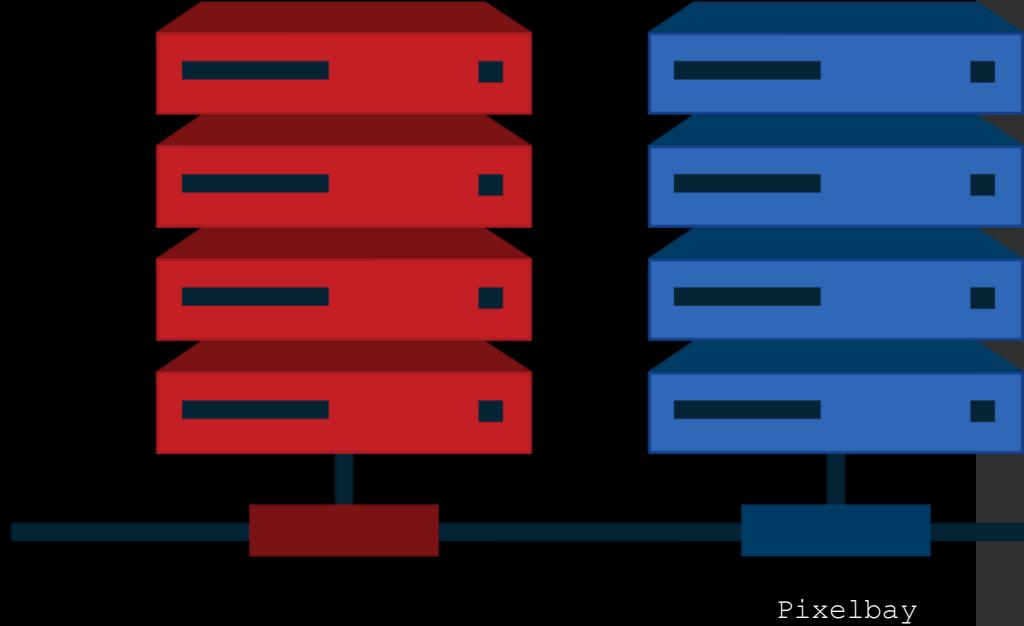


Taller Práctico: Guía de supervivencia para un Data Scientist

Parte I Oracle + Python



...

LUKE: Is **Perl** better than **Python**?

YODA: No... no... no. Quicker, easier, more seductive.

LUKE: But how will I know why **Python** is better than **Perl**?

YODA: You will know. When your code you try to **read** six months from now.

