



CRM

EN BASE DE DATOS

Autor : Sony Oyarzun
Asignatura : Taller De Desarrollo De Bases De Datos

ÍNDICE

Requerimientos	3
Estructura Base De Datos	4
1.Registro de Campañas.	5
2.Registro de Personas con datos personales	6
Estructura del XML	7
3.Cantidad de personas que postularon a las diferentes campañas.	7
Registrar cliente en una campaña	7
4.Efectividad de cada campaña a nivel de postulante y alumno.	10
5.Mantener histórico de campañas	11
6.Mantener histórico de postulaciones.	12
7.Conocer la historia completa de un estudiante	13
Datos Cliente	14
Intereses Cliente	15
Historial campañas	16
Estadísticas Cliente	17
Perfil Cliente	18
8. Acciones que se han realizado con el cliente	19
9.Mostrar efectividad una vez titulados	19
10.Conocer un listado de interesados que no quieren ser contactados por la institución	20
Datos Relevantes en Tablas	22

Requerimientos

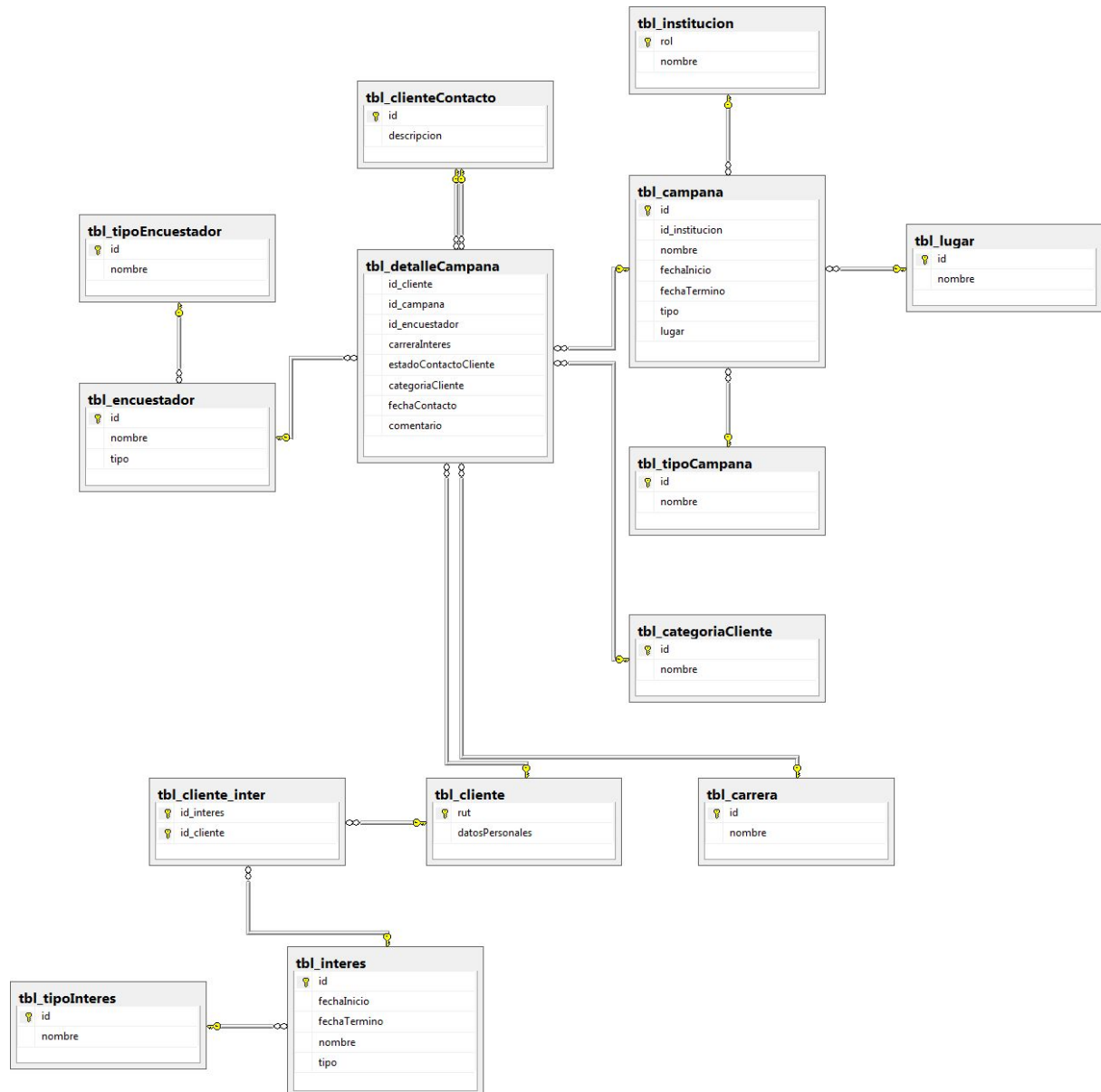
Precondiciones Obligatorias:

- a) Todas las funcionalidades necesarias deberán ser construidas basados en Procedimientos Almacenados, funciones y lo que estime conveniente para su realización.
- b) Se deberá controlar las reglas necesarias de integridad y validación de datos. Además, se deberá utilizar funciones y vistas si fuese necesario.
- c) La Base de Datos deberá operar en la modalidad **Transaccional**.
- d) Deberá entregar las políticas de respaldo y seguridad que Ud., estime conveniente para un ambiente de estas características.
- e) La base de Datos se deberá encontrar normalizada (tercera forma normal).
- f) Uso de XML (en la programación).
- g) Toda transacción registrada en la base de datos deberá contener un registro de auditoría (no se toma como transacción una inserción a una tabla maestra).

Se pide:

- 1. Registro de Campañas.
- 2. Registro de Personas con datos personales, fonos, donde viven (comuna, ciudad, región, país)
Categorizar a personas (interesado, postulante, alumno, titulado, exalumno y así puede continuar el siglo). Debe definir campañas tanto para pre-grado como postgrado.
- 3. Cantidad de personas que postularon a través de las diferentes campañas.
- 4. % de efectividad de cada campaña a nivel de postulante y alumno (venta concretada).
- 5. Mantener histórico de campañas, debe mostrar cantidad de interesados, postulantes y matriculados (con su respectivo porcentaje)
- 6. Mantener histórico de postulaciones.
- 7. Conocer la historia completa de un estudiante (desde el interés hasta la cantidad de productos y servicios que ha tenido con la institución. Una visión completa del cliente.
- 8. Mostrar las acciones que se han realizado con el interesado / postulante. (acciones = llamados telefónicos, acuerdos a los que se ha llegado, registro de acciones, agendamiento de llamadas, etc). Se conoce como seguimiento del flujo de trabajo.
- 9. Mostrar efectividad una vez titulados (esto significa de cuantos clientes se titularon y luego tomaron programas tales como cursos, diplomados, especializaciones, etc) y conocer sus tasas de efectividad.
- 10. Conocer un listado de interesados que no quieren ser contactados por la institución, debe registrar por qué no quiere ser contactado, fecha, hora y quien lo contactó.

Estructura Base De Datos



1.Registro de Campañas.

```

Alter Procedure insertarCampana
@institucion    varchar(50),
@nombreCampana  varchar(50),
@fechaInicio    varchar(50),
@FechaTermino   varchar(50),
@tipo           varchar(50),
@lugar          int
as
if Exists(Select rol From tbl_institucion where rol=@institucion)
Begin
    if Exists(Select id From tbl_tipoCampana where id = @tipo)
    Begin
        if Exists(Select id From tbl_lugar where id = @lugar)
        Begin
            Insert into tbl_campana (id_institucion,nombre,fechaInicio,fechaTermino,tipo,lugar)
            values(@institucion,@nombreCampana,@fechaInicio,@FechaTermino,@tipo,@lugar)
        End
    End
    Else
    Begin
        Print 'Lugar de la Campaña No Registrado'
    End
End
Else
Begin
    Print 'Tipo de Campaña no Registrada'
End
End
Else
Begin
    Print 'La institucion no esta Registrada'
End
End

```

Ejemplo de Inserción:

Execute insertarCampana '78212360-2','Nueva Campaña','2018-11-20','2019-04-30',1,5

Se pide como argumentos el rut de la institución el cual debe estar previamente registrado en la tabla **tbl_institucion**, el nombre de la campaña, fechas de inicio y término respectivamente, tipo de campaña, y finalmente lugar, los cuales deben estar registrados previamente en las tablas **tbl_tipoCampana** y **tbl_lugar**, sus datos se ven a continuación.

Tipo de Campaña

id	nombre
1	Pre-Grado
2	Post-Grado

Lugar

id	nombre
1	Lugar 1
2	Lugar 2
3	Lugar 3
4	Lugar 4
5	Lugar 5

2.Registro de Personas con datos personales

```
CREATE PROCEDURE INSERTAR_PERSONA
@rut varchar(12),@rutaXml Varchar(1000)
AS

IF NOT EXISTS(SELECT * FROM tbl_cliente WHERE rut=@rut)
BEGIN

DECLARE @xml xml
DECLARE @query nvarchar(max)

SELECT @query ='(SELECT @varOut = BulkColumn from Openrowset(Bulk "" + @rutaXml + "", Single_Blob)
AS x)'
EXECUTE sp_executesql @Query, N'@varOut xml out',@varOut=@xml output
INSERT INTO tbl_persona (rut,datosPersonales) VALUES(@rut, @xml)

END
ELSE
BEGIN
PRINT 'La persona ya esta registrada'
END
```

Estructura del XML

```
<Cliente>
    <nombre></nombre>
    <apellido></apellido>
    <comuna></comuna>
    <ciudad></ciudad>
    <region></region>
    <pais></pais>
    <Telefonos>
        <telefono></telefono>
    </Telefonos>
</Cliente>
```

Ejemplos de insercion:

```
execute INSERTAR_PERSONA '1-1','D:\Xml\Calvin.xml'
```

en donde se pide como argumentos el rut de la persona y la ruta donde se encuentra el xml

3.Cantidad de personas que postularon a las diferentes campañas.

