



UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD
DEL CUSCO

FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA,
INFORMÁTICA Y MECÁNICA
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE
SISTEMAS



ASIGNATURA: FUNDAMENTOS Y DISEÑO DE BASE DE DATOS
PROYECTO DE FIN DE SEMESTRE

DOCENTE: MG. ROBERT WILBERT ALZAMORA PAREDES
ELABORADO POR: JOHAN WILFREDO HUAMAN MENDOZA

SEMESTRE 2020 - II
CUSCO, PERÚ



ÍNDICE

DESCRIPCIÓN DE LOS ARCHIVOS ORIGINALES	3
ARCHIVO positivos_covid.csv:	3
Sobre las columnas:	3
Sobre las filas:	3
ARCHIVO fallecidos_covid.csv:	3
Sobre las columnas:	3
Sobre las filas:	3
ARCHIVO geodir-ubigeo-inei.csv:	4
Sobre las columnas:	4
Sobre las filas:	4
MODELAMIENTO DE LA BASE DE DATOS	4
MODELO CONCEPTUAL	4
MODELO LÓGICO	5
MODELO FÍSICO	5
MIGRACIÓN DE DATOS	6
TUbigeo	6
TPositivo, TFallecido	7
TPersona_Positivo, TPersona_Fallecido	7
Verificación	7
IMPLEMENTACIÓN DE SERVICIOS	7
IMPLEMENTACIÓN DE SERVICIOS ADICIONALES	11
CONCLUSIONES	14
BIBLIOGRAFÍA	14



DESCRIPCIÓN DE LOS ARCHIVOS ORIGINALES

Los archivos originales utilizados son dos, positivos_covid.csv y fallecidos_covid.csv; estos fueron descargados de la página datos abiertos del gobierno de Perú el 31 de enero del 2021.

Para el ubigeo se utilizó el archivo geodir-ubigeo-inei.csv el cual se obtuvo del repositorio geodir/ubigeo-peru de github.

1.1. ARCHIVO positivos_covid.csv:

El archivo positivos_covid.csv cuenta con un millón ciento cuarenta y dos mil setecientos diecisiete (1142717) filas de las cuales la primera contiene los títulos de cada columna y las siguientes filas contienen los datos recopilados.

1.1.1. Sobre las columnas:

El archivo contiene nueve columnas de las cuales los títulos son: FECHA_CORTE, UUID, DEPARTAMENTO, PROVINCIA, DISTRITO, METHODODX, EDAD, SEXO y FECHA_RESULTADO.

1.1.2. Sobre las filas:

Cada fila contiene a lo más nueve datos separados entre sí por un punto y coma. Las columnas FECHA_CORTE, UUID, DEPARTAMENTO, METHODODX, SEXO y FECHA_RESULTADO son no nulos, las columnas PROVINCIA y DISTRITO pueden contener el texto: EN INVESTIGACION; y finalmente la columna de EDAD en algunas filas viene nula, vacía.

Tipos de archivo: csv

1.2. ARCHIVO fallecidos_covid.csv:

El archivo fallecidos_covid.csv cuenta con un cuarenta y un mil ciento ochenta y dos (41182) filas de las cuales la primera contiene los títulos de cada columna y las siguientes filas contienen los datos recopilados.

1.2.1. Sobre las columnas:

El archivo contiene nueve columnas de las cuales los títulos son: FECHA_CORTE, UUID, FECHA_FALLECIMIENTO, EDAD_DECLARADA, SEXO, FECHA_NAC, DEPARTAMENTO, PROVINCIA, DISTRITO.

1.2.2. Sobre las filas:

Cada fila contiene a lo más nueve datos separados entre sí por un punto y coma. Las columnas FECHA_CORTE, UUID, FECHA_FALLECIMIENTO, SEXO y DEPARTAMENTO son no nulos, las columnas EDAD_DECLARADA, FECHA_NAC, PROVINCIA y DISTRITO datos nulos, vacíos.