python.org/doc/humor

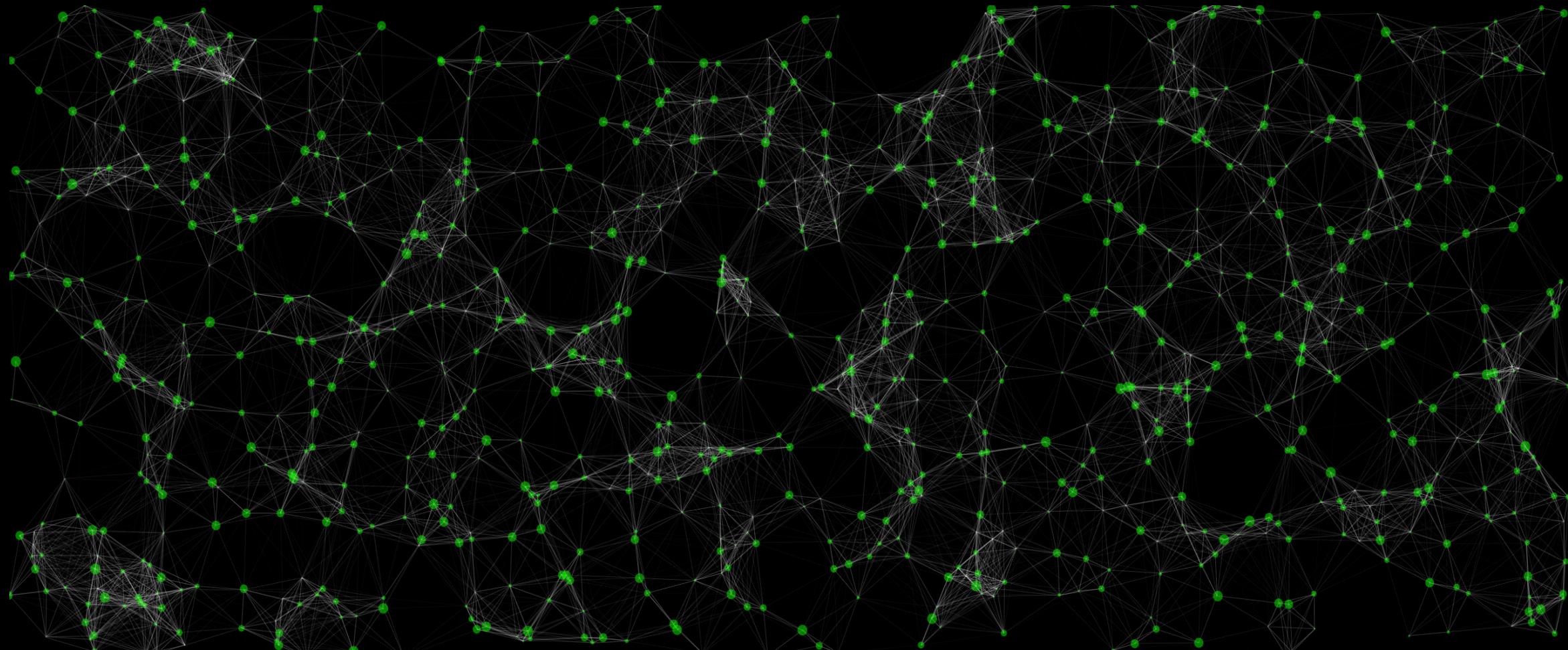
#DondeYComoNosConectamos

Puede que estéis...

- Utilizando varios routers de salto
 - Detrás de un o de varios firewalls
 - Interconectados por switches o redes inalambricas
 - Conectados por Citrix de la empresa
 - A traves de un servidor de APPs
 - Con un navegador
 - Con un cliente pesado
 - Utilizando protocolo SSH/HTTP
 - Con Proxy o sin Proxy server
 - ...



#EI_Elemento_U_LaRED



Generado con **particles.js**
(github.com/VincentGarreau/particles.js)

#Material_D_Trabajo

Hardware & Base de Datos

- Oracle 19c on Cloud
- 1 Autonomous Database EXADATA
- 1 OPC Core (2 Threads)
- 20GB / DB Storage
- 3GB SGA
- Local Core i7 32GB + SSD 512GB