Registrar cliente en una campaña

Primero se deben relacionar los clientes a las campañas con el procedimiento siguiente:

```
Alter Procedure CampanaCliente
@rut varchar(12),
@IdCampana int,
@IdEncuestador int,
@IdCarrera int,
@dispoCliente int,
@estadoCliente int,
@fecha varchar(10),
@comentario varchar(100)
as
---validar persona registrada,campaña no repetida en cliente, verificar si quieres ser llamados antes de llamar
IF Exists(Select rut from tbl_cliente Where rut=@rut)
Begin
IF Exists(Select id From tbl_campana Where id=@IdCampana)
    Begin
        IF Exists (Select id From tbl_encuestador where id = @IdEncuestador)
            Begin
                IF Exists(Select id From tbl_carrera Where id=@IdCarrera)
                    Begin
                        IF Exists(Select descripcion From tbl_clienteContacto Where id=@dispoCliente)
                            Begin
                                IF Exists(Select id From tbl_categoriaCliente Where id=@estadoCliente)
                                    Begin
                                        IF((Select fechaTermino From tbl_campana Where id=@IdCampana)>@fecha)
                                            Begin
                                                IF Not Exists(Select id_campana From tbl_detalleCampana Where id_cliente=@rut and
id_campana=@IdCampana and categoriaCliente=@estadoCliente)
                                                    Begin
                                                        Insert Into tbl_detalleCampana
(id_cliente,id_campana,id_encuestador,carreraInteres,estadoContactoCliente,categoriaCliente,fechaContacto,co
mentario)
values(@rut,@IdCampana,@IdEncuestador,@IdCarrera,@dispoCliente,@estadoCliente,@fecha,@comentario)
                                                    End
                                                ELSE
                                                    Begin
                                                        print 'Cliente ya se registro con esta categoria en la Campaña'
                                                    End
                                                End
                                            ELSE
                                                Begin
                                                    print 'Campaña fuera de Plazo'
                                                End
                                            End
                                        End
                                    End
                                End
                            End
                        End
                    End
                End
            End
        End
    End
End
ELSE
Begin
print 'Campaña fuera de Plazo'
End
End
ELSE
End
```

```

Begin
print 'Categoria no Registrada'
End
End
ELSE
Begin
print 'Disponibilidad incorrecta'
End
End
ELSE
Begin
print 'Carrera no Registrada'
End
End
ELSE
Begin
print 'Encuestador no Registrado'
End
End
ELSE
Begin
print 'Campaña No Registrada'
End
End
ELSE
Begin
print 'Rut No Registrado'
End

```

Ejemplo de Inserción :

```
execute CampanaCliente '1-5',3,2,3,2,4,'2018-11-28','No Quiere Molestad'
```

Como argumento se pide el rut del cliente, id de la campaña, id del encuestador, id de la carrera, id del estado, id de la categoría del cliente, fecha en que fue contactado, y algún comentario adicional

Mostrar Clientes por Campaña

Procedimiento individual :

```

Alter Procedure clientesXcampana @idCampana int as
if Exists (Select id From tbl_campana Where id = @idCampana)
begin
Select Count(id_cliente) From tbl_detalleCampana Where id_campana=@idCampana
end
Else
begin
Print 'Campaña no existe'
end

```

Ejemplo de Ejecución:

```
Execute clientesXcampana 1
```

Consiste ingresar la id de la campaña y devolvera la cantidad de clientes

Mostrar Cantidad de Clientes de Todas las Campaña

alter Procedure clientesXcampanaAll as

```
declare @idCampana          int
declare @nombreCampana      varchar(50)
declare @fechalnicio        varchar(50)
declare @fechaTermino       varchar(50)
declare @tipo                varchar(50)
declare @lugar               varchar(50)
declare @conteo              int

declare c1 cursor for
Select
tbl_campana.id,tbl_campana.nombre,tbl_campana.fechalnicio,tbl_campana.fechaTermino,tbl_tipoCampana.nombre,tbl_lugar.nombre
From tbl_campana,tbl_tipoCampana,tbl_lugar
where tbl_campana.tipo=tbl_tipoCampana.id
and tbl_campana.lugar=tbl_lugar.id

open c1
Fetch c1 Into @idcampana,@nombreCampana,@fechalnicio,@fechaTermino,@tipo,@lugar
While(@@FETCH_STATUS=0)
Begin
declare c2 cursor for
Select Count(id_cliente) From tbl_detalleCampana Where id_campana=@idCampana
open c2
fetch c2 into @conteo
While(@@FETCH_STATUS=0)
Begin
print '_____ '
print 'ID                               :'+Cast(@idCampana as varchar(50))
print 'Campaña                           :'+@nombreCampana
print 'Cantidad Clientes                 :'+Cast(@conteo as varchar(50))
print 'Fecha Inicio                       :'+@fechalnicio
print 'Fecha Termino                      :'+@fechaTermino
print 'Tipo Campaña                        :'+@tipo
print 'Lugar Campaña                       :'+@lugar
print '_____ '
fetch c2 into @conteo
end
close c2
deallocate c2

Fetch c1 Into @idcampana,@nombreCampana,@fechalnicio,@fechaTermino,@tipo,@lugar
End
close c1
deallocate c1
```

