

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y sistemas
Laboratorio de Seminario de Sistemas 2

Wilfred Stewart Perez Solorzano
201408419

Practica 1

Nombre: Wilfred Stewart Perez Solorzano

Carné: 201408419

DPI: 2990271470101

Objetivos

Objetivos Generales

1. Aprender el proceso de ETL
2. Brindar resultados con la información obtenida

Objetivos Específicos

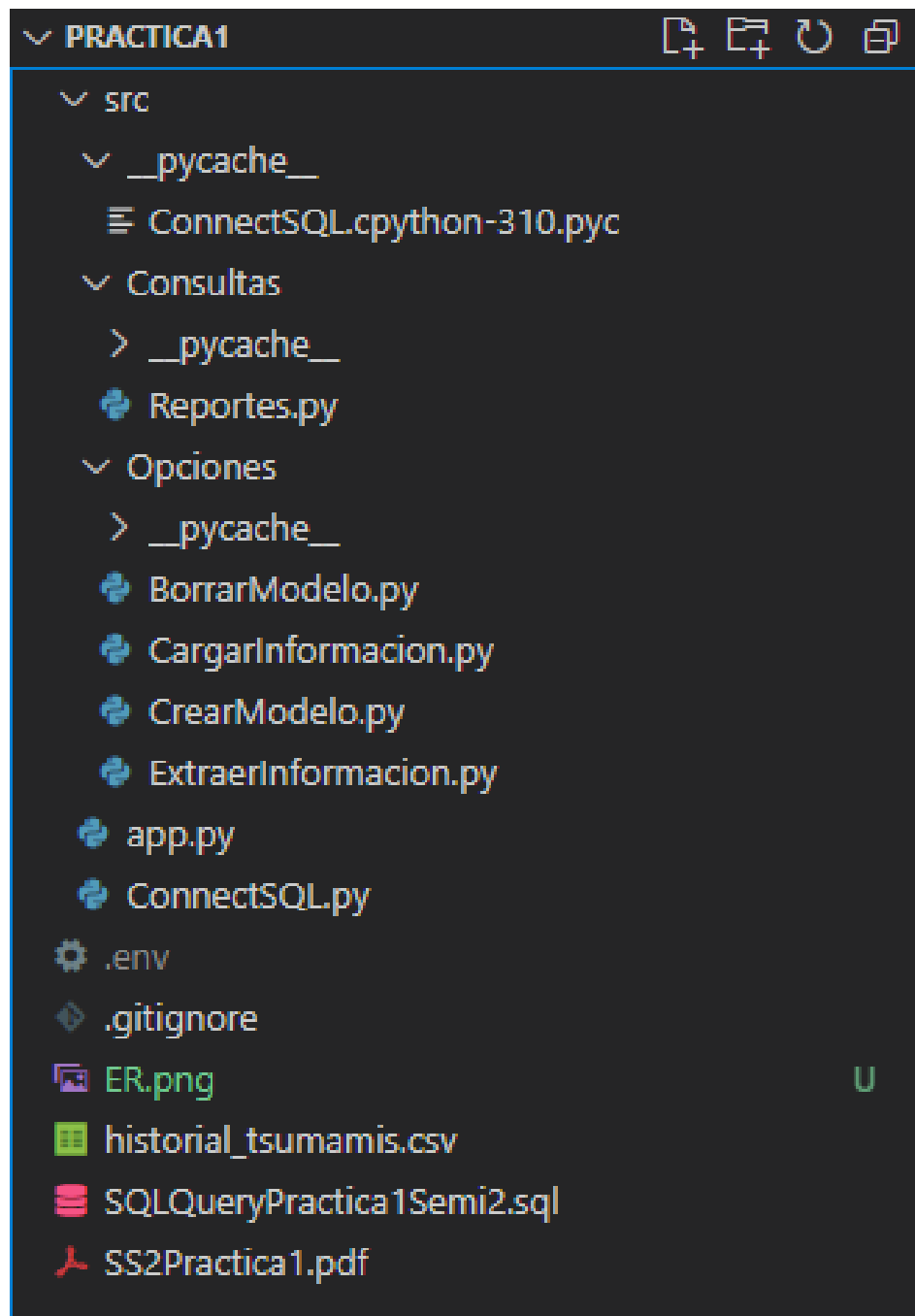
1. Utilizar el lenguaje Python para el procesamiento de información.
2. Limpiar datos.
3. Utilizar SQL Server para la creación de un Datawarehouse.