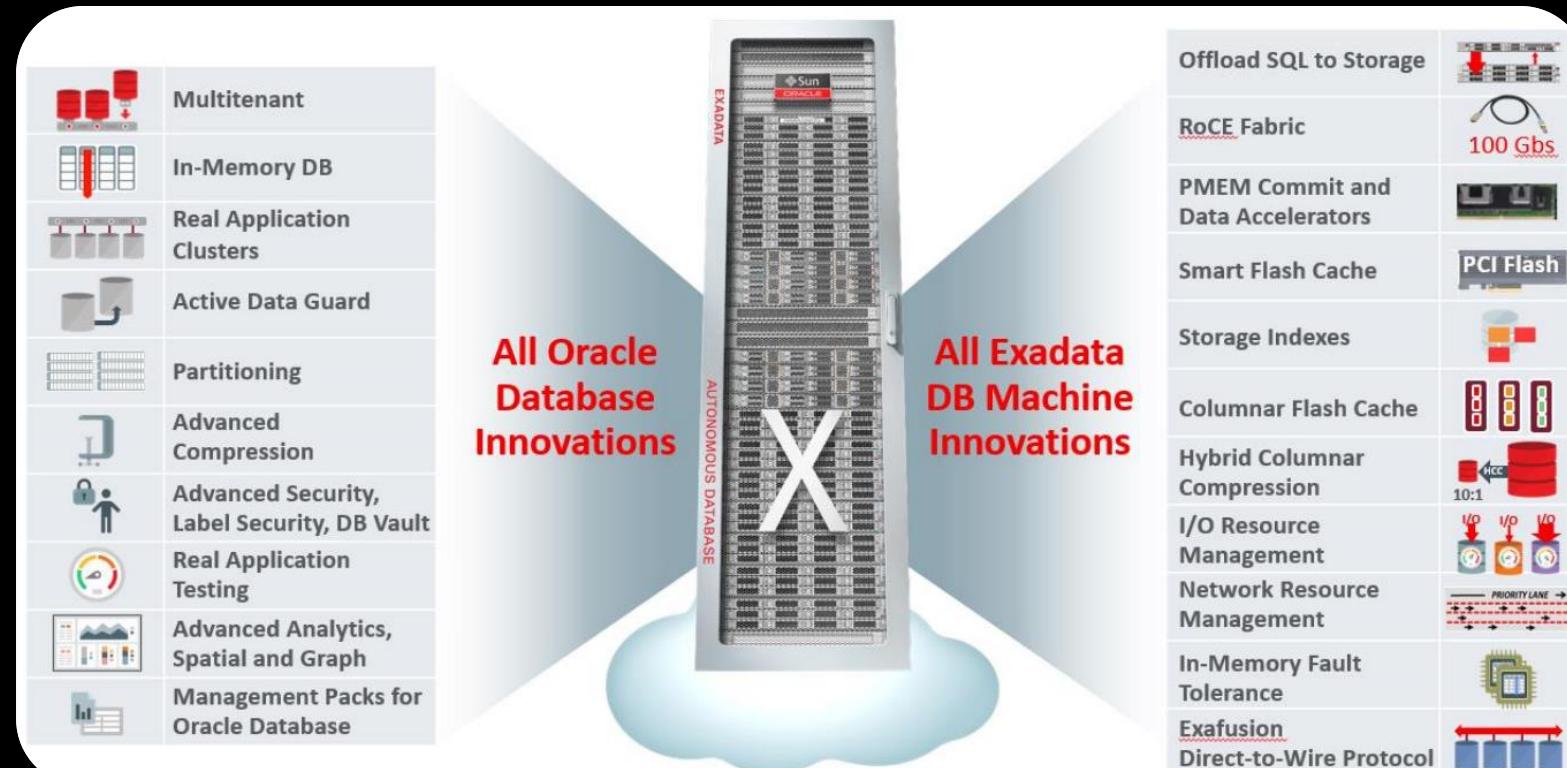
Software & Programación

- Python 3.8.5 x64
- Visual Studio Code + Oracle Developer Tools VS Code 19.3.2
- Oracle SQL Developer 20.2.0.175
- Oracle Instant Client 19.8 x64



#Snippet_On_EXADATA

- **Combinación de Hardware & Software (algunos son exclusivos en Exadata)**
- **Separación en dos capas de hardware**
- **Red de Interconnect con INFINIBAND**
- **Muy alta compresión de datos (HCC Hybrid Columnar Compression)**
- **Smart FLASH Cache (PCI)**
- **Sistema Operativo Solaris/Linux**



#Snippet_On_PY_cx_Oracle



- Versión actual **8.0**
- Módulo para Python 3.5 a 3.8 en la versión actual
- Sigue la especificación **DB API 2.0** (con algunas excepciones)
- Requiere instalación adicional de **Oracle Instant Client** 11.2 o superior
- Ejecución SQL y **PL/SQL** soportadas
- Control de inicio y parada de la Base de Datos
- **Exclusiones del estandar DB API:**
 - El tipo de datos **Time** no esta soportado en Oracle y no se ha implementado
 - El metodo `cursor.nextset()`, se utilizaría `cursor.getimplicitresults()` en su lugar

#Instalar_cx_Oracle_On_PY

```
$ python3 -m pip install cx_Oracle --upgrade
```



- Si no tenéis los binarios de python en el PATH, invocar con el path completo, por ejemplo **C:\Python\Python38\python.exe**
- Para instalar localmente a vuestro usuario (site-packages) utilizar la opción **--user**
- Respecto a la opción **--upgrade**, recomiendo antes invocar pip con la opción **show** y probar primero sin **--upgrade** hasta estar seguros de los cambios.

```
PS C:\> C:\Python\Python38\python.exe -m pip show cx_Oracle
Name: cx-Oracle
Version: 8.0.1
Summary: Python interface to Oracle
Home-page: https://oracle.github.io/python-cx_Oracle
Author: Anthony Tuininga
Author-email: anthony.tuininga@gmail.com
License: BSD License
Location: c:\python\python38\lib\site-packages
Requires:
Required-by:
```

#Snippet_On_Oracle_Instant_Client



- Versión actual **19.8.0.0.0**
- Dos opciones de paquetes, **Basic** y Light
- Es la base APIs de Node.js, Python, PHP y otros lenguajes
- Soporte para MS-Win, Linux, Solaris, macOS, AIX y HP-UX
- Mismas bibliotecas que Oracle Database (uso intensivo)
- Base para la conexión de aplicaciones de escritorio como **TOAD** y **SQL Developer**

