúa o no con la ejecución.

El tipo de error se catalogó para bases de datos DB2 y para bases de datos ORACLE.

- Errores en DB2:

- I. Error de conexión en el servidor de instancia central: En caso de que suceda, por favor proceda a revisar y validar las credenciales de conexión almacenadas.
- II. Error cliente DB2 no encontrado: En caso de que suceda, por favor proceda a validar las variables de ambiente del cliente DB2 o el correcto funcionamiento de el cliente db2.
- III. Error de conexión a base de datos: En caso de que suceda, por favor proceda a validar el si el SID suministrado corresponde al nombre de la base de datos.
- IV. Replica no Sincronizada: En caso de que suceda, por favor proceda a validar la sincronización de logs del servidor de instancia central con la réplica.

- Errores en ORACLE:

- I. Error de conexión en el servidor de instancia central: En caso de que suceda, por favor proceda a revisar y validar las credenciales de conexión almacenadas
- II. Error cliente SQLPLUS: En caso de que suceda, por favor proceda a validar las variables de ambiente del cliente de SQLPLUS o el correcto funcionamiento de el cliente sqlplus.
- III. Error ejecución Túnel hacia la réplica: En caso de que suceda, por favor revisar el puerto o la ip de la réplica a la cual se desea generar un túnel.
- IV. Error de conexión en el servidor de réplica o conexión por túnel rechazada: En caso de que suceda, por favor revisar y validar las credenciales de conexión almacenadas para la réplica o cambiar el puerto local que se está utilizando para generar el túnel.
- V. Error cliente SQLPLUS en replica: En caso de que suceda, por favor proceda a validar las variables de ambiente del cliente de SQLPLUS o el correcto funcionamiento de el cliente en el servidor de réplica.
- VI. Replica sincronizada pero no está siendo aplicados los logs: En caso de que suceda, por favor proceda a validar porque la réplica no está aplicando los logs sincronizados.
- VII. Replica no sincronizada: En caso de que suceda, por favor proceda a validar la sincronización de logs del servidor de instancia central con la réplica.

