



UNIVERSIDAD DEL ISTMO
FACULTAD DE INGENIERÍA

DESCRIPCIÓN BASE DE DATOS

JORGE EDUARDO GUERRERO GARCIA

ÍNDICE

1.	TABLAS	3
2.	VISTAS	18
3.	FUNCIONES	26
4.	PROCEDIMIENTOS ALMACENADOS	30
5.	TRIGGERS.....	44

1. TABLAS

a) Usuario

Atributo	Tipo	Descripción
DPI	Bigint(13)	Código único del Documento Personal de Identificación de Guatemala con 13 dígitos
Clave	Varchar(130)	Hash de la clave ingresada al momento de crear un usuario para el sistema
Nombre	Varchar(60)	Todos los nombres del usuario
Apellido	Varchar(60)	Todos los apellidos del usuario
Fecha nacimiento	Date	Fecha de nacimiento del usuario
Tipo usuario	Int(11)	Numero que indica los privilegios de los usuarios: 1 usuario normal, 2 usuario empleado, 3 usuario administrador
Inscripción	Date	Fecha en la que se realizó la creación de la cuenta en el sistema
Verificado	Boolean	Booleano evaluado al momento de ingresar a la plataforma, indicara que el usuario ha sido verificado posterior a la creación de la cuenta

Llave primaria: DPI

Llave foránea: no aplica

Índice: DPI (BTREE)

Descripción de la tabla: El propósito de esta tabla es la contendrá el registro de los usuarios que han creado cuentas dentro del sistema utilizando su DPI, el cual se utilizara como llave foránea en las demás tablas del sistema

b) Catalogo vacuna

Atributo	Tipo	Descripción
Vacuna	Varchar(60)	Nombre de la vacuna la cual será la llave foránea para la tabla vacuna, el nombre de la vacuna aparecerá en un listado al momento de asignar vacunas a un usuario y al momento de editar la vacuna desde la interfaz web
Disponibilidad	Boolean	Dependiendo del valor del campo, la vacuna aparecerá en el listado al momento de asignar vacunas a un usuario
Ndosis	Int(11)	Campo indicativo de cuantas dosis son administradas con dicha vacuna, dependiendo del valor se tomarán en cuenta o no los campos de tiempo de asignación para la 2da y 3ra dosis
Tiempo dosis 2	Int(11)	Tiempo en días para la asignación de la segunda dosis de la vacuna
Tiempo dosis 3	Int(11)	Tiempo en días para la asignación de la tercera dosis de la vacuna

Llave primaria: Vacuna

Llave foránea: no aplica

Índice: DPI (BTREE)

Descripción de la tabla: El propósito de esta tabla es el de tener el listado de vacunas las cuales va a manejar el sistema de vacunación, las vacunas no podrán ser borradas, pero la disponibilidad puede ser modificada para que esta ya no aparezca al momento de asignar una vacuna, la disponibilidad es global para todos los centros, cada centro se encargara de asignar la vacuna correspondiente que disponga el centro en ese momento.

c) Catalogo Centro

Atributo	Tipo	Descripción
Centro	Varchar(60)	Nombre de la centro el cual será la llave foránea para la tabla centro vacunación, el nombre del centro en un listado al momento de que un usuario cree una cuenta donde realizará su vacunación y al momento de editar el centro desde la interfaz web
Disponible	Boolean	Dependiendo del valor del campo, el centro de vacunación aparecerá en el listado al momento de crear una cuenta y asignar un centro
Días asignacion	Int(11)	Campo indicativo de a partir de cuantos días se empezarán a asignar fechas de vacunación en el centro al momento de que el usuario administrador asigne

		masivamente a los usuarios registrados sus fechas de vacunación
Capacidad diaria	Int(11)	Capacidad de personas que puede manejar un centro de vacunación, en el momento que se supere dicha cantidad en un centro al momento de asignar fechas masivamente, los usuarios serán asignados al siguiente día dependiendo del momento en que crearon su cuenta

Llave primaria: Centro

Llave foránea: no aplica

Índice: Centro (BTREE)

Descripción de la tabla: El propósito de esta tabla es el de tener el listado de los centros de vacunación que estarán disponibles para la asignación al momento de crear un usuario, los centros no podrán ser borrados, pero se podrá editar si dicho centro estará disponible o no por lo que si no esta disponible, un usuario ya no se podrá asignar dicho centro

d) Catalogo Enfermedad

Atributo	Tipo	Descripción
Enfermedad crónica	Varchar(80)	Nombre de una enfermedad cronica dentro del sistema

Registro	Boolean	Dependiendo del valor del campo, permitirá o no la creación de una cuenta sin cumplir el requisito de edad
----------	---------	--

Llave primaria: enfermedad cronica

Lave foránea: no aplica

Índice: Enfermedad crónica (BTREE)

Descripción de la tabla: El propósito de esta tabla es la de poseer un listado de enfermedades las cuales permitirán llevar un registro sobre las personas que posean dicha enfermedad y facilitar el proceso de vacunación a disposición del usuario administrador

e) Catalogo grupo

Atributo	Tipo	Descripción
Grupo	Varchar(80)	Nombre de un grupo prioritario dentro del sistema
Registro	Boolean	Dependiendo del valor del campo, permitirá o no la creación de una cuenta sin cumplir el requisito de edad

Llave primaria: Grupo

Lave foránea: no aplica

Índice: Grupo (BTREE)

Descripción de la tabla: El propósito de esta tabla es la de poseer un listado de grupos prioritarios los cuales permitirán llevar un registro sobre las personas que pertenezcan a dichos grupos y facilitar el proceso de vacunación a disposición del usuario administrador

f) Edad registro

Atributo	Tipo	Descripción
Fecha	Date	Campo que se tendrá una fecha como parámetro para el registro de usuarios

Llave primaria: Grupo

Llave foránea: no aplica

Índice: Grupo (BTREE)

Descripción de la tabla: El propósito de esta tabla es la de poseer la fecha que determinará si el usuario está habilitado para crear una cuenta o no, comparando si su fecha de nacimiento del usuario es menor a la fecha establecida en esta tabla, si no es menor, no se le permitirá al usuario crear la cuenta

g) Vacuna

Atributo	Tipo	Descripción
DPI	BigInt(13)	DPI del usuario
Vacuna	Varchar(60)	Vacuna asignada al usuario
Dosis 1 fecha	Date	Fecha establecida para la primera dosis
Dosis 2 fecha	Date	Fecha establecida para la segunda dosis
Refuerzo fecha	Date	Fecha establecida para la tercera dosis