Descripción

Descripción General

Se le solicita que realice un proceso de ETL (Extraer, Transformar, Cargar) de los archivos que se les brindaran. En base a los archivos usted deberá generar el modelo que se adapte mejor, y posteriormente, realizar las consultas que se le soliciten.

Clases utilizadas



Class Reportes.py

```
src > Consultas > Reportes.py > ...
1  import pandas as pd
2  import numpy as np
3  import ConnectSQL as connect
4
5  def Consulta(num):
6
7      if num == '1':
8          conn = connect.connect()
9          df = pd.read_sql('' select (select count(*) from Tsunami) as Tsunami,(select count(*) from Lugar)
10         print('\n\n\n\n\n')
11         print('***** Reporte 1 *****')
12         print(df.to_string())
13         print('\n\n\n\n')
14
15
16         elif num == '2':
17             conn = connect.connect()
18             df = pd.read_sql('' select Anio as year, sum(ID_Tsunami) as No_Tsunamis from Tsunami
19             inner join Tiempo on Tiempo.ID_tiempo = Tsunami.FK_id_tiempo
20             group by Anio
21             order by year; '',conn)
22             print('\n\n\n\n\n')
23             print('***** Reporte 2 *****')
24             print(df.to_string())
25             print('\n\n\n\n')
26
27
28         elif num == '3':
29             conn = connect.connect()
30             df = pd.read_sql('' select Anio as year, sum(ID_Tsunami) as No_Tsunamis from Tsunami
31             inner join Tiempo on Tiempo.ID_tiempo = Tsunami.FK_id_tiempo
32             group by Anio
33             order by year; '',conn)
34             print('\n\n\n\n\n')
35             print('***** Reporte 3 *****')
36             print(df.to_string())
37             print('\n\n\n\n')
38
39
40         elif num == '4':
41             conn = connect.connect()
42             df = pd.read_sql('' select Lugar.Pais, avg(Tsunami.Total_damage) as Promedio_Total_Damage from Tsu
43             inner join Lugar on Lugar.ID_lugar = Tsunami.FK_id_lugar
44             group by Lugar.Pais
45             order by Pais; '',conn)
```

Class BorraModelo.py

```
src > Opciones > Borrarmodelo.py > Borrar
1  import ConnectSQL as connect
2
3  def Borrar():
4      conn = connect.connect()
5      conn.cursor().execute('DROP TABLE IF EXISTS Temporal;')
6      conn.cursor().execute('DROP TABLE IF EXISTS Tsunami;')
7      conn.cursor().execute('DROP TABLE IF EXISTS Lugar;')
8      conn.cursor().execute('DROP TABLE IF EXISTS Ubicacion;')
9      conn.cursor().execute('DROP TABLE IF EXISTS Tiempo;')
10     conn.commit()
11     print('Base de datos vacia')
12
```

Class CargarInformacion.py

```
src > Opciones > CargarInformacion.py > ...
1 import pandas as pd
2 import numpy as np
3 import ConnectSQL as connect
4
5
6
7
8 def Cargar(df):
9     print('Se obtubo un dataframe:')
10    print(df)
11    #conexion a la base de datos
12    #print('Prueba de conexion a la base de datos')
13
14    conn = connect.connect()
15    for row in df.itertuples():
16        #print('Consulta:\n\n')
17        #print('INSERT INTO Practica1Semi2.dbo.Temporal(Anio,Mes,Dia,Hora,Minuto,Segundo,Tsunami_event')
18        conn.cursor().execute('INSERT INTO Practica1Semi2.dbo.Temporal(Anio,Mes,Dia,Hora,Minuto,Segundo,Tsunami_event) VALUES(?, ?, ?, ?, ?, ?, ?)', row)
19        #cursor.execute('INSERT INTO Practica1Semi2.dbo.prueba (nombre,numero) values(?,?)', row)
20
21    conn.cursor().execute('insert into dbo.Tiempo(Anio,Mes,Dia,Hora,Minutos, Segundo)')
22    select distinct Temporal.Anio, Temporal.Mes, Temporal.Dia, Temporal.Hora, Temporal.Minuto, Temporal.Segundo
23
24    conn.cursor().execute('insert into dbo.Ubicacion(Latitud,Longitud)')
25    select distinct Temporal.Latitude, Temporal.Longitude from Temporal where Latitude is not null and Longitude is not null
26
27    conn.cursor().execute('insert into Lugar (Pais,Ciudad)')
28    select distinct Temporal.Country, Temporal.Location_name from Temporal where Country is not null and Location_name is not null
29
30    conn.cursor().execute('Insert into Tsunami(FK_id_lugar ,FK_id_tiempo ,FK_id_ubicacion ,Tsunami_event_validity,Tsunami_event_id)')
31    select distinct ID_lugar, ID_tiempo, ID_ubicacion, Tsunami_event_validity, Tsunami_event_id
32    inner join Lugar on Lugar.Pais = Temporal.Country and Lugar.Ciudad = Temporal.Location_name
33    inner join Tiempo on Tiempo.Anio = Temporal.Anio and Tiempo.Mes = Temporal.Mes and Tiempo.Dia = Temporal.Dia and Tiempo.Hora = Temporal.Hora and Tiempo.Minutos = Temporal.Minuto and Tiempo.Segundo = Temporal.Segundo
34    inner join Ubicacion on Ubicacion.Latitud = Temporal.Latitude and Ubicacion.Longitud = Temporal.Longitude
35
36    conn.commit()
37    conn.close()
38
```