1.3. ARCHIVO geodir-ubigeo-inei.csv:

El archivo geodir-ubigeo-inei.csv cuenta con mil ochocientos setenta y cinco (1875) filas de las cuales la primera contiene los títulos de cada columna y las siguientes filas contienen los datos de ubigeo.

1.2.3. Sobre las columnas:

El archivo contiene ocho columnas de las cuales los títulos son: Ubigeo, Distrito, Provincia, Departamento, Poblacion, Superficie, Y y X; donde Y, X representa latitud y longitud.

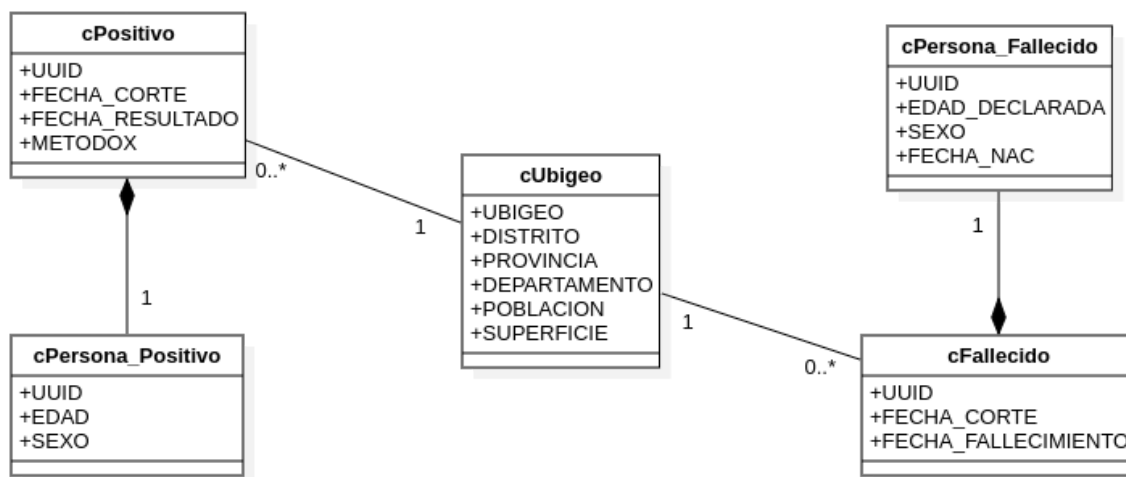
1.2.4. Sobre las filas:

Cada fila contiene ocho datos separados entre sí por un punto y coma. No existen datos nulos en cualquiera de las columnas.

MODELAMIENTO DE LA BASE DE DATOS

2.1. MODELO CONCEPTUAL

El modelo conceptual consta de cinco clases, por un lado se tienen cPersona_Positivo y cPositivo en las cuales se guardarán los datos del archivo positivos_covid.csv. Por otro lado se tienen cPersona_Fallecido y cFallecido, donde se guardarán los datos del archivo fallecidos_covid.csv. Como clase intermedia o enlace está la clase cUbigeo en la cual se guardarán los datos de ubigeo.





2.2. MODELO LÓGICO

	Restricciones
TPersona_Fallecido (<u>UUID</u> , EDAD, SEXO)	SEXO no nulo
TPositivo (<u>UUID</u> , <u>UBIGEO</u> , FECHA_CORTE, FECHA_RESULTADO, METODOX)	Todos los atributos son no nulos
TUbigeo (<u>UBIGEO</u> , DISTRITO, PROVINCIA, DEPARTAMENTO, POBLACION, SUPERFICIE)	DEPARTAMENTO no nulo
TFallecido (<u>UUID</u> , <u>UBIGEO</u> , FECHA_CORTE, FECHA_FALLECIDO)	Todos los atributos son no nulos
TPersona_Fallecido (<u>UUID</u> , EDAD_DECLARADA, SEXO, FECHA_NAC)	SEXO no nulo