Llave primaria: DPI

Llave foránea: DPI, Vacuna

Índice: DPI,Vacuna(BTREE)

Descripción de la tabla: El propósito de esta tabla es la de poseer un registro sobre las fechas asignadas para las diversas dosis de un usuario, inicialmente no estará asignada una vacuna ni fechas asignadas, al momento de que el usuario administrador asigne masivamente a los usuarios, será asignada su primera fecha y al momento de validar los datos el día de la vacunación, será asignada una vacuna y a partir de los datos de la tabla catalogo vacuna, si lo requiere será asignada la fecha de la siguiente dosis, para las siguientes dosis ya no será necesaria la asignación de vacuna.

h) Historial

Atributo	Tipo	Descripción
DPI	BigInt(13)	DPI del usuario
Dosis 1	Boolean	Registro que indica si ya fue efectuada la primera dosis de la vacuna
Dosis 1	Boolean	Registro que indica si ya fue efectuada la segunda dosis de la vacuna
Dosis 1	Boolean	Registro que indica si ya fue efectuada la tercera dosis de la vacuna

Llave primaria: DPI

Llave foránea: DPI

Índice: DPI(BTREE)

Descripción de la tabla: El propósito de esta tabla es la de poseer un registro sobre las dosis efectuadas de un usuario para tener un control sobre su proceso de vacunación y verificar si una dosis ha sido efectuada, programada o ha ocurrido una ausencia

i) Centro vacunación

Atributo	Tipo	Descripción
DPI	BigInt(13)	DPI del usuario
Centro	Varchar(80)	Centro de vacunación asignado al usuario

Llave primaria: DPI

Llave foránea: DPI, Centro

Índice: DPI,centro(BTREE)

Descripción de la tabla: El propósito de esta tabla es la de poseer un registro sobre los centro a los cuales están asignados los usuarios y realizar la verificación al momento de validar los datos para la aplicación de la vacuna

j) Enfermedad

Atributo	Tipo	Descripción
DPI	BigInt(13)	DPI del usuario
Enfermedad	Varchar(80)	Enfermedad crónica del usuario

Llave primaria: DPI

Llave foránea: DPI, Enfermedad

Índice: DPI,Enfermedad(BTREE)

Descripción de la tabla: El propósito de esta tabla es la de poseer un registro sobre todos los usuarios que poseen una enfermedad crónica

k) Grupo prioritario

Atributo	Tipo	Descripción
DPI	BigInt(13)	DPI del usuario
Grupo	Varchar(80)	Grupo prioritario al que pertenece el usuario

Llave primaria: DPI

Llave foránea: DPI, Grupo

Índice: DPI,Grupo(BTREE)

Descripción de la tabla: El propósito de esta tabla es la de poseer un registro sobre todos los usuarios que pertenecen a un grupo prioritario

l) Contacto

Atributo	Tipo	Descripción
DPI	BigInt(13)	DPI del usuario
Teléfono	int (11)	Teléfono brindado por el usuario
Email	Varchar(80)	Correo electrónico brindado por el usuario

Llave primaria: DPI

Llave foránea: DPI

Índice: DPI,Teléfono,Email(BTREE)

Descripción de la tabla: El propósito de esta tabla es la de poseer los datos de contacto del usuario que además serán utilizados para la verificación de la cuenta

m) Cola Vacuna

Atributo	Tipo	Descripción
DPI	BigInt(13)	DPI del usuario
Fecha registro	Datetime	Fecha y hora de la creación del usuario

Llave primaria: DPI

Llave foránea: DPI

Índice: DPI, Fecha registro (BTREE)

Descripción de la tabla: El propósito de esta tabla es la de registrar el momento en el que los usuarios crearon sus cuentas, esperando a que el usuario administrador realice la asignación de fechas para la primera dosis en orden dependiendo del momento en que fue creada la cuenta

n) Data csv

Atributo	Tipo	Descripción
DPI	Bigint(13)	DPI del usuario
Clave	Varchar(130)	Hash de la clave ingresada al momento de crear un usuario para el sistema
Nombre	Varchar(60)	Todos los nombres del usuario
Apellido	Varchar(60)	Todos los apellidos del usuario
Fecha nacimiento	Date	Fecha de nacimiento del usuario
Tipo usuario	Int(11)	Número que indica los privilegios de los usuarios: 1 usuario normal, 2 usuario empleado, 3 usuario administrador
Teléfono	Int(11)	Teléfono brindado por el usuario
Email	Varchar(60)	Correo electrónico brindado por el usuario
Centro	Varchar(80)	Centro de vacunación asignado al usuario
Enfermedad	Varchar(80)	Enfermedad crónica del usuario

Grupo	Varchar(80)	Grupo prioritario al que pertenece el usuario
-------	-------------	---

Llave primaria: DPI

Llave foránea: no aplica

Índice: DPI (BTREE)

Descripción de la tabla: El propósito de esta tabla es la almacenar y registra usuarios importados a través de un archivo csv brindado por el usuario administrador

o) Contenido

Atributo	Tipo	Descripción
Elemento	Varchar(255)	identificación del elemento de la pagina de noticias
Nombre imagen	Varchar(255)	Nombre de la imagen almacenada en el servidor Apache
Título	Varchar(255)	Título del elemento
Contenido	text	Texto del elemento
Url	Varchar(255)	Enlaces a páginas sobre la noticia

Llave primaria: Elemento

Llave foránea: nombre imagen

Índice: Elemento (BTREE)

Descripción de la tabla: El propósito de esta tabla es la de poseer un listado de los elementos de la sección de noticias del portal donde la información almacenada será presentada en el portal donde el usuario administrador podrá editar este contenido

p) Imágenes

Atributo	Tipo	Descripción
Nombre imagen	Varchar(255)	Nombre de la imagen almacenada en el servidor Apache

Llave primaria: Nombre imagen

Llave foránea: no aplica

Índice: Nombre imagen (BTREE)

Descripción de la tabla: El propósito de esta tabla es la de poseer un listado de los nombres de todas las imágenes importadas al servidor Apache por el usuario administrador las cuales estarán disponibles para ser publicadas en la página de noticias

q) Registro estado

Atributo	Tipo	Descripción
Código	Serial	Código de 20 dígitos para la verificación de datos sobre el proceso de vacunación
inscripción	Date	Día de la creación de la cuenta del usuario
Vacuna	Varchar(60)	Vacuna asignada al usuario
Dosis 1 fecha	Varchar(60)	Fecha asignada para la primera dosis
Dosis 1	Date	Indicativo de si la primera dosis ha sido efectuada, programada o no asignada
Dosis 2 fecha	Varchar(60)	Fecha asignada para la segunda dosis
Dosis 2	Date	Indicativo de si la segunda dosis ha sido efectuada, programada o no asignada
Refuerzo fecha	Varchar(60)	Fecha asignada para la tercera dosis
Dosis 3	Date	Indicativo de si la tercera dosis ha sido efectuada, programada o no asignada
Proceso	Varchar(80)	Indicativo acerca de si el proceso de vacunación ha sido completado o no
Creacion	Boolean	Registro del momento en el que fue generado el pdf