## 5. CONCLUSIONES

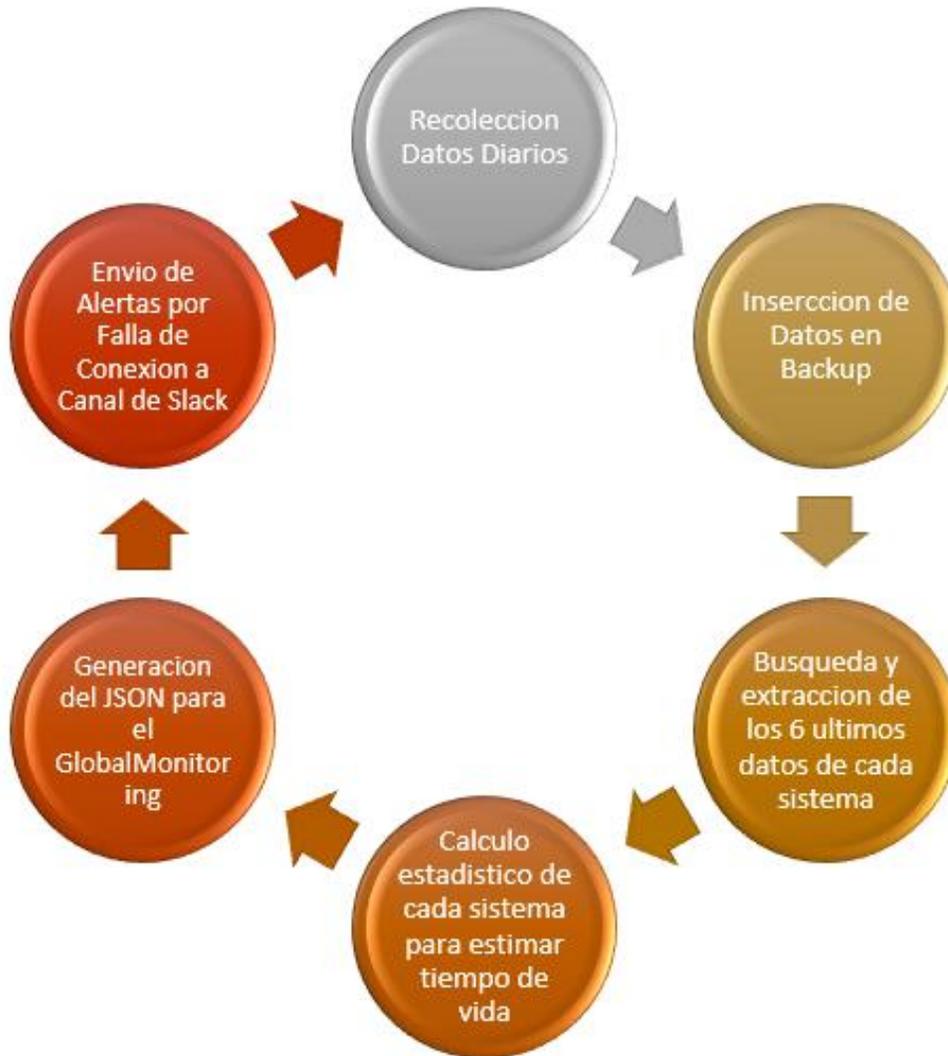
- Administracion SAP Basis a clientes específicos de IBM con sistemas SAP instalados en sus servidores.
- Mediante el monitoreo de base de datos se logró una reducción significativa de incidentes relacionados con caída de bases de datos por llenado de espacio, esto de manera paralela no solo ayuda al especialista para dar un óptimo cumplimiento a sus labores administrativas si no que también ayuda al cliente a no tener pérdidas por indisponibilidad del sistema.
- Diseño de un óptimo self-monitoring en servidores replicas, donde el especialista sabe cada 5 minutos en qué estado se encuentra su réplica, dándole así el tiempo suficiente de accionar y corregir los respectivos errores.
- Diseño de un automata capaz de realizar la actualización de binarios Kernel, manejando los errores posibles que se pueden presentar y reduciendo el tiempo de indisponibilidad de un sistema de 1 hora a 8 minutos que demora la ejecución, generando mucho impacto a nivel operativo y a nivel económico de parte del cliente dando así una posible oportunidad de ofrecer el automata como una herramienta de venta hacia los clientes.
- Se diseñaron una serie de herramientas preventivas que ayude a los especialistas de sistemas SAP a cumplir plenamente con sus labores administrativas, minimizando así la alta carga operativa del día a día.

# **ANEXOS**

Anexo A.

Manual Automatización DB-Monitoring

## Diagrama de Flujo



### Paso – Paso del Desarrollo:

Este desarrollo viene incorporado con toda la funcionalidad en un solo script, por ello se explicará la secuencia lógica que fue utilizada para el análisis de datos y estimación de vida en días. Los pasos son los siguientes:

- Importación de Librerías necesarias para la conexión a base de datos Oracle, DB2 y mariaDB, generación de JSON y conexión a Slack.
- Se creó una base de datos MariaDB alojada en SAP-RHEL, la cual es la encargada de almacenar toda la información extraída diariamente del monitoreo, en ella hay una tabla llamada “espacios\_ini”, la cual es la tabla referencia que tiene los datos de todos los sistemas que se desean monitorear. Para ingresar sistemas nuevos a dicha tabla, se tiene una web-page que se comunica diariamente con la base de datos para realizar la respectiva actualización de los nuevos sistemas ingresados.
- Tomando la tabla “espacios\_ini” como punto de partida, el primer paso es recorrer todos los sistemas que están allí alojados, conectarse remotamente a la base de datos de cada uno de ellos y extraer la información necesaria para el análisis de monitoreo. La información necesaria es:
  - Espacio libre de cada “tablespace”
  - Espacio Usado de cada “tablespace”
  - SID del sistema
  - Alias del sistema
  - Hostname del sistema
  - Puerto de enlace por el cual escucha la base de datos
  - Nombre Cliente
  - Nombre del “tablespace”

Este es el ciclo principal de la lectura de base de datos, donde se recorre toda la tabla “espacios\_ini” y se conecta a cada uno de los sistemas encontrados, su método de conexión cambia dependiendo si es Oracle o DB2, por ello se filtra de acuerdo con el tipo de base de datos, para que así se utilice la respectiva función. En cada una de las funciones se extrae la información necesaria del sistema que se esté leyendo en esos momentos e inmediatamente se realiza la actualización de la información en la tabla “espacios\_stats”, que es la encargada de almacenar toda la información capturada diariamente.