2.3. MODELO FÍSICO

cPersona_Positivo					
Atributo	Tipo	Longitud	PK	Integridad referencial (FK)	Integridad de dominio / Restricciones
UUID	varchar	32	SI	cPostivo	
EDAD	int				
SEXO	varchar	10			no nulo
cPostivo					
Atributo	Tipo	Longitud	PK	Integridad referencial (FK)	Integridad de dominio / Restricciones
UUID	varchar	32	SI		
UBIGEO	int			cUbigeo	no nulo
FECHA_CORTE	fecha				no nulo
FECHA_RESULTADO	fecha				no nulo
METODOX	varchar	10			no nulo
cUbigeo					
Atributo	Tipo	Longitud	PK	Integridad referencial (FK)	Integridad de dominio / Restricciones
UBIGEO	int		SI		
DISTRITO	varchar	50			
PROVINCIA	varchar	50			
DEPARTAMENTO	varchar	50			no nulo
POBLACION	int				
SUPERFICIE	varchar	20			



cFallecido					
Atributo	Tipo	Longitud	PK	Integridad referencial (FK)	Integridad de dominio / Restricciones
UUID	varchar	32	SI		
UBIGEO	int			cUbigeo	no nulo
FECHA_CORTE	fecha				no nulo
FECHA_FALLECIMIENTO	fecha				no nulo
cPersona_Fallecido					
Atributo	Tipo	Longitud	PK	Integridad referencial (FK)	Integridad de dominio / Restricciones
UUID	varchar	32	SI	cFallecido	
EDAD_DECLARADA	int				
SEXO	varchar	10			no nulo
FECHA_NAC	fecha				

MIGRACIÓN DE DATOS

Para la migración de datos de csv a sql se utilizó **BULK INSERT**, esta instrucción de sql permite importar archivos de datos a una tabla de sql server.

La estructura sql utilizada fue:

```
BULK INSERT nombre_tabla
FROM 'directorio_archivo'
WITH
(
    CODEPAGE = ';;'
    , FIRSTROW = 2
    , FIELDTERMINATOR = ';'
    , ROWTERMINATOR = '\n'
);
```

Antes de realizar el importe de datos se revisó que los datos de los archivos se encuentren correctos, se quitaron abreviaciones y se estandarizaron las escrituras, específicamente los datos de ubicación ya que se encontraron errores de digitación, abreviaturas y diferencias en tildes entre los datos del archivo geodir-ubigeo-inei.csv y los archivos positivos_covid.csv y fallecidos_covid.csv.

Para realizar la importación primero se crearon tablas temporales y a partir de estas se repartió a las diferentes tablas de la base de datos.

Primero se crearon 3 tablas temporales

#TTempUbigeo, #TTempPositivos y #TTempFallecidos, en estas se importó todos los datos correspondientes a cada tabla.

3.1. TUbigeo

Los datos de la tabla TUbigeo se rellenaron con las todas las columna de la tabla temporal #TTempUbigeo excepto por las columnas Y y X.



También se creó la tabla temporal #TTempUbigeo2 en la cual se guardó los departamentos y sus códigos respectivos ya que mucha de la data de los archivos csv contiene datos incompletos respecto a las columnas de distrito y provincia.

Al final se insertaron las filas de #TTempUbigeo2 a la tabla TUbigeo

3.2. TPositivo, TFallecido

Primero se insertó las filas que contienen datos completos de ubicación (distrito, provincia y departamento).

Luego se hizo una unión de la tabla temporal #TTempUbigeo2 y las tablas temporales #TTempPositivos y #TTempFallecidos de filas incompletas de ubicación para insertar los datos restantes en las tablas TPositivo y TFallecido respectivamente.

3.3. TPersona_Positivo, TPersona_Fallecido

Estas tablas no contienen información de ubicación por lo que se utilizó directamente las tablas #TTempPositivos y #TTempFallecidos para insertar los datos en TPersona_Positivo y TPersona_Fallecido respectivamente

3.4. Verificación

Para realizar la verificación se contabilizó la cantidad de datos de cada tabla y esta se comparó con la que se obtuvo al migrar los datos, se utilizó la instrucción: **select count(*) from nombre_tabla.**

Además se comparó que las tablas TPositivo y TFallecido estuvieran relacionadas con un id UBIGEO con las instrucción: **select UUID from tabla where UBIGEO is null** para verificar que no existiera nulos.