Llave primaria: Código

Llave foránea: no aplica

Índice: Codigo (BTREE)

Descripción de la tabla: El propósito de esta tabla es la contendrá el registro de los datos del proceso de vacunación del usuario en cierto momento, estos datos serán almacenados y pueden ser verificados en la misma sección de proceso del usuario o desde la página de información para los usuarios anónimos utilizando el código generado el cual será guardado en un pdf con el resto de datos. Estos registros serán borrados cada cierto tiempo comparando la fecha en que se creó el registro

2. VISTAS

a) Ausencia vacuna

Atributo	Tipo	Descripción
DPI	Bigint(13)	DPI del usuario
Nombre	Varchar(60)	Nombre del usuario
Apellido	Varchar(60)	Apellido del usuario
Centro	Varchar(80)	Centro del usuario
Vacuna	Varchar(60)	Vacuna
Dosis 1 fecha	Date	Fecha asignada para la primera dosis del usuario
Dosis 2 fecha	Date	Fecha asignada para la segunda dosis del usuario
Refuerzo fecha	Date	Fecha asignada para la tercera dosis del usuario

Dosis 1	Boolean	Registro de si ha sido efectuada o no la aplicación de la primera dosis
Dosis 2	Boolean	Registro de si ha sido efectuada o no la aplicación de la segunda dosis
Dosis 3	Boolean	Registro de si ha sido efectuada o no la aplicación de la tercera dosis

Descripción de la vista: El propósito de esta vista es la de realizar un filtrado sobre todas las personas de las que tienen asignadas una vacuna con su correspondiente historial y fechas de las aplicaciones para filtrar nuevamente aquellos que luego de 7 pasados de la fecha de vacunación y que no exista registro en el historial sean enviados al reporte de ausencia

b) Cola centro

Atributo	Tipo	Descripción
DPI	BigInt(13)	DPI del usuario
Fecha registro	Datetime	Fecha y hora de la creación del usuario
Centro	Varchar(80)	Centro asignado al usuario

Descripción de la vista: El propósito de esta vista es la de obtener el centro al que esta asignado al usuario para que al momento que el usuario administrador actualice masivamente las fecha, se ordenara la vista por centro y fecha en la que fue creado el usuario para ir asignando la fecha correspondiente al centro

c) Confirmación

Atributo	Tipo	Descripción
DPI	Bigint(13)	DPI del usuario
Nombre	Varchar(60)	Nombre del usuario
Apellido	Varchar(60)	Apellido del usuario
Fecha nacimiento	Date	Fecha de nacimiento del usuario
dpiVacuna	Bigint(13)	DPI de la tabla vacuna
Vacuna	Varchar(60)	Vacuna
Dosis 1 fecha	Date	Fecha asignada para la primera dosis del usuario
Dosis 2 fecha	Date	Fecha asignada para la segunda dosis del usuario
Refuerzo fecha	Date	Fecha asignada para la tercera dosis del usuario
Dosis 1	Boolean	Registro de si ha sido efectuada o no la aplicación de la primera dosis
Dosis 2	Boolean	Registro de si ha sido efectuada o no la aplicación de la segunda dosis
Dosis 3	Boolean	Registro de si ha sido efectuada o no la aplicación de la tercera dosis

Descripción de la vista: El propósito de esta vista obtener datos los cuales serán mostrados en la página de proceso relacionados a las fechas para las vacunaciones y si ya se ha realizado la aplicación de una dosis y la vacuna asignada.

d) Contacto enfermos

Atributo	Tipo	Descripción
DPI	Bigint(13)	DPI del usuario
Nombre	Varchar(60)	Nombre del usuario
Apellido	Varchar(60)	Apellido del usuario
Fecha nacimiento	Varchar(10)	Fecha de nacimiento del usuario en formato dd/mm/aaaa
Enfermedad crónica	Varchar(80)	DPI de la tabla vacuna
Centro	Varchar(80)	Centro asignado al usuario
Teléfono	Int(11)	Teléfono del usuario
Email	Varchar(80)	Correo del usuario
Dosis 1 fecha	Varchar(10)	Fecha asignada para la primera dosis en formato dd/mm/aaaa
Dosis 2 fecha	Varchar(10)	Fecha asignada para la segunda dosis en formato dd/mm/aaaa
refuerzo fecha	Varchar(10)	Fecha asignada para la tercera dosis en formato dd/mm/aaaa

Dosis 1	Varchar(40)	Indicativo de si la primera dosis ha sido efectuada, programada o no asignada
Dosis 2	Varchar(40)	Indicativo de si la segunda dosis ha sido efectuada, programada o no asignada
Dosis 3	Varchar(40)	Indicativo de si la tercera dosis ha sido efectuada, programada o no asignada

Descripción de la vista: El propósito de esta vista es la de realizar un informe sobre el proceso de las personas que tienen una enfermedad crónica por lo que se podrá observar con mas detalle el proceso donde se indica si una dosis no ha sido asignada, ha sido programado o ya ha sido efectuada y si existe una disponibilidad en algún centro es posible verificar en cual centro esta asignado la persona y se utilizan sus datos de contacto para informarle a la persona si desea realizar una vacunación extraordinaria.

e) Detalle centro

Atributo	Tipo	Descripción
DPI	Bigint(13)	DPI del usuario
Nombre	Varchar(60)	Nombre del usuario
Apellido	Varchar(60)	Apellido del usuario
Centro	Varchar(80)	Centro asignado al usuario
Vacuna	Varchar(60)	Teléfono del usuario