Teniendo ya almacenados los datos del día presente, se procede a la extracción de los últimos 5 datos almacenados de cada sistema para así poder estimar una tendencia de uso. La extracción de los 5 últimos datos toma los mismos sistemas leídos el día de hoy como referencia para extraer esos 5 últimos datos de cada uno almacenados en la tabla “espacios\_stats”.

Para garantizar que los datos son fiables, y que pertenecen a cada uno de sus respectivos parámetros, se realiza un procedimiento de filtrado de 4 capas, donde la primera capa se asegura de que este en el mismo cliente, la segunda capa se asegura que este sobre la misma descripción del sistema, la tercera capa asegura que este sobre el mismo tablespace del cual está queriendo saber la información y la cuarta asegura que se extraiga la información de la fecha que se quiere.

- Ya con los últimos 5 datos de cada sistema extraídos, se procede a realizar el respectivo calculo estadístico para estimar una tendencia de uso y así mismo un tiempo de vida útil.

Primeramente y basado en los últimos 5 datos recolectados del sistema, se estima el máximo delta de uso que tuvo el sistema y se cataloga en 3 niveles: Si  $\text{delta} = 0$ ,  $\text{delta} > 0$  o  $\text{delta} < 0$ .

- Si  $\text{delta} = 0$  quiere decir que ese tablespace no ha tenido ningún uso a nivel de espacios los últimos 5 días.
- Si  $\text{delta} > 0$  quiere decir que ese tablespace tuvo uso y disminuyo su espacio libre lo últimos 5 días
- Si  $\text{delta} < 0$  quiere decir que ese tablespace tuvo liberación de espacios y aumento su espacio libre en los últimos 5 días.

Con el máximo delta de uso de cada sistema catalogado, se procede a realizar el cálculo para estimar cuanto tiempo le queda de vida antes de que ese tablespace llegue al 100% de uso, para este cálculo se realiza una sola operación:

$$\text{LIFE} = \text{FreeSpaceCurrent} / \text{MaxDelta}$$

Se divide el espacio libre actual entre el máximo delta de uso calculado anteriormente, dando así una constante que equivale al número de veces que se debe repetir ese delta máximo para que ocupe el espacio libre que queda, y teniendo en cuenta que todos los datos son diarios, pues dicha constante es válida para la misma lógica, es decir, la cantidad de veces que se debe repetir delta es igual a la misma cantidad de días que le quedan al sistema para que se llene ese espacio libre.

- Ya teniendo los tiempos de vida calculados, se procede a realizar la actualización de el dato en su respectivo cliente, descripción, tablespace y fecha en la tabla de “espacios\_stats”, teniendo así un histórico de tiempos de vida de los diferentes sistemas, pudiendo así a futuro generar datos estadísticos de mayor rigor.

- Se genera los dos JSON necesarios para el Global-Monitoring, uno que representa los tiempos de vida de todos los sistemas incluyendo sus datos, y el otro representa los tiempos de vida de los sistemas que debieran ser alertados, para ello se dejó un margen límite de 5 días, es decir, si algún tablespace tiene menos de 5 días de vida, se procede a alertarse para generar la respectiva revisión o mantenimiento.
- Por último, se envía las notificaciones de los sistemas que no pudiera ser leídos el día de hoy a un canal de Slack, donde estarán los respectivos administradores del monitoreo que tomaran acción y corregirán el error que esté sucediendo con la lectura de determinado sistema.