Ejemplo de Ejecución:

Execute clientesXcampanaAll

No solicita argumentos,devuelve todas las campañas y cantidad de clientes relacionados a ella.

4.Efectividad de cada campaña a nivel de postulante y alumno.

```
Alter Procedure efectividadCampana @idCampana int as
declare @matriculados float
declare @interesadoPostulante float
declare @Porcentaje float
if Exists (Select id From tbl_campana Where id = @idCampana)
begin
    Set @matriculados=(Select Count(id_cliente) From tbl_detalleCampana Where
id_campana=@idCampana and categoriaCliente=3)
    Set @interesadoPostulante=(Select Count(id_cliente) From tbl_detalleCampana Where
id_campana=@idCampana and (categoriaCliente=1 or categoriaCliente=2))
    if(@interesadoPostulante>0)
        Begin
            set @Porcentaje = (@matriculados/@interesadoPostulante)*100
            Print 'Efectividad de Campaña es :' + Cast(@porcentaje as varchar(10)) + '%'
        End
    Else
        Begin
            Print 'No existen Interesados o Postulantes'
        End
end
Else
begin
    Print 'Campaña no existe'
end
```

Ejemplo de Ejecución

Execute efectividadCampana 1

Requiere Id de campaña y devuelve porcentaje de efectividad, basado en la cantidad de matriculados dividido por postulantes o interesados.

5.Mantener histórico de campañas

Debe mostrar cantidad de interesados, postulantes y matriculados (con su respectivo porcentaje)

Alter Procedure efectividadHistorico as

```
declare @idcampana          int
declare @campana            varchar(30)
```

```
declare @matriculados      float
declare @interesado        float
declare @postulante        float
declare @totalMIP          float
```

```
declare c1 cursor for
Select id From tbl_campana
```

```
open c1
Fetch c1 Into @idcampana
```

```
While(@@FETCH_STATUS=0)
```

```
Begin
```

```
Set @matriculados=(Select Count(id_cliente) From tbl_detalleCampana Where id_campana=@idcampana and
categoriaCliente=3)
```

```
Set @interesado=(Select Count(id_cliente) From tbl_detalleCampana Where id_campana=@idcampana and
categoriaCliente=1)
```

```
Set @postulante=(Select Count(id_cliente) From tbl_detalleCampana Where id_campana=@idcampana and
categoriaCliente=2)
```

```
Set @campana=(Select nombre From tbl_campana Where id= @idcampana)
```

```
set @totalMIP=@matriculados+@interesado+@postulante
```

```
Print 'Campaña                :'+@campana
```

```
print'_____'
```

```
Print 'Matriculados          :'+Cast(@matriculados as varchar(30))+'
```

Porcentaje

```
:'+Cast(((@matriculados/@totalMIP)*100)as varchar(30))+''%
```

```
Print 'Interesados          :'+Cast(@interesado as varchar(30))+'
```

Porcentaje

```
:'+Cast(((@interesado/@totalMIP)*100)as varchar(30))+''%
```

```
Print 'Postulantes          :'+Cast(@postulante as varchar(30))+'
```

Porcentaje

```
:'+Cast(((@postulante/@totalMIP)*100)as varchar(30))+''%
```

```
Print 'total                :'+Cast(@totalMIP as varchar(30))
```

```
print"
```

```
Fetch c1 Into @idcampana
```

```
end
```

```
Close c1
```

```
Deallocate c1
```

Ejemplo de Ejecución :

Execute efectividadHistorico

No requiere argumentos, devuelve cantidad de postulantes, matriculados e interesados con su respectivo porcentaje.

6.Mantener histórico de postulaciones.