Fecha	Date	Fecha en la que fue aplicada una dosis
-------	------	--

Descripción de la vista: El propósito de esta vista es la de llevar un registro sobre las vacunas aplicadas, el centro en el que fueron aplicadas y la fecha, por lo que dependiendo del número de dosis, un mismo usuario aparecerá varias veces donde existirá el registro de cuando se le aplico una dosis, por lo que esta tabla será ordenada por centro y posteriormente por fecha, con el fin de realizar reportes en la interfaz web

f) Historial grupo

Atributo	Tipo	Descripción
DPI	Bigint(13)	DPI del usuario
Nombre	Varchar(60)	Nombre del usuario
Apellido	Varchar(60)	Apellido del usuario
Fecha nacimiento	Varchar(10)	Fecha de nacimiento del usuario en formato dd/mm/aaaa
Grupo	Varchar(80)	Grupo prioritario al que pertenece el usuario
Dosis 1 fecha	Varchar(10)	Fecha asignada para la primera dosis en formato dd/mm/aaaa
Dosis 2 fecha	Varchar(10)	Fecha asignada para la segunda dosis en formato dd/mm/aaaa

refuerzo fecha	Varchar(10)	Fecha asignada para la tercera dosis en formato dd/mm/aaaa
Dosis 1	Varchar(40)	Indicativo de si la primera dosis ha sido efectuada, programada o no asignada
Dosis 2	Varchar(40)	Indicativo de si la segunda dosis ha sido efectuada, programada o no asignada
Dosis 3	Varchar(40)	Indicativo de si la tercera dosis ha sido efectuada, programada o no asignada

Descripción de la vista: El propósito de esta vista es la de realizar un informe sobre el proceso de las personas que pertenecen a algún grupo prioritario por lo que se podrá observar con más detalle el proceso donde se indica si una dosis no ha sido asignada, ha sido programado o ya ha sido efectuada, se podrá comprender de una mejor manera el proceso y poder establecer prioridades sobre algún grupo prioritario en particular si existe una falta de aplicación de vacunas a dichos individuos

g) Proceso

Atributo	Tipo	Descripción
DPI	Bigint(13)	DPI del usuario
Inscripción	Varchar(60)	Fecha de la creación del usuario en formato dd/mm/aaaa
Vacuna	Varchar(60)	Vacuna asignada al usuario

Dosis 1 fecha	Varchar(10)	Fecha asignada para la primera dosis del usuario en formato dd/mm/aaaa
Dosis 1	Boolean	Registro de si ha sido efectuada o no la aplicación de la primera dosis
Dosis 2 fecha	Varchar(10)	Fecha asignada para la segunda dosis del usuario en formato dd/mm/aaaa
Dosis 2	Boolean	Registro de si ha sido efectuada o no la aplicación de la segunda dosis
Refuerzo fecha	Varchar(10)	Fecha asignada para la tercera dosis del usuario en formato dd/mm/aaaa
Dosis 3	Boolean	Registro de si ha sido efectuada o no la aplicación de la tercera dosis
proceso	Varchar(40)	Indicativo de si el proceso de vacunación esta en proceso o ha si ha sido completado

Descripción de la vista: El propósito de esta vista obtener datos los cuales ingresados en el pdf generado desde la interfaz web donde los mismos datos serán ingresados dentro de la tabla registro estado a excepción del nombre de usuario

3. FUNCIONES

a) Bool dosis

```
CREATE FUNCTION bool_dosis (dosis tinyint, fecha date) RETURNS varchar(40)
| BEGIN
| declare valor varchar(40);
|   if dosis = true then
|     set valor = 'Efectuada';
|   else
|     if fecha != '' then
|       set valor = 'Programada';
|     else
|       set valor = 'No asignada';
|     end if;
|   end if;
|   RETURN valor;
| END;
```

Esta función recibe dos parámetros siendo un bool el cual será una dosis y una fecha, la fecha proviene de la tabla vacuna y las dosis provienen de la tabla historial, a esta función se le asigna la fecha de la primera dosis con el historial de la primera dosis, y así sucesivamente con las demás, si en el historial indica que si hay vacunación, indica que fue efectuada donde el valor se almacena en la variable valor que es un varchar, si no existe registro de vacuna se comprueba si hay fecha de vacunación, si existe una fecha de vacunación indica que la vacuna esta programada en la variable valor, de lo contrario, indica que no esta asignada.

La función esta hecha con el propósito de hacer legible la información al usuario acerca del proceso de vacunación

b) Bool vacuna

```
CREATE FUNCTION bool_vacuna(vacuna_x varchar(40)) RETURNS varchar(40)
BEGIN
declare valor varchar(40);

if vacuna_x != '' then
    set valor = vacuna_x;
else
    set valor = 'No asignada';
end if;

RETURN valor;
END;
```

Esta función recibe el parámetro de la vacuna de la tabla vacuna, esta función se encarga de revisar si esta asignada una vacuna o no, si esta asignada una vacuna, regresa el nombre que ya estaba establecido, de lo contrario indica que no ha sido asignada con el propósito de legibilidad para el usuario

c) Completado

```
CREATE FUNCTION completado(dpi_x bigint(13),vacuna_x varchar(40)) RETURNS varchar(40)
BEGIN
  declare valor varchar(40);
  declare numero int(11);
  declare bool_x tinyint;

  select ndosis into numero from catalogo_vacuna where vacuna = vacuna_x;

  if numero = 1 then
    select dosis1 into bool_x from historial where dpi = dpi_x;
    if bool_x = true then
      set valor = 'Vacunación completada';
    else
      set valor = 'Vacunación en proceso';
    end if;
  elseif numero = 2 then
    select dosis2 into bool_x from historial where dpi = dpi_x;
    if bool_x = true then
      set valor = 'Vacunación completada';
    else
      set valor = 'Vacunación en proceso';
    end if;
  elseif numero = 3 then
    select dosis3 into bool_x from historial where dpi = dpi_x;
    if bool_x = true then
      set valor = 'Vacunación completada';
    else
      set valor = 'Vacunación en proceso';
    end if;
  end if;

  RETURN valor;
END;
```

Esta función acepta un dpi de la tabla usuario y una vacuna de la tabla vacuna. Con el nombre de la vacuna se obtiene del catálogo la información sobre el numero de dosis que maneja la vacuna, se compara ese numero y dependiendo del valor se extrae el valor del historial del usuario en una dosis en especifico dependiendo del numero de dosis, si la dosis ya ha sido efectuada indicara que el proceso de vacunación ha sido completado, de lo contrario indicara que aun se encuentra en proceso

d) Porcentaje enfermedad

```
CREATE FUNCTION porcentaje_enfermedad( enfermedad_x varchar(80), dosis_x varchar(30)) RETURNS varchar(100)
BEGIN
declare total int(11);
declare division float;
declare texto varchar(100);

select count(*) into total from usuario;
if dosis_x = 'dosis1' then
select ((count(*)/total)*100) into division from historial h inner join enfermedad e on h.dpi = e.dpi
where h.dosis1 = true and e.enfermedad_cronica = enfermedad_x group by e.enfermedad_cronica;
set texto = ifnull(concat( enfermedad_x, ': ',division,'%'),concat( enfermedad_x, ': 0.00%'));
elseif dosis_x = 'dosis2' then
select ((count(*)/total)*100) into division from historial h inner join enfermedad e on h.dpi = e.dpi
where h.dosis2 = true and e.enfermedad_cronica = enfermedad_x group by e.enfermedad_cronica;
set texto = ifnull(concat( enfermedad_x, ': ',division,'%'),concat( enfermedad_x, ': 0.00%'));
elseif dosis_x = 'dosis3' then
select ((count(*)/total)*100) into division from historial h inner join enfermedad e on h.dpi = e.dpi
where h.dosis3 = true and e.enfermedad_cronica = enfermedad_x group by e.enfermedad_cronica;
set texto = ifnull(concat( enfermedad_x, ': ',division,'%'),concat( enfermedad_x, ': 0.00%'));
end if;

RETURN texto;
END;
```