También se utilizó la instrucción **select UUID from TPersona_tabla pt left join tabla t on pt.UUID == t.UUID where t.UUID is null**, para verificar que cada fila de las tablas TPersona_Positivo y TPersona_Fallecido estuvieran relacionadas con una fila de las tablas TPositivo y TFallecido respectivamente.

IMPLEMENTACIÓN DE SERVICIOS

4.1. Número de casos positivos por cada provincia de un determinado departamento. R (Provincia, NroCasos)

```
create function fnPostivosPorProvincia(@Departamento varchar (50))
returns table
as
return (
    select Provincia = U.PROVINCIA,
           NroCasos = sum(case when isnull(TP.UUID, 'null')='null' then 0 else 1 end)
    from TPositivo TP right join TUbigeo U
         on TP.UBIGEO = U.UBIGEO
    where isnull(U.PROVINCIA, 'null') != 'null'
         and U.DEPARTAMENTO = @Departamento
    group by U.PROVINCIA
)
```



```
19 select *
20 from fnPostivosPorProvincia('HUANUCO')
21 order by NroCasos desc
```

100 %

Results Messages

	Provincia	NroCasos
1	Huanuco	13402
2	Leoncio Prado	1762
3	Puerto Inca	1292
4	Ambo	759
5	Huamalies	659
6	Pachitea	424
7	Dos de Mayo	391
8	Marañon	198
9	Yarowilca	178
10	Huacaybamba	174
11	Lauricocha	73

- 4.2. Número de casos positivos por tipo de prueba y departamento entre dos fechas.
R(Departamento, Tipo1, Tipo2...)

```
create function fnPostivosPorPruebaDepartamento(@fechaIn date, @fechaFi date)
returns @taRespuesta table (Departamento varchar(50),
                             PR int, PCR int, AG int)
as
begin
    declare @taPositivos_Prueba_Departamento table(Metodox varchar(10),
                                                    Departamento varchar(50),
                                                    NroCasos int)

    insert into @taPositivos_Prueba_Departamento
    select TP.METODOX, U.DEPARTAMENTO,
           NroCasos = sum(case when isnull(TP.UUID, 'null')='null' then 0 else 1 end)
    from TPositivo TP inner join TUbigeo U
        on TP.UBIGEO = U.UBIGEO
    where year(TP.FECHA_RESULTADO) >= year(@fechaIn)
        and year(TP.FECHA_RESULTADO) <= year(@fechaFi)
        and month(TP.FECHA_RESULTADO) >= month(@fechaIn)
        and month(TP.FECHA_RESULTADO) <= month(@fechaFi)
        and day(TP.FECHA_RESULTADO) >= day(@fechaIn)
        and day(TP.FECHA_RESULTADO) <= day(@fechaFi)
    group by TP.METODOX, U.DEPARTAMENTO
    insert into @taRespuesta
    select Departamento, PR = isnull([PR],0), PCR = isnull([PCR], 0),
           AG = isnull([AG],0)

    from @taPositivos_Prueba_Departamento
        pivot (sum(NroCasos)
              for Metodox in ([PR],[PCR],[AG])
              ) as pvtTable

    return
end
```




```
48 select *
49 from fnPostivosPorPruebaDepartamento('2020-04-01', '2020-08-31')
50
```

	Departamento	PR	PCR	AG
1	Amazonas	10877	999	0
2	Ancash	15448	3577	0
3	Apurimac	2707	386	0
4	Arequipa	29774	4382	0
5	Ayacucho	6694	2133	0
6	Cajamarca	14245	2606	0
7	Callao	21750	7032	0
8	Cusco	10544	5262	1

Query executed... | DESKTOP-0UCG48C\SQLEXPRESS ... | DESKTOP-0UCG48C\jhno (55) | BDCovid_Peru | 00:00:02 | 25 rows