Create Procedure postHistorico as

```
declare @xml xml
declare @idcliente          varchar(12)
declare @campaña            varchar(30)
declare @encuestador        varchar(30)
declare @carrera            varchar(30)
declare @disponibilidad     varchar(30)
declare @fecha              varchar(30)
declare @comentario         varchar(1000)
declare @nombcliente        varchar(100)
declare @apecliente         varchar(100)
```

declare c1 cursor for

```
Select id_cliente,tbl_campana.nombre as 'Campaña',tbl_encuestador.nombre as
'Encuestador',tbl_carrera.nombre as 'Carrera',
tbl_clienteContacto.descripcion 'Disponibilidad',fechaContacto,comentario
From tbl_detalleCampana,tbl_campana,tbl_encuestador,tbl_carrera,tbl_clienteContacto
where tbl_campana.id=tbl_detalleCampana.id_campana
and tbl_encuestador.id=tbl_detalleCampana.id_encuestador
and tbl_carrera.id=tbl_detalleCampana.carreraInteres
and tbl_clienteContacto.id=tbl_detalleCampana.estadoContactoCliente
and categoriaCliente = 2
open c1
```

Fetch c1 Into @idcliente,@campaña, @encuestador,@carrera,@disponibilidad,@fecha,@comentario

While(@@FETCH_STATUS=0)

Begin

```
set @xml=(Select datosPersonales From tbl_cliente where rut = @idcliente)
set @nombcliente= (SELECT x.Rec.query('/nombre').value('.', 'nvarchar(2000)') AS 'nombre'
FROM @xml.nodes('/Cliente') as x(Rec))
set @apecliente= (SELECT x.Rec.query('/apellido').value('.', 'nvarchar(2000)') AS 'apellido'
FROM @xml.nodes('/Cliente') as x(Rec))
```

```
print 'Campaña                :'+@campaña
print 'Rut                   :'+@idcliente
print 'Cliente               :'+@nombcliente+' '+@apecliente
print 'Encuestador           :'+@encuestador
print 'Carrera               :'+@carrera
print 'Disponibilidad        :'+@disponibilidad
print 'Fecha                 :'+@fecha
print 'Comentario            :'+@comentario
print '_____'
```

Fetch c1 Into @idcliente,@campaña, @encuestador,@carrera,@disponibilidad,@fecha,@comentario

end

Close c1

Deallocate c1

Ejemplo de ejecucion:

Execute postHistorico

No requiere argumentos,devuelve Listado de postulantes a campañas.

7. Conocer la historia completa de un estudiante

Desde el interés hasta la cantidad de productos y servicios que ha tenido con la institución. Una visión completa del cliente.

Es un Procedimiento compuesto de más procedimientos, a continuación se detallan.

Datos Cliente

```
Alter Procedure mostrarCliente @idcliente varchar(12) as
Declare @xml xml
--fetch cliente
declare @nombre          varchar(30)
declare @apellido        varchar(30)
declare @comuna          varchar(30)
declare @ciudad          varchar(30)
declare @region          varchar(30)
--fetch telefonos
declare @telefono        varchar(30)
if Exists(Select rut From tbl_cliente Where rut=@idcliente)
Begin
set @xml=(Select datosPersonales From tbl_cliente where rut = @idcliente)
declare c1 cursor for
SELECT
x.Rec.query('/nombre').value('.', 'nvarchar(2000)') AS 'nombre',
x.Rec.query('/apellido').value('.', 'nvarchar(2000)') AS 'apellido',
x.Rec.query('/comuna').value('.', 'nvarchar(2000)') AS 'comuna',
x.Rec.query('/ciudad').value('.', 'nvarchar(2000)') AS 'ciudad',
x.Rec.query('/region').value('.', 'nvarchar(2000)') AS 'region'
FROM @xml.nodes('/Cliente') as x(Rec)

declare c2 cursor for
SELECT
x.Rec.query('/').value('.', 'nvarchar(2000)') AS 'telefono'
FROM @xml.nodes('/Cliente/Telefonos/telefono') as x(Rec)

declare @contador int
open c1
Fetch c1 Into @nombre,@apellido,@comuna,@ciudad,@region
print '_____ '
print 'Cliente'
print '_____ '
While(@@FETCH_STATUS=0)
Begin
print 'nombre          :'+@nombre
print 'apellido        :'+@apellido
print 'comuna          :'+@comuna
print 'ciudad          :'+@ciudad
print 'region          :'+@region
```

```
--Telefonos
print '_____ '
print 'Telefonos'
print '_____ '

    open c2
    Fetch c2 Into @telefono
    set @contador=1
    While(@@FETCH_STATUS=0)
    Begin
    print 'telefono '+Cast(@contador as varchar(10))+ ' :'+@telefono
    Fetch c2 Into @telefono
    set @contador=@contador+1
    end
    close c2
    deallocate c2

Fetch c1 Into @nombre,@apellido,@comuna,@ciudad,@region
end
close c1
deallocate c1
End
Else
Begin
Print 'Cliente no Registrado'
End
```

Ejemplo de ejecución:

```
execute mostrarCliente '1-1'
```

requiere rut de cliente, devuelve datos personales desde el xml.