## **Ubicación, Ejecución y Contrab del Script de desarrollo:**

El script de desarrollo se encuentra alojado en el “home” del usuario “coecogni” en el servidor SAP-RHEL. La ruta es la siguiente:

PATH: /home/coecogni/espacios/run\_espacios.py

El script se ejecuta de forma automáticamente todos los días a las 4:30 por medio del crontab que ejecuta el script en segundo plano. Dado que el crontab se ejecuta con las librerías de sistema y no las del usuario directamente involucrado en la creación de el crontab, se debe exportar y crear las respectivas variables de entorno necesarias para la óptima ejecución del script. Debido a ello se creó un Shell que exporta las variables de entorno y se direcciona a la ruta donde se encuentra alojado el script de desarrollo para ejecutarlo.

## **Si hay Fallos ¿qué hacer?**

En caso de fallos se debe direccionar al archivo de log “espacios\_error.log” que se guarda en la misma ruta donde está alojado el script de desarrollo, con el fin de detallar que impresión de error se pudo a ver generado en la ejecución del script. Basado en ello realizar las respectivas correcciones.

Ruta: /home/coecogni/espacios/espacios\_error.log

## Descripción tablas usadas base de datos MariaDB:

En el script de desarrollo se hacen uso de las siguientes tablas:

- **espacios\_ini**: Es la tabla encargada de almacenar toda la información de conexión de todos los sistemas que hasta el momento están siendo monitoreados en espacios.
- **espacios\_stats**: Es la tabla encargada de almacenar todo el histórico de información diaria recogida de cada uno de los sistemas monitoreados.  
Descripción:

## Parámetros de conexión Módulos Oracle y DB2:

DB2:

La conexión a DB2 por medio del módulo “**ibm\_db**” en Python es satisfactoria siempre y cuando se garantice la veracidad de los siguientes parámetros:

- SID -----> Representa el nombre de la base de datos
- HOSTNAME -----> Representa el nombre del servidor donde está alojada la base de datos
- PROTOCOLO -----> Representa el protocolo de comunicación que se desea usar
- USER ID -----> Usuario con los permisos para acceder a la base de datos
- PASSWORD USER ID -----> Contraseña del usuario con permisos de ingreso
- PUERTO DE ENLACE -----> Numero de puerto por el cual escucha la base de datos.

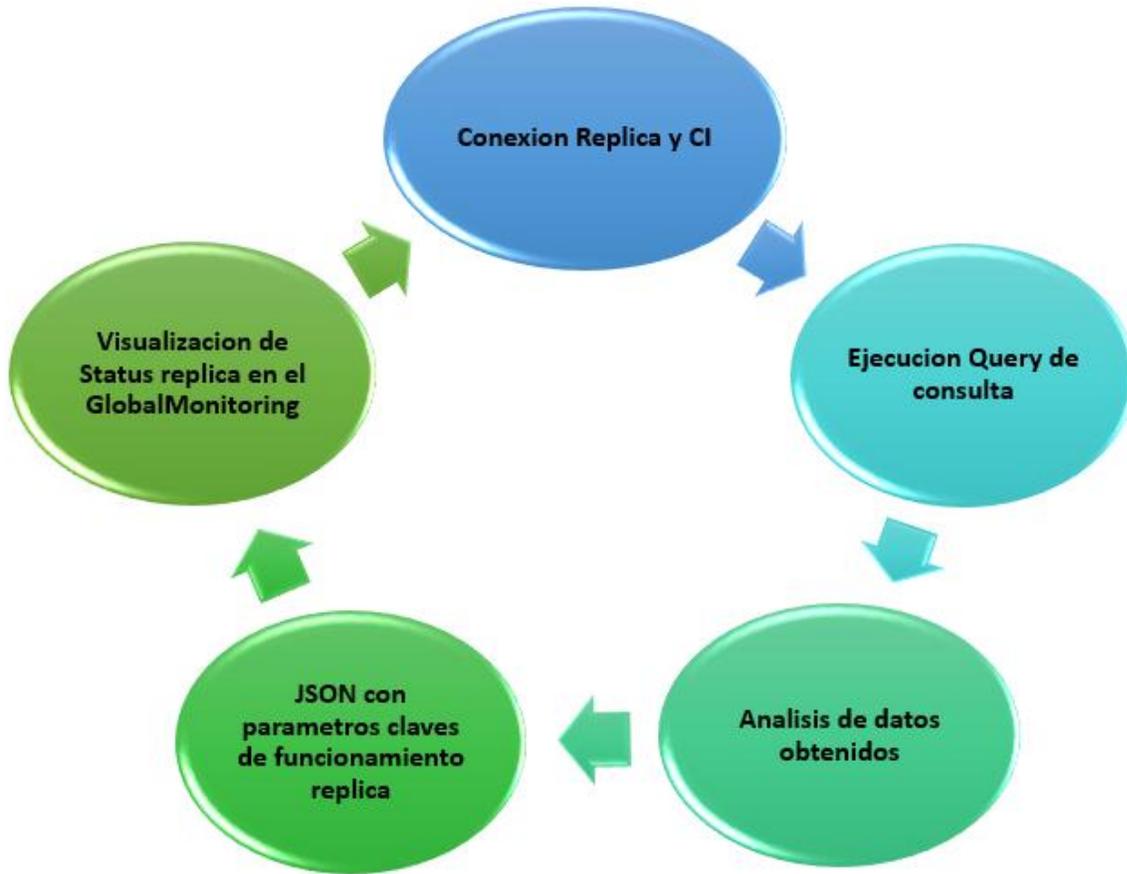
ORACLE:

La conexión a ORACLE por medio del módulo “**cx\_Oracle**” en Python es satisfactoria siempre y cuando se garantice la veracidad de los siguientes parámetros:

- HOST -----> Representa el nombre del servidor donde está alojada la base de datos
- SID -----> Representa el nombre de la base de datos
- PORT -----> Numero de puerto por el cual escucha la base de datos.
- USER -----> Usuario con los permisos para acceder a la base de datos
- PASSWORD -----> Contraseña del usuario con permisos de ingreso

Anexo B.  
Manual Automatización Monitoreo de  
Replicas

## Diagrama de Flujo



### Paso – Paso Funcionamiento Script:

A continuación, se explicará la secuencia lógica y la respectiva estructura del Script:

- El script tiene un único parámetro de entrada y es el “ID” con el cual se encuentra grabado en la tabla “**replicas\_init**” con base de datos MariaDB local del sistema que se desee monitorear. A partir de ese “ID” se extrae toda la información requerida por parte del desarrollo para su respectivo análisis.
- Teniendo ya la información necesaria, se procede a evaluar qué tipo de base de datos tiene determinado sistema SAP y dependiendo de ello se procede a ejecutar los respectivos Querys de consulta.

La salida de las consultas SQL realizadas muestran el LOG actual en el cual se encuentra la instancia central y la réplica. El criterio de alerta que se maneja es que la diferencia entre ambos no sea mayor a "X" (rango → parámetro incluido por parte del especialista en la interfaz gráfica) LOGS.

Después de ser capturada la información necesaria, se procede a realizar la respectiva evaluación de funcionamiento e inmediatamente se genera el JSON con los parámetros claves.

Parámetros:

- Tipo de base de datos
- Cliente
- Descripción del sistema
- SID
- Estado Actual de funcionamiento
- Resultado del estado actual de funcionamiento
- Hora de verificación

### **Catálogo de Errores:**

Se realizó un análisis de los respectivos errores que pudieran suceder en la extracción de la información y en el análisis de esta. Dependiendo del tipo de error, el autómata verifica si continua o no con la ejecución.

El tipo de error se catalogó para bases de datos DB2 y para bases de datos ORACLE.

Errores en DB2:

- Error de conexión en el servidor de instancia central:

En caso de que suceda, por favor proceda a revisar y validar las credenciales de conexión almacenadas.

- Error cliente DB2 no encontrado:

En caso de que suceda, por favor proceda a validar las variables de ambiente del cliente DB2 o el correcto funcionamiento de determinado cliente.