- 4.3. Número y porcentaje de fallecidos por sexo en los distritos de una provincia determinada
R(Distrito, PositivosVarones, %Varones, PositivosMujeres, %Mujeres)

```
create function fnFallecidosProvinciaSexo (@Provincia varchar(50))
returns @taRespuesta table (Distrito varchar(50),
                             FallecidosVarones int, PorcentVarones float,
                             FallecidosMujeres int, PorcentMujeres float)
as
begin
    -- Crear variables
    declare @taFallecidosPorDistrito table(Distrito varchar (50),
                                             NroFallecidos int,
                                             FallecidosVarones int,
                                             FallecidosMujeres int)

    -- Obtener datos
    insert into @taFallecidosPorDistrito
    select Distrito = U.DISTRITO,
           NroFallecidos = sum(case when isnull(F.UUID, 'null') = 'null'
                                   then 0 else 1 end),
           FallecidosVarones = sum(case when isnull(PF.SEXO, 'null') = 'null'
                                       or PF.SEXO = 'FEMENINO' then 0 else 1 end),
           FallecidosMujeres = sum(case when isnull(PF.SEXO, 'null') =
                                       'null' or PF.SEXO = 'MASCULINO'
                                       then 0 else 1 end)
    from TUbigeo U left join TFallecido F on U.UBIGEO = F.UBIGEO
    left join TPersona_Fallecido PF on F.UUID = PF.UUID
    where U.PROVINCIA = @Provincia
    group by U.DISTRITO
    insert into @taRespuesta
    select Distrito, FallecidosVarones, case when NroFallecidos = 0 then 0
                                             else FallecidosVarones*100/NroFallecidos end,
           FallecidosMujeres, case when NroFallecidos = 0 then 0
                                   else FallecidosMujeres*100/NroFallecidos end
    from @taFallecidosPorDistrito
return
end
```



84 | `select * from fnFallecidosProvinciaSexo('URUBAMBA')`

.00 %

Results Messages

	Distrito	FallecidosVarones	PorcentVarones	FallecidosMujeres	PorcentMujeres
1	Chincho	2	50	2	50
2	Huaylabamba	0	0	0	0
3	Machupicchu	0	0	0	0
4	Maras	2	100	0	0
5	Ollantayambo	1	25	3	75
6	Urubamba	6	54	5	45
7	Yucay	0	0	1	100

- 4.4. Número de positivos y fallecidos por departamento, mes a mes durante el año 2020
R(Departamento, Condición, Enero, Febrero, Marzo.....) Condición = {"Positivos", "Fallecidos"}

```
create function fnPositivosFallecidosDepartamento (@Anio int)
returns @taRespuesta table (Departamento varchar(50),
                             Condicion varchar(20),
                             Enero int, Febrero int, Marzo int, Abril int, Mayo int,
                             Junio int, Julio int, Agosto int, Septiembre int,
                             Octubre int, Noviembre int, Diciembre int)
as
begin
    -- declarar variables
    declare @PositivosResumen table (Departamento varchar(50),
                                     Mes varchar(20), Conteo int)
    declare @FallecidosResumen table (Departamento varchar(50),
                                      Mes varchar(20), Conteo int)

    --Obtencion de datos
    insert into @PositivosResumen
    select U.DEPARTAMENTO, Mes = month(P.FECHA_RESULTADO),
           Conteo = count(P.UUID)
    from TUbigeo U inner join TPositivo P on U.UBIGEO = P.UBIGEO
    where @Anio = year(P.FECHA_RESULTADO)
    group by U.DEPARTAMENTO, month(P.FECHA_RESULTADO)

    insert into @FallecidosResumen
    select U.DEPARTAMENTO, Mes = month(F.FECHA_FALLECIMIENTO),
           Conteo = count(F.UUID)
    from TUbigeo U inner join TFallecido F on U.UBIGEO = F.UBIGEO
    where @Anio = year(F.FECHA_FALLECIMIENTO)
    group by U.DEPARTAMENTO, month(F.FECHA_FALLECIMIENTO)

    -- Union
    insert into @taRespuesta
    select Departamento, Condicion = 'Positivos',
           Enero = isnull([1], 0), Febrero = isnull([2], 0),
           Marzo = isnull([3], 0), Abril = isnull([4], 0),
           Mayo = isnull([5], 0), Junio = isnull([6], 0),
           Julio = isnull([7], 0), Agosto = isnull([8], 0),
           Septiembre = isnull([9], 0),
           Octubre = isnull([10], 0), Noviembre = isnull([11], 0),
           Diciembre = isnull([12], 0)
```



```

from @PositivosResumen1
    pivot (sum(Conteo)
        for Mes
        in ([1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12])
        ) as pvtTable

union
select Departamento, Condicion = 'Fallecidos',
    Enero = isnull([1], 0), Febrero = isnull([2], 0),
    Marzo = isnull([3], 0), Abril = isnull([4], 0),
    Mayo = isnull([5], 0), Junio = isnull([6], 0),
    Julio = isnull([7], 0), Agosto = isnull([8], 0),
    Septiembre = isnull([9], 0), Octubre = isnull([10], 0),
    Noviembre = isnull([11], 0), Diciembre = isnull([12], 0)
from @FallecidosResumen1
    pivot (sum(Conteo)
        for Mes
        in ([1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12])
        ) as pvtTable

return
end