Intereses Cliente

```
Alter Procedure InteresesCliente @idcliente varchar(12) as
declare @contador          int
--fetch intereses
declare @id                int
declare @fechal            varchar(30)
declare @fechaT            varchar(30)
declare @interes varchar(300)
declare @tipo              varchar(30)

if Exists(Select rut From tbl_cliente Where rut=@idcliente)
Begin

declare c3 cursor for
Select tbl_interes.id,tbl_interes.nombre,tbl_tipoInteres.nombre,tbl_interes.fechaInicio,tbl_interes.fechaTermino
```

```
From tbl_interes,tbl_cliente_inter,tbl_tipoInteres
Where tbl_interes.id=tbl_cliente_inter.id_interes
and tbl_interes.tipo=tbl_tipoInteres.id
and tbl_cliente_inter.id_cliente=@idcliente
```

```
--intereses
open c3
Fetch c3 Into @id,@interes,@tipo,@fechal,@fechaT
set @contador=1
print '_____ '
print 'INTERESES'
print '_____ '
While(@@FETCH_STATUS=0)
Begin
print 'ID                               :'+Cast(@id as varchar(10))
print 'Interes                        :'+Cast(@interes as varchar(300))
print 'Fecha Inicio                    :'+Cast(@fechal as varchar(10))
print 'Fecha Termino                  :'+Cast(@fechaT as varchar(10))
print 'Tipo                             :'+Cast(@tipo as varchar(10))
print '_____ '
Fetch c3 Into @id,@interes,@tipo,@fechal,@fechaT
set @contador=@contador+1
end
close c3
deallocate c3
End

Else
Begin
Print 'Cliente no Registrado'
End
```

Ejemplo de ejecución:

```
execute InteresesCliente '1-1'
```

Requiere rut cliente,devuelve listado de intereses relacionados a el.

Historial campañas

```
Alter Procedure historial @idcliente varchar(12) as
declare @contador int
--historial
declare @idcl varchar(30)
declare @camp varchar(30)
declare @cat varchar(30)
declare @contact varchar(30)
declare @carre varchar(30)
declare @estaContact varchar(30)
declare @fechaContact varchar(30)
declare @comentar varchar(30)
```

```
if Exists(Select rut From tbl_cliente Where rut=@idcliente)
Begin
```

```
declare c4 cursor for
Select id_cliente,tbl_campana.nombre,tbl_categoriaCliente.nombre,tbl_encuestador.nombre,tbl_carrera.nombre,
tbl_clienteContacto.descripcion,fechaContacto,comentario
From tbl_detalleCampana,tbl_campana,tbl_encuestador,tbl_carrera,tbl_clienteContacto,tbl_categoriaCliente
where tbl_campana.id=tbl_detalleCampana.id_campana
and tbl_encuestador.id=tbl_detalleCampana.id_encuestador
and tbl_carrera.id=tbl_detalleCampana.carreraInteres
and tbl_clienteContacto.id=tbl_detalleCampana.estadoContactoCliente
and tbl_categoriaCliente.id=tbl_detalleCampana.categoriaCliente
and id_cliente = @idcliente
```

```
--Historico campañas
open c4
Fetch c4 Into @idcl,@camp,@cat,@contact,@carre,@estaContact,@fechaContact,@comentar
set @contador=1
print '_____ '
print 'Historial Campañas'
print '_____ '
While(@@FETCH_STATUS=0)
Begin
print 'Campaña' :'+Cast(@camp as
varchar(100))
print 'Categoria' :'+Cast(@cat as varchar(100))
print 'Contactador' :'+Cast(@contact as varchar(100))
print 'Carrera' :'+Cast(@carre as varchar(100))
print 'Estado Contacto' :'+Cast(@estaContact as varchar(100))
print 'Fecha Contacto' :'+Cast(@fechaContact as varchar(100))
print 'Comentario' :'+Cast(@comentar as
varchar(100))
print '_____ '
Fetch c4 Into @idcl,@camp,@cat,@contact,@carre,@estaContact,@fechaContact,@comentar
set @contador=@contador+1
end
close c4
deallocate c4

End
Else
Begin
Print 'Cliente no Registrado'
End
```

Ejemplo de ejecución:

```
execute historial '1-1'
```

Requiere rut, devuelve listado de postulaciones matrículas etc. referente al cliente y todas las campañas en que haya participado.