- Error de conexión a base de datos:

En caso de que suceda, por favor proceda a validar el si el SID suministrado corresponde al nombre de la base de datos.

- Replica no Sincronizada:

En caso de que suceda, por favor proceda a validar la sincronización de logs del servidor de instancia central con la réplica.

#### Errores en ORACLE:

- Error de conexión en el servidor de instancia central:

En caso de que suceda, por favor proceda a revisar y validar las credenciales de conexión almacenadas

- Error cliente SQLPLUS:

En caso de que suceda, por favor proceda a validar las variables de ambiente del cliente de SQLPLUS o el correcto funcionamiento de el cliente.

- Error ejecución Túnel hacia la replica:

En caso de que suceda, por favor revisar el puerto o la ip de la réplica a la cual se desea generar un túnel.

- Error de conexión en el servidor de réplica o conexión por túnel rechazada:

En caso de que suceda, por favor revisar y validar las credenciales de conexión almacenadas para la réplica o cambiar el puerto local que se está utilizando para generar el túnel.

- Error cliente SQLPLUS en replica:

En caso de que suceda, por favor proceda a validar las variables de ambiente del cliente de SQLPLUS o el correcto funcionamiento de el cliente en el servidor de réplica.

- Replica sincronizada pero no está siendo aplicados los logs:

En caso de que suceda, por favor proceda a validar porque la réplica no está aplicando los logs sincronizados.

- Replica no sincronizada:

En caso de que suceda, por favor proceda a validar la sincronización de logs del servidor de instancia central con la réplica.

## Cómo Usar?

El script se ejecuta de una manera muy sencilla y practica a nivel de sistema operativo y dentro del servidor SAP-RHEL:

Comando de ejecución:

**bash statusReplicas\_ORA\_DB2.sh ID**

**bash** --> Interprete de comandos de consola

**statusReplicas\_ORA\_DB2.sh** ----> Shell con toda la lógica de comandos

**ID** --> Número de identificación con el cual este guardado el sistema en la base de datos de espacios en MariaDB.

### Criterio de Alerta:

- El criterio de alerta que el autómata usa para confirmar si la réplica esta sincronizada o no, es basado en un rango requerido como parámetro que escoge el especialista:
- La diferencia de LOG actual en el servidor de réplica y el LOG actual en el servidor de instancia central no sea mayor a X LOGS en bases de datos ORACLE. En bases de datos DB2 es mucho más fácil, dado que el Query que se ejecuta, le dice automáticamente al autómata si esa replica tiene problemas o no.

### Pre-requisitos de ejecución:

- Es indispensable que el sistema sea agregado por medio de la interfaz gráfica de monitoreo TeamSAP, dado que el autómata se basa en dicha base de datos.
- Después de ser agregado el sistema a la interfaz gráfica, es necesario generar la relación de confianza entre el usuario de aplicación “**coecogni**” y el servidor de instancia central destino.
- El autómata da por hecho que el usuario de servidor de instancia central tiene relación de confianza con el servidor de réplica.

Anexo C.  
Manual Automatización de Actualización  
Kernel



## **Paso – Paso del Desarrollo:**

Este desarrollo viene incorporado con toda la funcionalidad en un solo script, por ello se explicará la secuencia lógica que fue utilizada para la implementación y ejecución de comandos. Los pasos son los siguientes:

- Recepción de parámetros requeridos para la ejecución:
- Tipo de base de datos
- Usuario de instancia central
- Clave de usuario de instancia central
- IP de instancia central
- Usuario de base de datos
- Clave de usuario de base de datos
- IP de base de datos
- SID
- Numero de instancia
- Nombre completo de la instancia
- Ruta donde están alojados los nuevos binarios dentro de SAP-RHEL
- Usuario con permisos de super-administrador
- Clave de usuario super-administrador

Pasos ejecutados después de proporcionar los datos pedidos:

- Basado en los parámetros suministrados se realiza un test de conexión tanto en la base de datos como en la instancia central, validando así las credenciales correctas para poder seguir con la ejecución de comandos.
- Ya validada la información de credenciales, se procede a realizar la búsqueda de la ruta donde se encuentran alojados los binarios del Kernel por medio del log “sapcpe”. En caso de que el log no sea encontrado, el autómata le pedirá que ingrese la ruta manualmente.
- Teniendo ya la ruta de Kernel, se procede a realizar la validación de disponibilidad en espacio de ese “filesystem” para la transferencia de los nuevos binarios hacia el servidor. En caso de que no tenga el espacio disponible para la transferencia de binarios, el autómata se detendrá y pedirá que asignen para seguir con la ejecución.

- Comprobado ya la disponibilidad de espacio, se procede a bajar la instancia SAP y a realizar la limpieza de semáforos. En caso que el autómata no pueda dar de baja a la instancia SAP, se detendrá y pedirá que revise el comando.
- Seguidamente se procede a bajar la base de datos dependiendo de su tipo. En caso que el autómata no pueda bajar la base de datos, se detendrá y pedirá que revise el comando.
- Teniendo ya la instancia SAP y la base de datos detenida, se procede a realizar la transferencia de los nuevos binarios Kernel hacia el servidor destino de instancia central e inmediatamente se realiza el “backup” de los binarios Kernel actualmente implementados.
- Se ingresa a la ruta del nuevo directorio de binarios Kernel alojados y se procede a realizar la ejecución de los scripts con el usuario superadministrador en los cuales se setea los permisos de los archivos que contiene.
- Se procede a subir la base de datos dependiendo de su tipo.
- Siguiendo el orden jerárquico, se procede a subir la instancia SAP.

Finalmente, teniendo ya la instancia SAP y la base de datos en ejecución, se procede a validar el funcionamiento de los nuevos binarios kernel alojados en el servidor. Para ello, se realiza dos tipos de evaluaciones:

- Se captura la versión del Kernel antes de realizar la transferencia de nuevos binarios y se vuelve a capturar la versión después de realizada la transferencia, teniendo así una comparación, que en caso de ser iguales se denotara que ocurrió algún error en la actualización.
- Se captura la hora de ejecución del SAPCPE antes de realizar la transferencia de nuevos binarios y se vuelve a capturar después de realizada la transferencia, que, en caso de ser iguales, se denotara que el “sapcpe” no se ejecutó y por tanto no realizó su trabajo de copia de binarios en los aplicativos.

## **Catalogación de Errores:**

Dado que el procedimiento de actualización Kernel es delicado a nivel de sistema operativo, se hace de vital importancia realizar un manejo de errores posibles a medida que se realiza la ejecución lógica de comandos en consola. Dicho esto, se catalogaron los siguientes errores:

### **1. Error en credenciales de conexión:**

En caso de obtener este error, por favor revise y valide las credenciales suministradas para la conexión al servidor de instancia centra y servidor de base de datos.

### **2. No hay espacio disponible para la transferencia de binarios:**

En caso de obtener este error, por favor revise y agregue más espacio al "filesystem" donde se encuentran alojados los binarios Kernel del servidor destino para poder realizar la transferencia satisfactoria de los nuevos binarios.

### **3. Error bajando instancia SAP:**

En caso de obtener este error, por favor revise la óptima ejecución del script "stopsap".

### **4. Error limpieza de semáforos:**

En caso de obtener este error, por favor revise la óptima ejecución del comando "cleanipc" para dicha instancia.

### **5. Error bajando base de datos:**

En caso de obtener este error y dependiendo del tipo de base de datos, por favor revise y valide la bajada de esta manualmente.

### **6. Error en transferencia de nuevos binarios:**

En caso de obtener este error, ejecute el script nuevamente y si el error persiste por favor contacte a personal de soporte técnico de I+D.

### **7. Error subiendo base de datos:**

En caso de obtener este error y dependiendo del tipo de base de datos, por favor revise y valide la subida de esta manualmente.

### **8. Error subiendo instancia SAP:**

En caso de obtener este error, por favor revise la óptima ejecución del script "startsap".