```

152 select * from fnPositivosFallecidosDepartamento(2020)

100 %

Results Messages

	Departamento	Condicion	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
1	Amazonas	Fallecidos	0	0	0	0	26	79	66	20	22	15	15	11
2	Amazonas	Positivos	0	0	0	165	618	1762	4172	5159	3752	1681	822	428
3	Ancash	Fallecidos	0	0	1	92	306	400	199	173	126	73	59	88
4	Ancash	Positivos	0	0	13	945	3802	3589	4298	6391	4227	3762	1950	2074
5	Apurimac	Fallecidos	0	0	0	0	4	16	21	24	26	25	22	21
6	Apurimac	Positivos	0	0	0	156	130	332	417	2058	1882	969	690	1041
7	Arequipa	Fallecidos	0	0	2	16	47	215	491	400	188	101	78	92
8	Arequipa	Positivos	0	0	30	803	2863	3547	8073	18870	8454	2565	1444	1862

Query executed successfully. | DESKTOP-0UCG48C\SQLEXPRESS ... | DESKTOP-0UCG48C\jhno (55) | BDCovid_Peru | 00:00:01 | 50 rows

IMPLEMENTACIÓN DE SERVICIOS ADICIONALES

5.1. Mostrar el porcentaje de la población infectada y fallecida por departamento. Esta tabla nos permite tener una idea de la cantidad de personas infectadas o fallecidas por cada 100 personas en cada departamento.

La tabla contiene las columnas R(Departamento, Condición, Conteo, Poblacion, Porcentaje).

```

create function fnResumenPorcentajePorDepartamentos (@Anio int)
returns @taRespuesta table (Departamento varchar(50),
    Condicion varchar(15),
    Conteo int, Poblacion int,
    Porcentaje decimal(10,2))

as
begin
    -- declarar variables
    declare @PositivosResumen table (Departamento varchar(50), Conteo int)
    declare @FallecidosResumen table (Departamento varchar(50), Conteo int)
    declare @HabitantesDepartamento table (Departamento varchar(50), Poblacion int)
    --Obtencion de datos
    insert into @HabitantesDepartamento
    select Departamento, sum(cast(Poblacion as int))
    from TUBigeo

```



```
where isnull(PROVINCIA, 'null') != 'null'
group by DEPARTAMENTO

insert into @PositivosResumen
select U.DEPARTAMENTO, count(P.UUID)
from TUbigeo U inner join TPositivo P on U.UBIGEO = P.UBIGEO
where @Anio = year(P.FECHA_RESULTADO)
group by U.DEPARTAMENTO

insert into @FallecidosResumen
select U.DEPARTAMENTO, count(F.UUID)
from TUbigeo U inner join TFallecido F on U.UBIGEO = F.UBIGEO
where @Anio = year(F.FECHA_FALLECIMIENTO)
group by U.DEPARTAMENTO

-- Union
insert into @taRespuesta
select PR.Departamento, Condicion = 'Positivos', Conteo, Poblacion,
      (100/cast(Poblacion as decimal(10,2)))*Conteo
from @PositivosResumen PR inner join @HabitantesDepartamento HD
on PR.Departamento = HD.Departamento

union
select Fr.Departamento, Condicion = 'Fallecidos', Conteo, Poblacion,
      (100/cast(Poblacion as decimal(10,2)))*Conteo
from @FallecidosResumen FR inner join @HabitantesDepartamento HD
on FR.Departamento = HD.Departamento

return
end
go
```