Class CargarModelo.py

```
src > Opciones > CrearModelo.py > Crear
1  import ConnectSQL as connect
2
3  def Crear():
4      conn = connect.connect()
5      conn.cursor().execute('''create Table Temporal(
6          Anio VARCHAR(100) NULL,
7          Mes VARCHAR(100) NULL,
8          Dia VARCHAR(100) NULL,
9          Hora VARCHAR(100) NULL,
10         Minuto VARCHAR(100) NULL,
11         Segundo VARCHAR(100) NULL,
12         Tsunami_event_validity VARCHAR(100) NULL,
13         Tsunami_cause_code VARCHAR(100) NULL,
14         Earthquake_magnitude VARCHAR(100) NULL,
15         Deposits VARCHAR(100) NULL,
16         Latitude VARCHAR(100) NULL,
17         Longitude VARCHAR(100) NULL,
18         Maximun_water_height VARCHAR(100) NULL,
19         Number_of_runups VARCHAR(100) NULL,
20         Tsunami_magnitude VARCHAR(100) NULL,
21         Tsunami_intensity VARCHAR(100) NULL,
22         Total_deaths VARCHAR(100) NULL,
23         Total_missing VARCHAR(100) NULL,
24         Total_missing_description VARCHAR(100) NULL,
25         Total_injuries VARCHAR(100) NULL,
26         Total_damage VARCHAR(100) NULL,
27         Total_damage_description VARCHAR(100) NULL,
28         Total_houses_destroyed VARCHAR(100) NULL,
29         Total_houses_damaged VARCHAR(100) NULL,
30         Country VARCHAR(100) NULL,
31         Location_name VARCHAR(100) NULL
32     )''')
33     conn.cursor().execute('''Create Table Tiempo(
34         ID_tiempo int Not null identity(1,1) primary key,
35         Anio int NULL,
36         Mes int NULL,
37         Dia int NULL,
38         Hora float NULL,
39         Minutos float NULL,
40         Segundo float NULL
41     )''')
42     conn.cursor().execute('''Create Table Ubicacion(
43         ID_ubicacion int Not null identity(1,1) primary key,
44         Latitud float NULL,
45         Longitud float NULL
```

Class ExtraerInformacion.py

```
src > Opciones > ExtraerInformacion.py > Extraercsv
1  import pandas as pd
2  import numpy as np
3
4  def Extraercsv():
5
6      Contenedocsv = pd.read_csv('F:/WillOP/u/Seminario2/Laboratorio/Practica1/historial_tsumamis.csv')
7      df = pd.DataFrame(Contenedocsv)
8      #print('//////////////////////////////// Imprimiendo df //////////////////////////////////')
9      #print(df)
10     #print('//////////////////////////////// Imprimiendo df.astype //////////////////////////////////')
11     df = df.replace({np.nan:0})
12     df = df.replace("\"", '')
13     df = df.replace("\'", '')
14     df = df.iloc[1: , :] #Eliminando la primera fila
15
16     return df
```