Esta función acepta como parámetro el nombre de la enfermedad crónica de la tabla catalogo enfermedad y un texto ingresado como varchar, inicialmente se declaran las variables, en la variable total se hará un conteo de todos los usuarios existentes en la tabla usuario, posteriormente dependiendo del valor del texto ingresado procederá a analizar una dosis en concreto, si el texto es para la primera dosis, se hace un conteo de todos los usuarios que posean la enfermedad ingresada en el parámetro y que ya hayan efectuado la primera dosis. El conteo se dividirá entre el total de usuarios y se multiplicará por 100 y se guardará en la variable división la cual es un float, posterior a eso dentro de la variable texto que es un varchar se guardará la información de las operaciones, si no existe registro de personas vacunadas con alguna enfermedad, la división resultará en un valor null, por ello si el valor resultante no es null, se concatenará la enfermedad y el porcentaje obtenido de la división, si el resultado es null, se concatenará la enfermedad con un 0%. Esta función se utilizó para la página de información en el sitio web

4. PROCEDIMIENTOS ALMACENADOS

a) Asignación dosis

```
create procedure asignacion_dosis(in dpi_x bigint(13),
                                  in dosis varchar(60)
                                  )
begin
    start transaction;

    update vacuna set vacuna = dosis where dpi = dpi_x;

    commit;
end;
```

El procedimiento toma como parámetros un dpi y un nombre de una vacuna que exista en el catalogo vacuna, se actualiza la tabla vacuna asignando la vacuna especificada al usuario

b) Asignar fecha

```
create procedure asignar_fecha()
begin
    declare done boolean default false;
    declare a bigint(13) default 0;
    declare b datetime default '';
    declare c varchar(80) default '';
    declare centro_anterior varchar(80) default '';
    declare capacidad_x int default '';
    declare dias_x int default '';
    declare contador1 int;
    declare contador2 int;
    declare curl cursor for select dpi, fecha_registro, centro FROM cola_centro;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    start transaction;
```

```

open curl;
read_loop: loop
  fetch curl into a,b,c;
  if done then
    leave read_loop;
  end if;
  if c != centro_anterior then
    set centro_anterior = c;
    select dias_asignacion, capacidad_diaria into dias_x, capacidad_x FROM catalogo_centro where centro = c;
    set contador1 = 1;
    set contador2 = 0;
    update vacuna set dosis1_fecha = DATE_ADD( now(), INTERVAL (dias_x + contador2) DAY) where dpi = a;
  else
    if contador1 > capacidad_x then
      set contador2 = contador2 + 1;
    end if;
    update vacuna set dosis1_fecha = DATE_ADD( now(), INTERVAL (dias_x + contador2) DAY) where dpi = a;
  end if;

  set contador1 = contador1 + 1;

end loop read_loop;
close curl;

delete from cola_vacuna;

commit;

end;

```

El procedimiento consta de un cursor, se declara un cursor con las columnas dpi, fecha registro y centro de la vista cola centro, esta vista esta ordenada por centro y por fecha, los resultados del fetch se almacenaran en las variables declaradas a,b y c, correspondientes a las columnas de la vista, inicialmente se declara la variable done para finalizar el curso y se inicia el cursor y dentro de este un loop, si el cursor no ha finalizado, se realiza el fetch y por medio de la variable centro anterior que inicialmente es vacio, se compara y de no ser igual, el centro anterior pasa a ser el centro obtenido del fetch, ya que el centro cambio, se almacenan los días para la asignación y la capacidad diaria del centro en las variables capacidad_x, días_x, cambiamos la variable contador1 a 1 y contador 2 a 0, el contador 1 permitirá llevar el conteo de cuantos usuarios del mismo centro se van asignando en el día y el segundo contador indicara cuantos días se le sumara a la asignación cuando se supere el limite diario, luego por medio de las variables obtenidas se le asigna al usuario su primera fecha de vacunación, antes de finalizar el ciclo se suma un valor al contador1, en el siguiente ciclo si el centro del fetch es el mismo al anterior, se verifica si el contador 1 es mayor a la capacidad diaria, si se cumple se le suma un valor al contador2, luego se le asigna la fecha de la primera dosis al usuario, se continuara el ciclo hasta que el cursor termine de recorrer la tabla, luego de haber asignado las

fechas, se eliminan todos los registros de la tabla para generar una nueva cola para la actualización masiva

c) Borrar usuario

```
Create procedure borrar_usuario(in dpi_x bigint(13))
begin
    start transaction;
    delete from centro_vacunacion where dpi = dpi_x;
    delete from contacto where dpi = dpi_x;
    delete from grupo_prioritario where dpi = dpi_x;
    delete from enfermedad where dpi = dpi_x;
    delete from vacuna where dpi = dpi_x;
    delete from historial where dpi = dpi_x;
    delete from usuario where dpi = dpi_x;
    commit;
end;
```

El procedimiento toma como parámetro un dpi de un usuario y procede a eliminar el registro de dicho usuario en cada tabla y finalmente borra el usuario de la tabla usuario

d) Cambiar usuario

```
create procedure cambiarUsuario(in dpi_x bigint(13),
                                in tipo_usuario_x varchar(60))
Begin
    start transaction;
    update usuario set tipo_usuario = tipo_usuario_x where dpi = dpi_x;
    commit;
end;
```

El procedimiento toma como parámetro un dpi de un usuario y un tipo de usuario, se actualiza la tabla usuario y se cambian los privilegios del usuario seleccionado dependiendo del valor ingresado

e) Cambiar imagen

```
create procedure cambiar_imagen(in elemento_x varchar(255),
| | | | | | | | | | in imagen_x varchar(255))
begin
| start transaction;
| update contenido set nombre_imagen = imagen_x where elemento = elemento_x;
| commit;
end;
```