208 select * from fnResumenPorcentajePorDepartamentos (2020)

100 %

	Departamento	Condicion	Conteo	Poblacion	Porcentaje
1	Amazonas	Fallecidos	254	425832	0.06
2	Amazonas	Positivos	18559	425832	4.36
3	Ancash	Fallecidos	1517	1166180	0.13
4	Ancash	Positivos	31051	1166180	2.66
5	Apurimac	Fallecidos	159	464583	0.03
6	Apurimac	Positivos	7675	464583	1.65
7	Arequipa	Fallecidos	1630	1329801	0.12
8	Arequipa	Positivos	48511	1329801	3.65
9	Ayacucho	Fallecidos	378	711058	0.05
10	Ayacucho	Positivos	15353	711058	2.16
11	Cajamarca	Fallecidos	611	1540009	0.04
12	Cajamarca	Positivos	25446	1540009	1.65
13	Callao	Fallecidos	1971	1053028	0.19
14	Callao	Positivos	43631	1053028	4.14
15	Cusco	Fallecidos	533	1338897	0.04
16	Cusco	Positivos	25493	1338897	1.90



5.2. Mostrar el rango de edades en las que el coronavirus es más crítico, la tabla se puede utilizar para saber el rango de edades en las que se debe iniciar a vacunar a las personas, también nos sirve para identificar el rango de edades que es más propenso a contagiarse.

Contagiados y fallecidos organizados por meses y edades R(Mes, 0_19,20_39, 40_59, 60_79, 79_Mas)

```
create function fnResumenEdadesMesPorDepartamento (@departamento varchar(50),
                                                    @anio int)
returns @taRespuesta table (Mes int, MesNombre varchar(20), Condicion varchar(15),
                             Conteo0_19 float, Conteo20_39 float,
                             Conteo40_59 float, Conteo60_79 float,
                             Conteo80_Mas float)
as
begin
    -- declarar variables
    declare @Temporal1 table (Mes int, MesNombre varchar(20),
                              Condicion varchar(20), Conteo0_19 float, Conteo20_39 float,
                              Conteo40_59 float, Conteo60_79 float, Conteo79_Mas float)
    declare @Temporal2 table (Mes int, MesNombre varchar(20),
                              Condicion varchar(20), Conteo0_19 float, Conteo20_39 float,
                              Conteo40_59 float, Conteo60_79 float, Conteo79_Mas float)

    --Obtencion de datos
    insert into @Temporal1
    select Mes = month(P.FECHA_RESULTADO),
           dateName(month, DateAdd(month, month(P.FECHA_RESULTADO), -1 )),
           'Positivo',
           sum(case when Edad < 20 then 1 else 0 end),
           sum(case when Edad > 19 and Edad < 40 then 1 else 0 end),
           sum(case when Edad > 39 and Edad < 60 then 1 else 0 end),
           sum(case when Edad > 59 and Edad < 80 then 1 else 0 end),
           sum(case when Edad > 79 then 1 else 0 end)
    from TPositivo P inner join TUbigeo u on P.UBIGEO = U.UBIGEO
    inner join TPersona_Positivo PP on P.UUID = PP.UUID
    where year(P.FECHA_RESULTADO)=2020 and PP.EDAD != 0
    and U.DEPARTAMENTO = @departamento
    group by month(P.FECHA_RESULTADO)

    insert into @Temporal2
    select month(F.FECHA_FALLECIMIENTO),
           dateName(month, DateAdd(month, month(F.FECHA_FALLECIMIENTO) , -1 )),
           'Fallecido',
           sum(case when EDAD_DECLARADA < 20 then 1 else 0 end),
           sum(case when EDAD_DECLARADA > 19 and EDAD_DECLARADA < 40 then 1 else 0 end),
           sum(case when EDAD_DECLARADA > 39 and EDAD_DECLARADA < 60 then 1 else 0 end),
           sum(case when EDAD_DECLARADA > 59 and EDAD_DECLARADA < 80 then 1 else 0 end),
           sum(case when EDAD_DECLARADA > 79 then 1 else 0 end)
    from TFallecido F inner join TUbigeo u on F.UBIGEO = U.UBIGEO
    inner join TPersona_Fallecido PF on F.UUID = PF.UUID
    where year(F.FECHA_FALLECIMIENTO)=2020 and PF.EDAD_DECLARADA != 0
    and U.DEPARTAMENTO = @departamento
    group by month(F.FECHA_FALLECIMIENTO)

    insert into @taRespuesta
    select * from @Temporal1
    union
    select * from @Temporal2

return
end
go
```



```

261 select *
262 from fnResumenEdadesMesPorDepartamento('Cusco', 2020)
263 order by Mes, Condicion