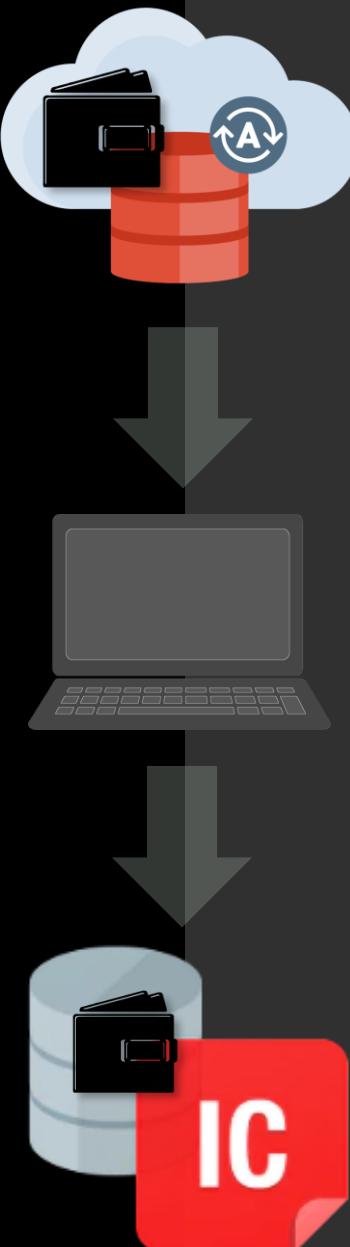
#Instalar_IC_On_PY



Descomprimir en la carpeta C:\instantclient_19_8

- La versión puede cambiar, así que **_19_8** podría verse modificada, por lo que se ha de adaptar a la nueva localización
- Inicializar las variables de entorno y/o métodos dependientes a la ubicación de la instalación de **IC**, **TNS_NAMES**, **PATH** y **cx_Oracle.init_oracle_client(path)**
- Para el desarrollo de los ejemplos que vienen a continuación, incluiremos en la variable de entorno **PATH** para Windows la ubicación de IC e inicializaremos los ejemplos con el método **cx_Oracle.init_oracle_client**

#Oracle_ADbase_IC_Y_Wallet



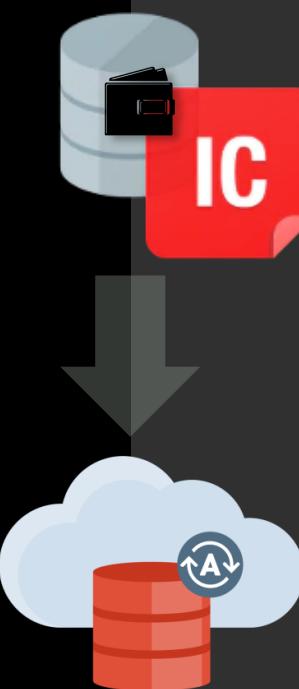
- Podemos bajarnos el respectivo **Wallet** de la ADB
- Lo descomprimiremos en la carpeta <ic_path/network/admin>
- Alternativamente, se puede instalar en otros directorios, para esto, se deberá inicializar la variable **TNS_ADMIN**
- Contiene el certificado **SSL** con una fecha de caducidad, normalmente 4Y
- Incluye el fichero **tnsnames.ora** y **sqlnet.ora**
- Importante para la conexión de las aplicaciones de escritorio como **TOAD** y **SQL Developer**, así como la utilización de las APIs de programación con Python, Perl, etc