El procedimiento toma como parámetro el nombre de un elemento de la tabla contenido, y el nombre de una imagen de la tabla imágenes, luego se procede a actualizar el nombre de la imagen del elemento seleccionado con la nueva imagen ingresada

f) Cambiar link

```
create procedure cambiar_link(in elemento_x varchar(255),
| | | | | | | | | | in link_x varchar(255))
begin
| start transaction;
| update contenido set url = link_x where elemento = elemento_x;
| commit;
end;
```

El procedimiento toma como parámetros un elemento de la tabla contenido y un enlace en forma de texto, luego se actualiza el valor del url con el nuevo enlace del elemento seleccionado

g) Cambiar texto

```
create procedure cambiar_texto(in elemento_x varchar(255),
| | | | | | | | | | in titulo_x varchar(255),
| | | | | | | | | | in texto_x text)
begin
| start transaction;
| update contenido set titulo = titulo_x where elemento = elemento_x ;
| update contenido set contenido = texto_x where elemento = elemento_x;
| commit;
end;
```

El procedimiento toma como parámetro el nombre de un elemento de la tabla contenido y toma el texto del nuevo título y el nuevo texto, luego se actualiza el nombre y el contenido del elemento seleccionado con los valores ingresados

h) Cambio atributos

```
create procedure cambio_atributos(in dpi_x bigint(13),
                                  in atributo varchar(60),
                                  in valor varchar(80))
Begin
  DECLARE EXIT HANDLER FOR SQLEXCEPTION
  begin
    rollback;
  end;
  start transaction;

  if atributo = 'nombre' then
    update usuario set nombre = valor where dpi = dpi_x;
  elseif atributo = 'apellido' then
    update usuario set apellido = valor where dpi = dpi_x;
  elseif atributo = 'fecha_nacimiento' then
    update usuario set fecha_nacimiento = STR_TO_DATE(valor, '%d/%m/%Y') where dpi = dpi_x;
  elseif atributo = 'telefono' then
    update contacto set telefono = valor where dpi = dpi_x;
  elseif atributo = 'email' then
    update contacto set email = valor where dpi = dpi_x;
  elseif atributo = 'enfermedad_cronica' then
    update enfermedad set enfermedad_cronica = valor where dpi = dpi_x;
  elseif atributo = 'grupo' then
    update grupo_prioritario set grupo = valor where dpi = dpi_x;
  end if;

  commit;

end;
```

El procedimiento toma como parámetros un dpi de un usuario, un atributo a cambiar y un valor correspondiente, dentro del procedimiento existe la opción de actualizar el nombre, apellido, fecha de nacimiento, teléfono, correo, enfermedad crónica o grupo, si el atributo corresponde con uno de estos, se procederá a actualizar la tabla correspondiente a ese atributo y le asignará el valor ingresado al atributo seleccionado

i) Confirmar dosis

```
create procedure confirmar_dosis(dpi_x bigint(13))
begin
    declare vacuna_x varchar(60);
    declare ndosis_x int(11);
    declare dias1_x int(11);
    declare dias2_x int(11);
    declare dosis1_x tinyint(1);
    declare dosis2_x tinyint(1);
    declare dosis3_x tinyint(1);
    start transaction;

    select vacuna into vacuna_x FROM vacuna where dpi = dpi_x;
    select ndosis, tiempo_dosis2, tiempo_dosis3 into ndosis_x, dias1_x, dias2_x from catalogo_vacuna where vacuna = vacuna_x;
    select dosis1, dosis2, dosis3 into dosis1_x, dosis2_x, dosis3_x from historial where dpi = dpi_x;
    if dosis1_x = false then
        if ndosis_x > '1' then
            update vacuna set dosis2_fecha = date_add(now(), interval dias1_x day) where dpi = dpi_x;
        end if;
        update historial set dosis1 = true where dpi = dpi_x;
    elseif dosis2_x = false then
        if ndosis_x = '3' then
            update vacuna set refuerzo_fecha = date_add(now(), interval dias2_x day) where dpi = dpi_x;
        end if;
        update historial set dosis2 = true where dpi = dpi_x;
    elseif dosis3_x = false then
        update historial set dosis3 = true where dpi = dpi_x;
    end if;
    commit;
end;
```

El procedimiento tomara como parámetro un dpi de la tabla usuario, almacenara el valor de la vacuna asignada al usuario en la variable vacuna_x, se guardarán los valores de el numero de dosis y el tiempo para las siguientes dosis correspondientes a la vacuna que posee el usuario de la tabla catalogo vacuna y se obtendrá el historial de vacunas del usuario. Se verifica el historial del usuario si posee la primera vacuna, si no esta asignada se verifican si la vacuna asignada requiere otra dosis, de ser el caso se asigna la siguiente fecha de vacunación del usuario utilizando los valores obtenidos del catálogo de vacunas, además de la verificación posterior a eso se actualiza el historial del usuario. Si se posee la primera vacuna, pero no la segunda se realiza el mismo procedimiento que el anterior, se verifica si se requiere otra dosis y la asigna según lo establecido en el parámetro y se actualiza el historial, si la ultima dosis es la que falta, únicamente se actualiza el historial.

j) Crear vacuna

```
create procedure crear_vacuna(in vacuna_x varchar(60),
                             in disponible_x varchar(10),
                             in ndosis int(11),
                             in diasDosis2 int(11),
                             in diasDosis3 int(11))
Begin
    start transaction;

    if disponible_x = 'Si' then
        INSERT INTO catalogo_vacuna (vacuna, disponibilidad, ndosis, tiempo_dosis2, tiempo_dosis3)
        VALUES (
            vacuna_x,
            true,
            ndosis,
            diasDosis2,
            diasDosis3
        );
    elseif disponible_x = 'No' then
        INSERT INTO catalogo_vacuna (vacuna, disponibilidad, ndosis, tiempo_dosis2, tiempo_dosis3)
        VALUES (
            vacuna_x,
            false,
            ndosis,
            diasDosis2,
            diasDosis3
        );
    end if;
    commit;
end;
```

El procedimiento toma como parámetros el nombre de una vacuna, un texto de si esta disponible o no, el número de dosis que maneja la vacuna, el tiempo para la segunda y tercera dosis. Se verifica si la disponibilidad es si o no, en el caso de si, se procede se insertan los valores ingresados en la tabla catalogo vacuna, con el valor de disponibilidad true, en el caso de no, se insertan los datos igualmente en la tabla catalogo centro, pero con el valor disponibilidad false

k) Disponibilidad vacuna

```
create procedure disponibilidad_vacuna(in vacuna_x varchar(60),
                                      in disponibilidad varchar(10))
Begin
    start transaction;

    if disponibilidad = 'Si' then
        update catalogo_vacuna set disponibilidad = true where vacuna = vacuna_x;
    elseif disponibilidad = 'No' then
        update catalogo_vacuna set disponibilidad = false where vacuna = vacuna_x;
    end if;
    commit;
end;
```