```

100 %

Results Messages

	Mes	MesNombre	Condicion	Conteo0_19	Conteo20_39	Conteo40_59	Conteo60_79	Conteo80_Mas
1	3	Marzo	Fallecido	0	0	0	3	0
2	3	Marzo	Positivo	0	22	6	6	0
3	4	Abril	Fallecido	0	1	1	1	1
4	4	Abril	Positivo	10	165	101	8	0
5	5	Mayo	Fallecido	0	1	2	2	1
6	5	Mayo	Positivo	74	386	291	82	4
7	6	Junio	Fallecido	0	2	9	15	5
8	6	Junio	Positivo	83	389	226	68	9
9	7	Julio	Fallecido	1	1	24	37	8
10	7	Julio	Positivo	145	1166	1245	430	66
11	8	Agosto	Fallecido	0	5	54	113	43
12	8	Agosto	Positivo	447	4356	4354	1456	224
13	9	Septiembre	Fallecido	1	3	14	43	30
14	9	Septiembre	Positivo	313	2176	1716	640	106
15	10	Octubre	Fallecido	2	5	8	13	13
16	10	Octubre	Positivo	170	899	616	288	32
17	11	Noviembre	Fallecido	0	3	6	14	7
18	11	Noviembre	Positivo	144	618	378	172	27
19	12	Diciembre	Fallecido	2	5	6	15	12
20	12	Diciembre	Positivo	227	530	369	155	30

CONCLUSIONES

Los datos proporcionados por el gobierno peruano respecto de las personas contagiadas y fallecidas a causa del covid19 en muchos casos tienen errores los cuales deben ser corregidos para realizar un buen análisis de datos. La información que se puede obtener de esta data puede servir de mucha ayuda al momento de tomar decisiones dependiendo del nivel de información que se puede obtener al realizar los análisis. La base de datos creada tiene la posibilidad de relacionar los contagiados y fallecidos por el covid19 mediante la ubicación, también se permite relacionar estos datos mediante fechas, edades y sexo de las personas.

BIBLIOGRAFÍA

- Plataforma Nacional de Datos Abiertos Perú. (2021, 01 31). *Fallecidos por COVID-19 - [Ministerio de Salud - MINSA]*.
<https://www.datosabiertos.gob.pe/dataset/fallecidos-por-covid-19-ministerio-de-salud-minsa>
- Plataforma Nacional de San Antonio Abad del Cusco. (2021, 01 31). *Casos positivos por COVID-19 - [Ministerio de Salud - MINSA]*.
<https://www.datosabiertos.gob.pe/dataset/casos-positivos-por-covid-19-ministerio-de-salud-minsa>
- Geodir. (2019, 01 03). *Ubigeo Perú*. <https://github.com/geodir/ubigeo-peru>