#Iniciamos_CnxTestOADCLOUD



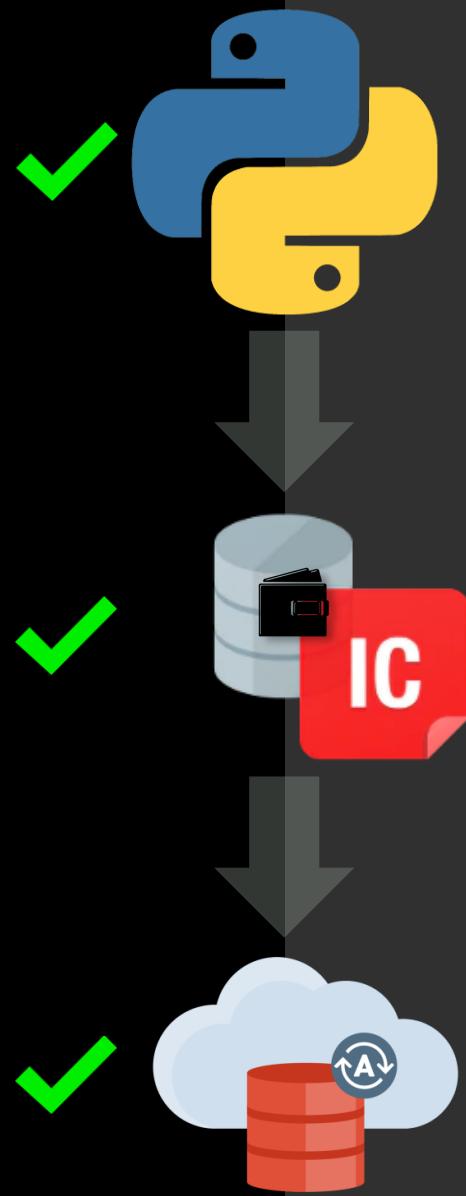
- Realiza un testing rápido, probamos los componentes básicos de la conexión para validar la misma

```
Instant Client Version (19, 8, 0, 0, 0) IC ✓
cx_Oracle Version           8.0.1 .py ✓
DBX Version ADB            19.5.0.0.0 Cloud ✓
DBX DSN Conexion          webinar01_high
DBX Encoding                UTF-8
DBX stmtcachesize          20
DBX TNS entrada            webinar01_high
DBX callTimeout              0
DBC arraysize                100
DBC bindarraysize             1
[INFO] No se han capturado excepciones durante las conexiones
[INFO] Cerrar cursos y conexion a Oracle Cloud ADB
[INFO] Cerrar cursos y conexion a Oracle Cloud ADB
[INFO] No se han capturado excepciones durante las conexiones
DBC arraysizes
T
```



#DEBUG_cx_Oracle_PS_Win10

```
PS \python> $env:DPI_DEBUG_LEVEL = 64
PS \python> C:\Python\Python38\python.exe
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import cx_Oracle
ODPI [24384] 2020-09-16 13:46:52.584: ODPI-C 4.0.2
ODPI [24384] 2020-09-16 13:46:52.586: debugging messages initialized at level
64
>>> ORA_INST_CLIENT_TNSC = r"webinar01_high"
>>> oc_user = 'admin'
>>> oc_pass = '*****'
>>> dbx = cx_Oracle.connect(oc_user, oc_pass, ORA_INST_CLIENT_TNSC)
ODPI [24384] 2020-09-16 13:47:57.280: Context Parameters:
ODPI [24384] 2020-09-16 13:47:57.288: Environment Variables:
ODPI [24384] 2020-09-16 13:47:57.290: PATH =>
"C:\Python\Python38\Scripts\;C:\Python\Python38\...;C:\instantclient_19_8"
ODPI [24384] 2020-09-16 13:47:57.414: check module directory
ODPI [24384] 2020-09-16 13:47:57.416: module name is
C:\Python\Python38\lib\site-packages\cx_Oracle.cp38-win_amd64.pyd
ODPI [24384] 2020-09-16 13:47:57.428: load in dir
C:\Python\Python38\lib\site-packages
ODPI [24384] 2020-09-16 13:47:57.432: load with name
C:\Python\Python38\lib\site-packages/oci.dll
ODPI [24384] 2020-09-16 13:47:57.434: load by OS failure: The specified
module could not be found
ODPI [24384] 2020-09-16 13:47:57.445: load with OS search heuristics
ODPI [24384] 2020-09-16 13:47:57.449: load with name oci.dll
ODPI [24384] 2020-09-16 13:47:57.451: load by OS successful
ODPI [24384] 2020-09-16 13:47:57.460: validating loaded library
>>> dbx.close()
>>> exit()
```



#US_ConsumerFinance_170MB

Abordaremos lo siguiente

- Utilizaremos un ORM (**SQLAlchemy** en este caso)
- Modulo Python **Pandas**
- Formato de Origen **CSV**
- Formato de destino **Tabla Oracle**
- Analizaremos **tiempos de carga** y tipos de **columnas** creados en destino (base de datos)
- Analizaremos y obtendremos conclusiones sobre los tiempos de carga y metadata en destino
- En caso de incidencias, buscaremos una **alternativa**