El procedimiento toma como parámetro un nombre de una vacuna de la tabla catalogo vacuna y un texto si o no, si el texto es si, se actualiza la tabla de catalogo vacuna estableciendo la vacuna como disponible, si el texto es no, se deshabilita la vacuna

1) Editar centro

```

create procedure editar_centro(in centro_x varchar(80),
                               in campo varchar(10),
                               in valor int(11))
Begin
    start transaction;

    if campo = 'dias' then
        update catalogo_centro set dias_asignacion = valor where Centro = centro_x;
    elseif campo = 'capacidad' then
        update catalogo_centro set capacidad_diaria = valor where Centro = centro_x;
    end if;

    commit;
end;
commit;
select * from catalogo centro;

```

El procedimiento acepta como parámetro un nombre de un centro de la tabla catalogo centro, un texto que indica el atributo a cambiar y un valor del nuevo atributo, si el texto del atributo es días, se procede a cambiar el número de días para la asignación del centro seleccionado, si el texto es capacidad, se actualiza el numero de personas que soporta diariamente el centro seleccionado

m) Editar disponibilidad

[illegible]

El procedimiento toma como parámetro un nombre de un centro de la tabla catalogo centro, y un texto si o no, si el texto es si, se actualiza la tabla de catalogo centro estableciendo que el centro seleccionado esta disponible, si el texto es no, se deshabilita el centro

n) Editar vacuna

```
create procedure editar_vacuna(in vacuna_x varchar(60),
                               in campo varchar(10),
                               in valor int(11))
Begin
    start transaction;

    if campo = 'ndosis' then
        update catalogo_vacuna set ndosis = valor where vacuna = vacuna_x;
    elseif campo = 'dosis2' then
        update catalogo_vacuna set tiempo_dosis2 = valor where vacuna = vacuna_x;
    elseif campo = 'dosis3' then
        update catalogo_vacuna set tiempo_dosis3 = valor where vacuna = vacuna_x;
    end if;
    commit;
end;
```

El procedimiento acepta como parámetro un nombre de una vacuna de la tabla catalogo vacuna, un texto que indica el atributo a cambiar y un valor del nuevo atributo, si el texto del atributo es ndosis, se procede a cambiar el número de dosis que maneja la vacuna, si el texto es dosis2, se actualiza el número de días para la asignación de la segunda dosis y si el texto el dosis3 se actualiza el número de días para la asignación de la tercera dosis

o) Generar pdf

```
create procedure generar_pdf(in codigo_x varchar(60),
                             in inscripcion_x varchar(60),
                             in vacuna_x varchar(60),
                             in dosis1_fecha_x varchar(60),
                             in dosis1_x varchar(60),
                             in dosis2_fecha_x varchar(60),
                             in dosis2_x varchar(60),
                             in dosis3_fecha_x varchar(60),
                             in dosis3_x varchar(60),
                             in proceso_x varchar(60)
                             )
begin
    start transaction;
    INSERT INTO registro_estado (codigo, inscripcion, vacuna, dosis1_fecha, dosis1,
    VALUES (
        codigo_x,
        STR_TO_DATE(inscripcion_x, '%d/%m/%Y'),
        vacuna_x,
        ifnull(STR_TO_DATE(dosis1_fecha_x, '%d/%m/%Y'), '0000-00-00'),
        dosis1_x,
        ifnull(STR_TO_DATE(dosis2_fecha_x, '%d/%m/%Y'), '0000-00-00'),
        dosis2_x,
        ifnull(STR_TO_DATE(dosis3_fecha_x, '%d/%m/%Y'), '0000-00-00'),
        dosis3_x,
        proceso_x,
        now()
    );
    commit;
end;
```

El procedimiento toma como parámetro un código serial de 20 dígitos ingresado como varchar y datos obtenidos de la vista proceso, luego se realiza un insert en la tabla registro estado con el código generado aleatoriamente, las fechas de inscripción, 1ra dosis, 2da dosis y tercera dosis son transformadas de un texto a un formato date y se ingresan los demás valores

p) Insertar imagen

```
create procedure insertar_imagen(in url_x varchar(255))
begin
    start transaction;
    insert into imagenes (
        nombre_imagen
    ) VALUES (
        url_x
    );
    commit;
end;
```

El procedimiento toma como parámetro el nombre de una imagen de cualquier formato como un texto, luego se realiza un insert en la tabla imágenes con el nombre ingresado

q) Nuevo centro

```
create procedure nuevo_centro(in centro_x varchar(80),
                             in disponible_x varchar(10),
                             in dias int(11),
                             in capacidad int(11))

Begin
start transaction;
if disponible_x = 'Si' then
INSERT INTO catalogo_centro (Centro, disponible, dias_asignacion, capacidad_diaria)
VALUES (
    centro_x,
    true,
    dias,
    capacidad
);
elseif disponible_x = 'No' then
INSERT INTO catalogo_centro (Centro, disponible, dias_asignacion, capacidad_diaria)
VALUES (
    centro_x,
    false,
    dias,
    capacidad
);
end if;
commit;

end;
```

El procedimiento toma como parámetros el nombre de un centro, un texto de si esta disponible o no, el número de días para la asignación y la capacidad diaria del centro. Se verifica si la disponibilidad es si o no, en el caso de si, se procede se insertan los valores ingresados en la tabla catalogo centro, con el valor de disponible true, en el caso de no, se insertan los datos igualmente en la tabla catalogo centro, pero con el valor disponible false