Class app.py

```
src > app.py > ...
1  from asyncio.windows_events import NULL
2  import pandas as pd
3  import Opciones.BorrarModelo as opcion1
4  import Opciones.CrearModelo as opcion2
5  import Opciones.ExtraerInformacion as opcion3
6  import Opciones.CargarInformacion as opcion4
7  import Consultas.Reportes as Reportes
8
9
10 def menu():
11     print('***** MENU *****')
12     print('* 1. Borrar modelo      *')
13     print('* 2. Crear modelo         *')
14     print('* 3. Extraer información   *')
15     print('* 4. Cargar información     *')
16     print('* 5. Realizar consultas    *')
17     print('* 0. Salir                 *')
18     print('*****')
19
20
21 if __name__ == '__main__':
22     bandera = True
23     informacion = NULL
24     while bandera:
25         menu()
26         opcion = input()
27         if opcion == '1':
28             print('Borrar modelo')
29             opcion1.Borrar()
30         elif opcion == '2':
31             print('Crear modelo')
32             opcion2.Crear()
33         elif opcion == '3':
34             print('Extraer información')
35             informacion = opcion3.Extraercsv()
36             print('Se extrajo la siguiente información')
37             print(informacion)
38         elif opcion == '4':
39             print('Cargar Informacion')
40             opcion4.Cargar(informacion)
41         elif opcion == '5':
42             print('Consultas')
43             print('1. Debe mostrar un SELECT COUNT(*) de todas las tablas para ver que si realizo la ca')
44             print('2. Cantidad de tsunamis por año. ')
45             print('3. Tsunamis por país y que se muestren los años que han tenido tsunamis')
```

Class ConnectSQL.py

```
src > ConnectSQL.py > ...
1  import pyodbc
2  from decouple import config
3
4  def connect():
5      try:
6          conn = pyodbc.connect(
7              'Driver={SQL Server};'
8              'Server='+config('SERVER_SQL')+';'
9              'Database='+config('DATABASE')+';'
10             'UID='+config('USER')+';'
11             'Trusted_Connection=yes;'
12          )
13          print('Conexion exitosa')
14          return conn
15     except Exception as e:
16         print('Ocurrio un error al conectar con sql server: ',e)
```


Query.sql

```
CREATE DATABASE Practica1Semi2;

use Practica1Semi2;

create Table Temporal(
    Anio VARCHAR(100) NULL,
    Mes VARCHAR(100) NULL,
    Dia VARCHAR(100) NULL,
    Hora VARCHAR(100) NULL,
    Minuto VARCHAR(100) NULL,
    Segundo VARCHAR(100) NULL,
    Tsunami_event_validity VARCHAR(100) NULL,
    Tsunami_cause_code VARCHAR(100) NULL,
    Earthquake_magnitude VARCHAR(100) NULL,
    Deposits VARCHAR(100) NULL,
    Latitude VARCHAR(100) NULL,
    Longitude VARCHAR(100) NULL,
    Maximun_water_height VARCHAR(100) NULL,
    Number_of_runups VARCHAR(100) NULL,
    Tsunami_magnitude VARCHAR(100) NULL,
    Tsunami_intensity VARCHAR(100) NULL,
    Total_deaths VARCHAR(100) NULL,
    Total_missing VARCHAR(100) NULL,
    Total_missing_description VARCHAR(100) NULL,
    Total_injuries VARCHAR(100) NULL,
    Total_damage VARCHAR(100) NULL,
    Total_damage_description VARCHAR(100) NULL,
    Total_houses_destroyed VARCHAR(100) NULL,
    Total_houses_damaged VARCHAR(100) NULL,
    Country VARCHAR(100) NULL,
    Location_name VARCHAR(100) NULL
)

delete from Temporal;
select * from Temporal;
DROP TABLE IF EXISTS Temporal;

select * from Tiempo;
Create Table Tiempo(
    ID_tiempo int Not null identity(1,1) primary key,
    Anio int NULL,
```

Variables de entorno

```
🔧 .env
1  SERVER_SQL = DESKTOP-Q05E3IP\SQLEXPRESS
2  DATABASE = Practica1Semi2
3  USER = DESKTOP-Q05E3IP\will0P
```