<https://www.kaggle.com/cfpb/us-consumer-finance-complaints>

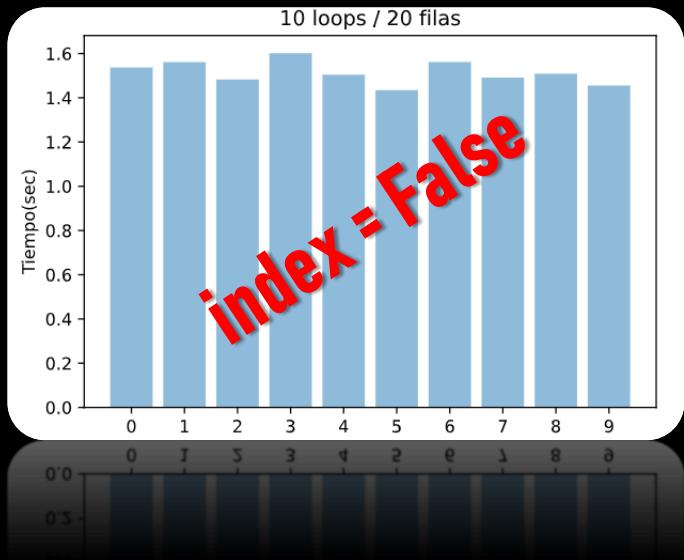


#TEST_USCFCsv2OADCLOUD

Análisis de los resultados (TTLD)

- Los tiempos de carga son de ~ 0.63 rows/sec
- Tiempo estimado total 500K ~ **20 días!**
- Para una prueba de 500K, obtenemos el error:

```
OperationalError: (cx_Oracle.OperationalError) ORA-03114: not connected to ORACLE  
(Background on this error at: http://sqlalche.me/e/13/e3q8)
```



#TEST_USCFCsv2OADCLOUD

Análisis de los resultados (TMTD)

- Se crean automáticamente campos del tipo **CLOB**
- No se crea el índice con la opción **index=False**, comportamiento correcto.
- Con la opción **index=True** se crean índices en estado **VISIBLE**, atención con **NESTED LOOPS**.
- Evitar el **error** por UC¹ y lc¹ en el nombre de la tabla, con replace se realiza el **DROP** pero luego no puede actualizar el estado de la acción realizada:

InvalidRequestError: Could not reflect: requested table(s) not available



#TEST_USCFCsv2OADCLOUD

TIPS para los tipos de datos en BD (texto)

- Utilizar el tipo **CLOB** solo cuando sea necesario según los requerimientos
- Si se requiere una columna de texto, utilizar en lo posible VARCHAR2(n **CHAR**) (**UTF8** utiliza de 1 a 4 Bytes **U+10FFFF**)
- Con las nuevas versiones de Oracle, se soporta hasta:
 - VARCHAR2(**32767** CHAR) en modo **EXTENDED**
 - VARCHAR2(4000 CHAR) en modo **STANDARD**
- Validar la configuración en BD con la sentencia:
 - SHOW PARAMETER MAX_STRING_SIZE
- Para las categorías detectadas, se puede normalizar previa o posteriormente a la carga en BD, tratando los nulos con clave normalizada 0:
 - 0=None, Mortgage=1, Credit card=2, etc
 - 0>No, 1=Yes
 - 0=None, Closed with explanation=1, Closed with non-monetary relief=2, etc



#TEST_USCFCsv2OADCLOUD

TIPS para los tipos de datos en BD (numérico)

- Por defecto se crean en base de datos los tipos **NUMBER(19)** y **FLOAT(126)** para **int64** y **float64** de python respectivamente
- El tipo FLOAT(126) tiene una precisión de 126 binario o 38 decimal
- El tipo NUMBER ocupa en base de datos entre 1 y **22 bytes**, siendo **FLOAT** un **subtipo** de NUMBER (FLOAT para compatibilidad con ANSI FLOAT)
- Para rendimiento y optimización del espacio, para el tipo float64 se puede utilizar el tipo **BINARY_DOUBLE** de ORACLE, el cual aporta mayor rendimiento y menor coste de almacenamiento general (8 BYTES)
- El tipo **BINARY_DOUBLE** es representación **binaria**, NUMBER y FLOAT son representación decimal, ejemplo:
 - `select to_char(to_number(0.1d)) from dual; -- .10000000000000001`
 - `select to_char(0.1) from dual; -- .1`
- BINARY_DOUBLE tiene representación de **NaN**, **-inf** y **+inf** alineadas con Numpy y el tipo `cx_Oracle.DB_TYPE_BINARY_FLOAT`