r) Registro

```
create PROCEDURE registro(in dpi_x bigint(13),
                           in clave_x varchar(130),
                           in nombre_x varchar(60),
                           in apellido_x varchar(60),
                           in fecha_nacimiento_x date,
                           in tipo_usuario_x int(11),
                           in telefono_x int(11),
                           in email_x varchar(80),
                           in centro_x varchar(60),
                           in enfermedad_x varchar(80),
                           in grupo_x varchar(80)
                           )
begin
    declare new_enfermedad int(11);
    declare new_grupo int(11);
    declare bool_enfermedad tinyint(1);
    declare bool_grupo tinyint(1);
    declare edad date;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        rollback;
    END;

    start transaction;

    select * into edad from edad_registro;

    if enfermedad_x != '' then
        select registro into bool_enfermedad from catalogo_enfermedad where enfermedad_cronica = enfermedad_x;
    else
        set bool_enfermedad = false;
    end if;

    if grupo_x != '' then
        select registro into bool_grupo from catalogo_grupo where grupo = grupo_x;
    else
        set bool_grupo = false;
    end if;

    if edad > fecha_nacimiento_x or bool_enfermedad = true or bool_grupo = true or tipo_usuario_x = '2' then
        insert into usuario (dpi ,clave, nombre, apellido, fecha_nacimiento, tipo_usuario, inscripcion, verificado)
        values (
            dpi_x,
            clave_x,
            nombre_x,
            apellido_x,
            fecha_nacimiento_x,
            tipo_usuario_x,
            now(),
            true
        );
    end if;

    insert into centro_vacunacion (dpi , centro )
    values (
        dpi_x,
        centro_x
    );
end;
```

```

insert into contacto (dpi , telefono, email )
values (
    dpi_x,
    telefono_x,
    email_x
);

insert into vacuna( dpi, vacuna, dosis1_fecha, dosis2_fecha, refuerzo_fecha)
values (
    dpi_x,
    '',
    '',
    '',
    ''
);

insert into historial( dpi, dosis1, dosis2, dosis3 )
values (
    dpi_x,
    false,
    false,
    false
);

if enfermedad_x != '' then
insert into enfermedad (dpi, enfermedad_cronica )
values (
    dpi_x,
    enfermedad_x
);
end if;

if grupo_x != '' then
insert into grupo_prioritario (dpi, grupo )
values (
    dpi_x,
    grupo_x
);
end if;

```

El procedimiento toma como parámetro todos los valores obtenidos de la creación de la cuenta, dentro de la variable edad se obtiene el valor que se encuentra en la tabla edad registro. Si el usuario ha indicado que tiene una enfermedad, dentro de la variable bool enfermedad se obtiene de la tabla catalogo enfermedad si dicha enfermedad esta habilitada para el registro, de la misma forma se verifica con el grupo prioritario en la tabla catalogo centro, posterior a eso se verifica que la fecha de nacimiento sea anterior a limite establecido por la tabla edad registro o si la enfermedad o el grupo le habilita o no el registro, si es afirmativo se procederá a insertar la data ingresada en los campos que lo requieran, otra data no especificada se trabajara con valores default, en caso de algún fallo se realiza un rollback.

s) Registro enfermedad

```
create procedure registro_enfermedad( in enfermedad_x varchar(80),
                                     in valor varchar(10))
Begin
    start transaction;
    if valor = 'Si' then
        update catalogo_enfermedad set registro = true where enfermedad_cronica = enfermedad_x;
    elseif valor = 'No' then
        update catalogo_enfermedad set registro = false where enfermedad_cronica = enfermedad_x;
    end if;
    commit;
end;
```

El procedimiento toma como parámetro una enfermedad del catalogo enfermedad y un texto si o no, si el valor del texto es si, actualiza el registro de la enfermedad seleccionada a true, si es un no, actualiza el registro con valor false

t) Registro fecha

```
create procedure registro_fecha( in fecha_x varchar(80))
Begin
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        rollback;
    END;

    start transaction;
    update edad_registro set fecha = STR_TO_DATE( fecha_x , '%d/%m/%Y');
    commit;
end;
```

El parámetro acepta una fecha en texto con el formato dd/mm/aaaa, se procede a actualizar el valor de la tabla edad registro a la fecha ingresada transformando el formato al tipo date

u) Registro grupo

```
create procedure registro_grupo( in grupo_x varchar(80),
                                in valor varchar(10))
Begin
    start transaction;
    if valor = 'Si' then
        update catalogo_grupo set registro = true where grupo = grupo_x;
    elseif valor = 'No' then
        update catalogo_grupo set registro = false where grupo = grupo_x;
    end if;
    commit;
end;
```

El procedimiento toma como parámetro una enfermedad del catálogo grupo y un texto si o no, si el valor del texto es si, actualiza el registro del grupo seleccionado a true, si es un no, actualiza el registro con valor false

5. TRIGGERS

a) Cola vacuna

```
CREATE TRIGGER cola_vacuna AFTER
INSERT ON usuario
FOR EACH ROW
BEGIN
    INSERT INTO cola_vacuna (dpi, fecha_registro)
    VALUES (NEW.dpi , now());
END;
```

El siguiente trigger esta anclado a la tabla usuario, cada vez que se cree un nuevo usuario por medio de la pagina web o por medio de una importación, tomara el momento en el que fue creado el usuario y hará un insert en la tabla cola vacuna con el dpi del usuario y el momento de la creación.

b) Import csv

```
create trigger import_csv after
insert on data_csv
for each row
Begin
insert into usuario (dpi ,clave, nombre, apellido, fecha_nacimiento, tipo_usuario, inscripcion, verificado)

values (
    New.dpi,
    New.clave,
    New.nombre,
    New.apellido,
    New.fecha_nacimiento,
    New.tipo_usuario,
    now(),
    true
);

insert into centro_vacunacion (dpi , centro )

values (
    New.dpi,
    New.centro
);

insert into contacto (dpi , telefono, email )

values (
    New.dpi,
    New.telefono,
    New.email
);

insert into contacto (dpi , telefono, email )

values (
    New.dpi,
    New.telefono,
    New.email
);

insert into vacuna( dpi, vacuna, dosis1_fecha, dosis2_fecha, refuerzo_fecha)
values (
    New.dpi,
    '',
    '',
    '',
    ''
);

insert into historial( dpi, dosis1, dosis2, dosis3 )
values (
    New.dpi,
    false,
    false,
    false
);

if New.enfermedad != '' then
insert into enfermedad (dpi, enfermedad_cronica )
values (
    New.dpi,
    New.enfermedad
);
end if;

if New.grupo != '' then
insert into grupo_prioritario (dpi, grupo )
values (
    New.dpi,
    New.grupo
);
end if;
end;
```

El siguiente trigger está anclado a la tabla data csv, cada vez que se cree un nuevo usuario por medio de una importación con un archivo csv se procederá a realizar el mismo proceso de la creación de usuarios, en esta ocasión no se realiza verificación de edad ya que el usuario administrador determina que registros incluir en el archivo, se procede a realizar los inserts en las diversas tablas con los valores existentes en la tabla y otros valores default y si existe una enfermedad o grupo prioritario se procederá a hacer un insert en las tablas correspondientes con sus valores default, este a su vez accionara el trigger de cola vacuna para la asignación de fechas de vacunación para dichos usuarios