Git ignore

.env

Modelo Entidad Relación

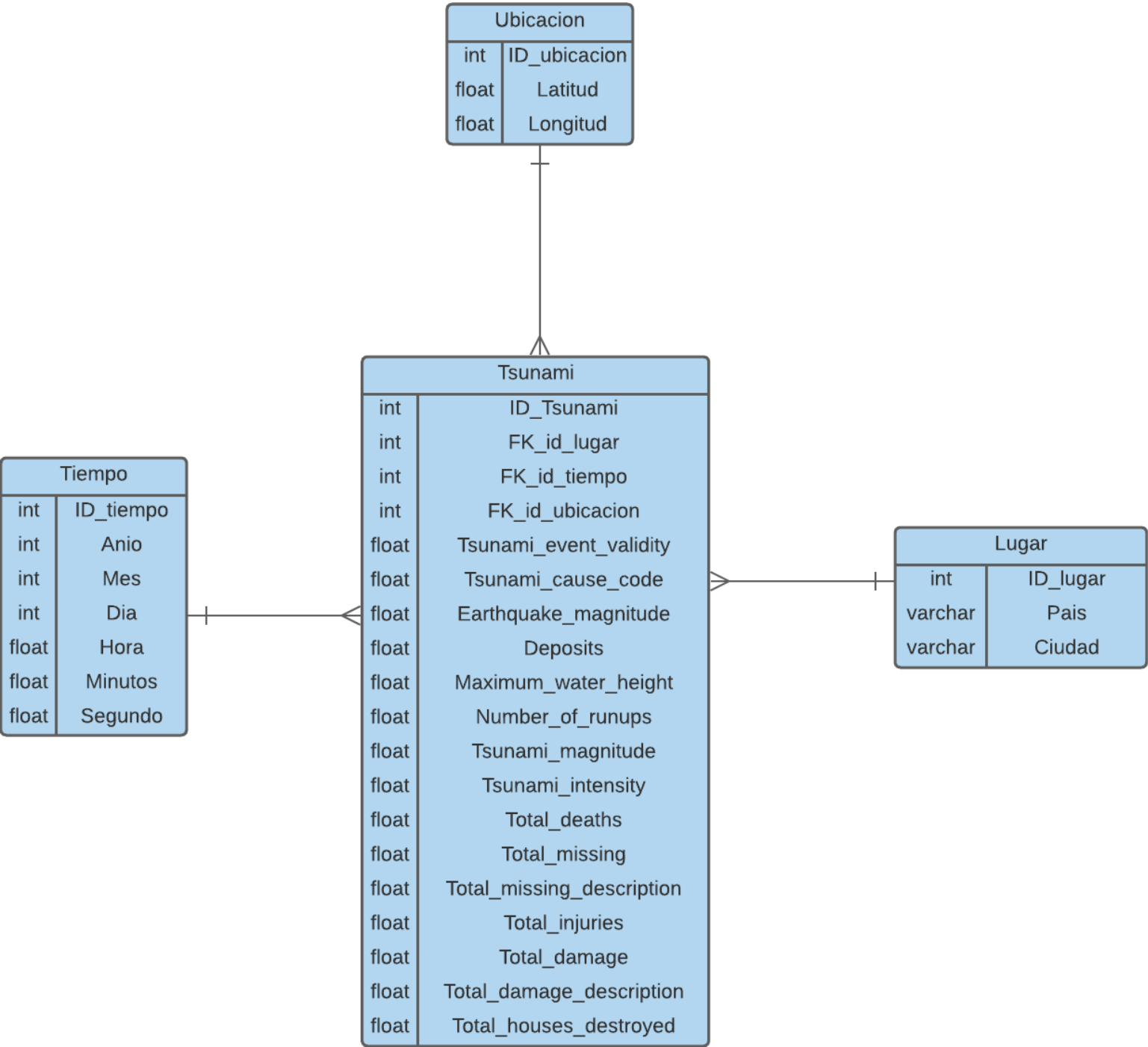
Tabla de Hechos

- Tsunami

Tablas de Dimensiones

- Ubicación
- Tiempo
- Lugar

Diagrama Entidad Relación



Consultas por realizar

1. Debe mostrar un SELECT COUNT(*) de todas las tablas para ver que si realizo la carga en las tablas del modelo.
2. Cantidad de tsunamis por año.
3. Tsunamis por país y que se muestren los años que han tenido tsunamis de la siguiente forma:
País Año 1 Año 2 Año 3 Guatemala 1901 1902 1903.
4. Promedio de Total Damage por país.
5. Top 5 de países con más muertes.
6. Top 5 de años con más muertes.
7. Top 5 de años que más tsunamis han tenido.
8. Top 5 de países con mayor número de casas destruidas.
9. Top 5 de países con mayor número de casas dañadas.
10. Promedio de altura máxima del agua por cada país.