#TEST_USCFCsvPUREcxORACLE

Implementar una carga sin SQLAlchemy

- Analizar los datos y crear una estructura óptima.
- Validar limitación de 2GB reservado para cualquiera de los parámetros (**DPI-1015**: array size of <n> is too large).
- Crear chunks del DataFrame principal (100K/chunk)
- Preparar datos para insertar (hacernos cargo 'on the fly' de valores nulos **NaN** y **NAT** para tipos de fechas).

```
0 date_received      555957 non-null object
1 product           555957 non-null object
2 sub_product        397635 non-null object
3 issue             555957 non-null object
4 sub_issue          212622 non-null object
5 consumer_complaint_narrative 66806 non-null object
6 company_public_response 85124 non-null object
7 company           555957 non-null object
8 state              551070 non-null object
9 zipcode            551452 non-null object
10 tags               77959 non-null object
11 consumer_consent_provided 123458 non-null object
12 submitted_via      555957 non-null object
13 date_sent_to_company 555957 non-null object
14 company_response_to_consumer 555957 non-null object
15 timely_response     555957 non-null object
16 consumer_disputed? 555957 non-null object
17 complaint_id       555957 non-null int64
dtypes: int64(1), object(17)
memory usage: 76.3+ MB
None
```

```
09/16/2020, 21:46:29.114033 [INFO] Inicio de la carga en base de datos
09/16/2020, 21:48:56.841424 [INFO] Carga de 555957 filas en tiempo 147.72739052772522
09/16/2020, 21:48:57.053509 [INFO] Fin del Test
```

~ 550K filas en 148 segundos



#TEST_USCFCsvPUREcxORACLE

Conclusiones Generales de la PARTE I

- El tiempo de carga es **razonable** con cx_Oracle
- Con la limitación de **2GB** para evitar el error DPI-1015 de la capa OCI, tenemos batchs de 100K rows.
- En BBDD, el tamaño de la tabla es de 486MB, esto se debe al campo CLOB utilizado (ver **MAX_STRING_SIZE**).
- La compresión se encuentra activada con valor **COMPRESS FOR QUERY HIGH**. Mejora el rendimiento al optimizar el ancho de banda entre los nodos y las celdas en Exadata en entornos de uso intensivo general
- Durante la carga, se realiza el pre-proceso de sustitución de valores **nan** y **nat**, por lo que es una carga adicional que **no afecta significativamente** inclusive en un volumen importante de filas.
- Se deben tomar en cuenta las **latencias** entre el **origen** y la **BBDD**!
- El aumento del **t** de la carga es **lineal** con el aumento de las filas a insertar, la **transferencia de la red** es un punto crítico (minimizar este punto).