Estadísticas Cliente

```
Alter Procedure estadisticaCliente @idclient varchar(12) as
declare @interesad float
declare @postul float
declare @matric float
declare @titul float
declare @ex float
declare @tot float
declare @porce float
if Exists(Select rut From tbl_cliente Where rut=@idclient)
Begin
--interesados
set @interesad = (Select count(id_cliente) From tbl_detalleCampana where categoriaCliente=1
and id_cliente=@idclient)
--postulantes
set @postul = (Select count(id_cliente) From tbl_detalleCampana where
categoriaCliente=2 and id_cliente=@idclient)
--matriculados
set @matric = (Select count(id_cliente) From tbl_detalleCampana where
categoriaCliente=3 and id_cliente=@idclient)
--titulados
set @titul = (Select count(id_cliente) From tbl_detalleCampana where
categoriaCliente=4 and id_cliente=@idclient)
--Ex Alumnos
set @ex = (Select count(id_cliente) From tbl_detalleCampana where
categoriaCliente=5 and id_cliente=@idclient)
set @tot = @interesad+@postul+@matric+@titul+@ex
print '_____ '
print 'Estadísticas Totales'
print '_____ '
print 'Interesado:'+Cast(@interesad as varchar(30))+ ' veces '+Cast(((@interesad/@tot)*100)as
varchar(30))+ '%'
print 'Postulado :'+Cast(@postulas varchar(30))+ ' veces '+Cast(((@postul/@tot)*100)as
varchar(30))+ '%'
print 'Matriculado :'+Cast(@matricas varchar(30))+ ' veces '+Cast(((@matric/@tot)*100)as
varchar(30))+ '%'
print 'Titulado :'+Cast(@titul as varchar(30))+ ' veces '+Cast(((@titul/@tot)*100)as varchar(30))+ '%'
print 'Ex Alumno :'+Cast(@ex as varchar(30))+ ' veces '+Cast(((@ex/@tot)*100)as
varchar(30))+ '%'
print 'Total :'+Cast(@tot as varchar(30))+ ' veces '
End
Else
Begin
Print 'Cliente no Registrado'
End
```

Ejemplo de ejecución

Execute estadisticaCliente '1-1' requiere rut de cliente, devuelve cantidad de interacciones con las campañas y su respectivo porcentaje

Perfil Cliente

Finalmente el método que reúne todos los anteriores.

```
Alter Procedure perfilCliente @idclient varchar(12) as
if Exists(Select rut From tbl_cliente Where rut=@idclient)
Begin
Execute mostrarCliente @idclient
Execute InteresesCliente @idclient
Execute historial @idclient
Execute estadisticaCliente @idclient
End
Else
Begin
Print 'Cliente no Registrado'
End
```

execute perfilCliente '1-1'.

Requiere rut de cliente, devuelve datos personales, intereses, historial de campañas y estadísticas.

8. Acciones que se han realizado con el cliente

```
Create Procedure seguimientoCliente @idcliente varchar(12) as
IF Exists(Select rut From tbl_cliente Where rut=@idcliente)
Begin
Select id_cliente as 'Rut cliente',tbl_campana.nombre as 'Campaña',tbl_categoriaCliente.nombre as
'Categoria',tbl_encuestador.nombre as 'Encuestador',tbl_tipoEncuestador.nombre as 'Medio de Contacto',
tbl_carrera.nombre as 'Carrera' ,tbl_clienteContacto.descripcion as 'Disponibilidad',fechaContacto as 'Fecha
Contacto',comentario as 'Comentario Adicional'
From
tbl_detalleCampana,tbl_campana,tbl_encuestador,tbl_carrera,tbl_clienteContacto,tbl_categoriaCliente,tbl_tipoEn
cuestador
where tbl_campana.id=tbl_detalleCampana.id_campana
and tbl_encuestador.id=tbl_detalleCampana.id_encuestador
and tbl_carrera.id=tbl_detalleCampana.carreraInteres
and tbl_clienteContacto.id=tbl_detalleCampana.estadoContactoCliente
and tbl_categoriaCliente.id=tbl_detalleCampana.categoriaCliente
and tbl_tipoEncuestador.id=tbl_encuestador.tipo
and id_cliente = @idcliente
End
Else
Begin
Print 'Cliente no Registrado'
End
```

Ejemplo de Ejecucion:

Execute seguimientoCliente '1-1'

Requiere rut cliente, devuelve acciones que ha realizado en campañas y sus interacciones.

9. Mostrar efectividad una vez titulados

Cuantos clientes se titularon y luego tomaron programas tales como cursos, diplomados, especializaciones, etc) y conocer sus tasas de efectividad.

```
Alter Procedure efectividadTitulados as
declare @rut varchar(12)
declare @fechaC varchar(50)
declare @cantidad float
declare @acumulado float
declare @titulados float
set @titulados = (Select Count(id_cliente)
From tbl_detalleCampana,tbl_categoriaCliente
where tbl_categoriaCliente.id=tbl_detalleCampana.categoriaCliente
and tbl_categoriaCliente.nombre='Titulado')

declare c1 cursor for
Select id_cliente as 'Rut cliente',fechaContacto as 'Fecha Contacto'
From tbl_detalleCampana,tbl_categoriaCliente
where tbl_categoriaCliente.id=tbl_detalleCampana.categoriaCliente
and tbl_categoriaCliente.nombre='Titulado'
open c1
Fetch c1 Into @rut,@fechaC
While(@@FETCH_STATUS=0)
Begin
    declare c2 cursor for
    Select COUNT(id_cliente) From tbl_detalleCampana where id_cliente = @rut and
fechaContacto>@fechaC and categoriaCliente=3
    open c2
    set @acumulado=0

    Fetch c2 Into @cantidad
    While(@@FETCH_STATUS=0)
    Begin
        set @acumulado = @acumulado+@cantidad
        Fetch c2 Into @cantidad
    end
    close c2
    deallocate c2
Fetch c1 Into @rut,@fechaC
End
close c1
deallocate c1
print 'Total Titulados :'+Cast(@titulados as varchar(50))
print 'Total Titulados Reincidentes :'+Cast(@acumulado as varchar(50))
print 'Efectividad Campañas :'+Cast(((@acumulado/@titulados)*100) as varchar(50))+'%'
```

Ejemplo de ejecución:

execute efectividadTitulados

No requiere argumentos, devuelve cantidad de clientes que se han matriculado después de la fecha en que se contactaron y se informaron como titulados.