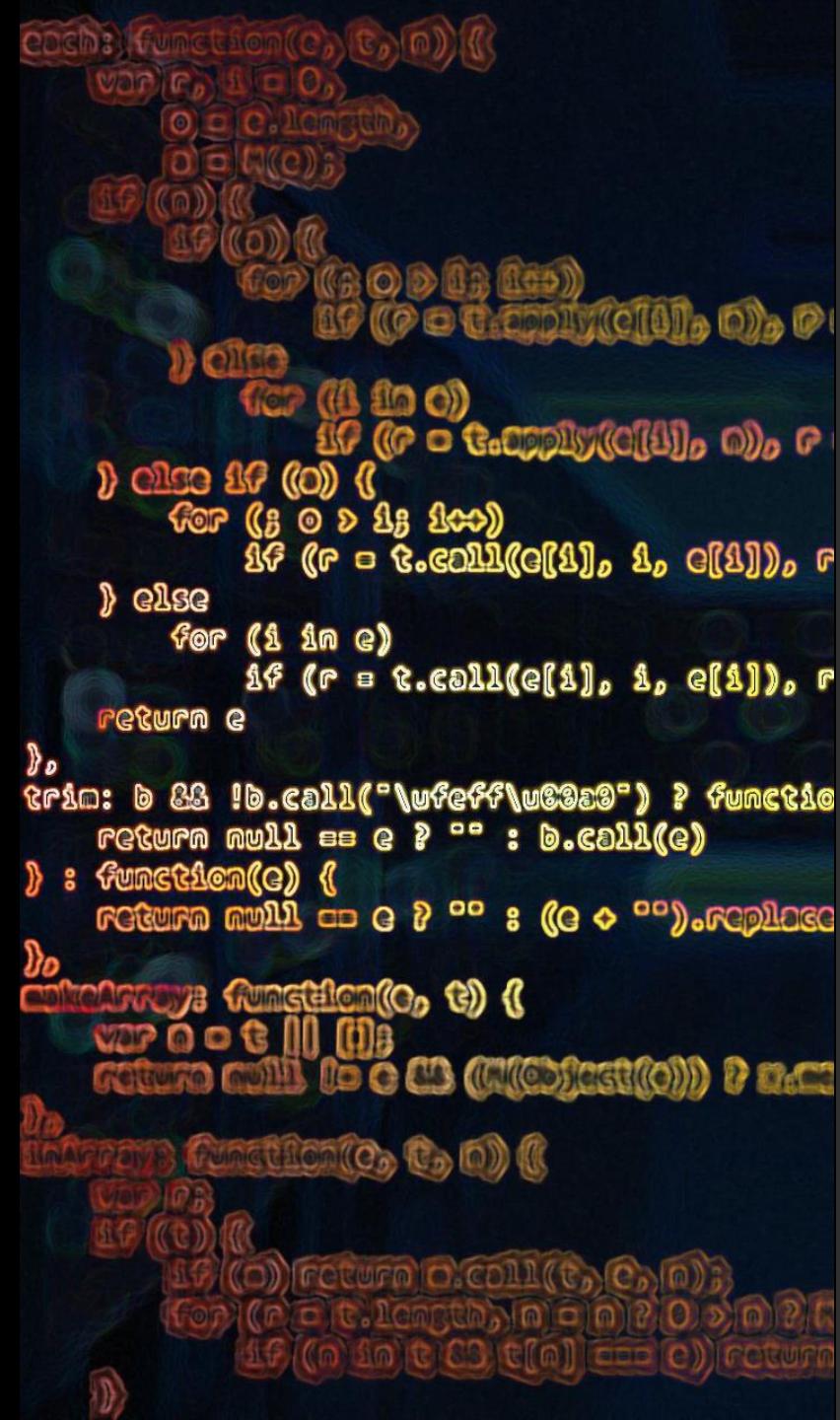
#GRACIAS_A_TODOS

Fin de la Parte I

Hemos finalizado la parte I del taller. Durante la Parte II, realizaremos ejercicios prácticos con herramientas que nos permitan agilizar el trabajo localmente o sobre servidores de proceso con grandes volúmenes de información, minimizando así la transferencia de datos en la red.

Contacto sobre consultas y dudas del Taller:

Pablo Sebastian Pereira Oromí (pspereira@dbxperts.es)



```
coch: function(c, t, r) {
    var p, i = 0,
        o = c.length,
        o = t(c);
    if (r) {
        if (t) {
            for (c = 0; i < o; i++) {
                if (r = t.apply(c[i], r), r)
                    break;
            }
        } else
            for (i in c)
                if (r = t.apply(c[i], r), r)
                    break;
    } else if (o) {
        for (i = 0; i < o; i++)
            if (r = t.call(c[i], i, c[i]), r)
                break;
    } else
        for (i in c)
            if (r = t.call(c[i], i, c[i]), r)
                break;
    return r
},
trim: b =&& !b.call("\ufe0f\ufe0a") ? function(e) {
    return null == e ? "" : b.call(e)
} : function(e) {
    return null == e ? "" : (e + "").replace(
),
makeArray: function(c, t) {
    var a = t || [];
    return null != c ? a.concat((Object(c)) ? Object(c) : [c]) : a;
},
InArray: function(c, t, r) {
    var p;
    if (t) {
        if (r) {
            if (p = c.indexOf(t, r), p >= 0)
                return p;
            for (p = c.length, n = 0; n < p; n++)
                if (c[n] === t)
                    return n;
        } else
            for (p = c.length, n = 0; n < p; n++)
                if (c[n] === t)
                    return n;
    } else
        for (p = c.length, n = 0; n < p; n++)
            if (c[n] === t)
                return n;
    return -1;
}
}
```