10. Conocer un listado de interesados que no quieren ser contactados por la institución

Debe registrar por qué no quiere ser contactado, fecha, hora y quien lo contactó.

Alter Procedure interesadosNoContactar as

declare @xml xml

declare @nomCliente varchar(50)

declare @apeCliente varchar(50)

declare @idcliente varchar(12)

declare @campaña varchar(30)

declare @categoria varchar(30)

declare @encuestador varchar(30)

declare @medio varchar(30)

declare @carrera varchar(30)

declare @disponibilidad varchar(30)

declare @fecha varchar(30)

declare @comentario varchar(1000)

declare c1 Cursor for

Select

id_cliente, tbl_campana.nombre, tbl_categoriaCliente.nombre, tbl_encuestador.nombre, tbl_tipoEncuestador.nombre, tbl_carrera.nombre,

tbl_clienteContacto.descripcion, fechaContacto, comentario

From

tbl_detalleCampana, tbl_campana, tbl_encuestador, tbl_carrera, tbl_clienteContacto, tbl_categoriaCliente, tbl_tipoEncuestador

where tbl_campana.id=tbl_detalleCampana.id_campana

and tbl_encuestador.id=tbl_detalleCampana.id_encuestador

and tbl_carrera.id=tbl_detalleCampana.carreraInteres

and tbl_clienteContacto.id=tbl_detalleCampana.estadoContactoCliente

and tbl_categoriaCliente.id=tbl_detalleCampana.categoriaCliente

and tbl_tipoEncuestador.id=tbl_encuestador.tipo

and tbl_categoriaCliente.nombre='Interesado'

and tbl_clienteContacto.descripcion='No Contactar'

open c1

Fetch c1 Into

@idcliente, @campaña, @categoria, @encuestador, @medio, @carrera, @disponibilidad, @fecha, @comentario

While (@@FETCH_STATUS=0)

Begin

set @xml=(Select datosPersonales From tbl_cliente where rut = @idcliente)

set @nomCliente= (SELECT x.Rec.query('/nombre').value('.', 'nvarchar(2000)') AS 'nombre'
FROM @xml.nodes('/Cliente') as x(Rec))

set @apeCliente=(SELECT x.Rec.query('/apellido').value('.', 'nvarchar(2000)') AS 'apellido'
FROM @xml.nodes('/Cliente') as x(Rec))

print 'Campaña :'+@campaña

print 'Rut :'+@idcliente

print 'Cliente :'+@nomCliente+' '+@apeCliente

```
print 'Categoria           :'+@categoria
print 'Encuestador         :'+@encuestador
print 'Tipo contacto       :'+@medio
print 'Carrera             :'+@carrera
print 'Disponibilidad      :'+@disponibilidad
print 'Fecha               :'+@fecha
print 'Comentario          :'+@comentario
print '_____ '
Fetch c1 Into
@idcliente,@campaña,@categoria,@encuestador,@medio,@carrera,@disponibilidad,@fecha,@comentario
end
close c1
deallocate c1
```

Ejemplo de Ejecución:

```
execute interesadosNoContactar
```

No requiere argumentos, devuelve clientes INTERESADOS que no quieren ser contactados y el comentario respectivo.

Datos Relevantes en Tablas

tbl_clienteContacto	
id	
descripcion	

id	descripcion
1	Contactar
2	No Contactar

Hace Referencia a estado del cliente, si después de ser contactado por primera vez desea seguir recibiendo encuestas o información.

tbl_tipoEncuestador	
id	
nombre	


id	nombre
1	Telefonico
2	Correo

Se refiere al medio que utiliza el Encuestador para contactarse con el cliente.

tbl_tipoCampana	
id	
nombre	

id	nombre
1	Pre-Grado
2	Post-Grado

Se refiere al Tipo de Campaña, Pre Grado : donde existe el pre requisito haber Egresado de enseñanza media, Post Grado : donde el pre requisito es haber terminado un grado de estudio superior.

tbl_categoriaCliente	
	id
	nombre

id	nombre
1	Interesado
2	Postulante
3	Matriculado
4	Titulado
5	Ex Alumno

Se refiere a la categoría del estudiante, la cual podrá quedar registrada solo una vez por cliente y campaña.

tbl_tipoInteres	
	id
	nombre

id	nombre
1	Curso
2	Diplomado
3	Especializaciones
4	Taller

Se refiere al tipo de servicio otorgado al cliente.

