

---

# UNIVERSIDAD NACIONAL AUTONOMA DE HONDURAS

---

Facultad de Ingeniería  
Departamento de Ingeniería en Sistemas

## **Documentación del Sistema de reportes MP**

**Asignatura:** Ingeniería de Software

**Catedrático:** Ing. Miguel Saucedo

### **Presentado por:**

<b>Nombre</b>	<b>Número de Cuenta</b>
Edson Froylan Bonilla	20081003488
Erick Alexander Zelaya	20092500066
Emil Daniel Ordoñez	20091001347
Sindy Danely García	20091004217
Osly Salinas	20081000979

**Tegucigalpa M.D.C, 22 de Agosto del 2014**

## Tabla de contenido

Marco Lógico.....	4
Identificación y descripción del problema.....	4
Análisis de involucrados.....	5
Tabla de ponderación de involucrados en el proyecto .....	5
Árbol de Problema.....	6
Árbol de Soluciones.....	7
Estructura Analítica del Proyecto .....	8
Construcción de matriz de marco lógico.....	9
Especificaciones de escenarios de uso .....	10
Caso de Uso 01: Ingreso al sistema.....	10
Caso de Uso 02: Registro de usuarios.....	10
Caso de Uso 03: Generación reportes de presupuesto .....	11
Reportes.....	12
Reporte empleados con cuenta presupuestaria individual .....	12
Reporte de Ingresos de empleados .....	18
Reporte de Deducciones de empleados .....	19
Reporte Presupuestario por Programa .....	20
Reporte Presupuestario por Grupo .....	21
Reporte Presupuestario por Subgrupo .....	22
Reporte Comparación Presupuesto Original.....	23
Reporte Transacciones .....	24
Diccionario de Datos.....	25
Base de Datos: mpsiafireportes.....	25
Nombre Tabla: tbl_rol .....	25
Nombre Tabla: tbl_rol_usuario .....	25
Nombre Tabla: tbl_usuario .....	26
Diagrama Relacional .....	28
Base de datos: mpsiafi .....	28
Nombre Tabla: bgadjm .....	28
Nombre Tabla: bgtrfm .....	29
Nombre Tabla: prempg .....	29
Nombre Tabla: HRJobs .....	30

Nombre Tabla: apchck .....	30
Nombre Tabla: bgbudp .....	30
Nombre Tabla: bgbudt .....	31
Nombre Tabla: prmisd .....	31
Nombre Tabla: prmisc.....	32
Nombre Tabla: prdept.....	33
Nombre Tabla: prempy .....	33
Diagrama Relacional .....	36

# Marco Lógico

## Identificación y descripción del problema

El ministerio público, ente independiente de los poderes del estado, posee un sistema ERP capaz de manejar y administrar todos los procesos y recursos administrativos de la empresa, siendo de vital importancia la administración correcta y control del presupuesto. Actualmente el ERP es capaz de generar los reportes necesarios para la toma de decisiones gerenciales, el problema que presenta el sistema es esencialmente en la eficiencia del mismo para generar los reportes, ya que para obtener el presupuesto a una determinada fecha este se debe realizar ejecutando varias consultas al sistema por parte de las personas encargadas, lo que conlleva la inversión de más tiempo y recursos en la generación de un solo reporte.

Es importante que el sistema refleje la situación económica real de la institución debido a que en ocasiones es necesario hacer uso de todos los fondos con los que se cuenta en un determinado momento para hacer frente a una situación de carácter urgente, por lo que, conocer con exactitud de cuánto dinero se dispone se vuelve algo indispensable.

Se desea disminuir el tiempo y el esfuerzo invertido en la elaboración de reportes que sirvan a la administración de la institución, además es importante que se cuente con presupuestos y estados de cuentas reales en cualquier momento para tener un conocimiento con bases sólidas que permita efectuar tomas de decisiones gerenciales rápidas en situaciones que así lo ameriten.

## Análisis de involucrados



## Tabla de ponderación de involucrados en el proyecto

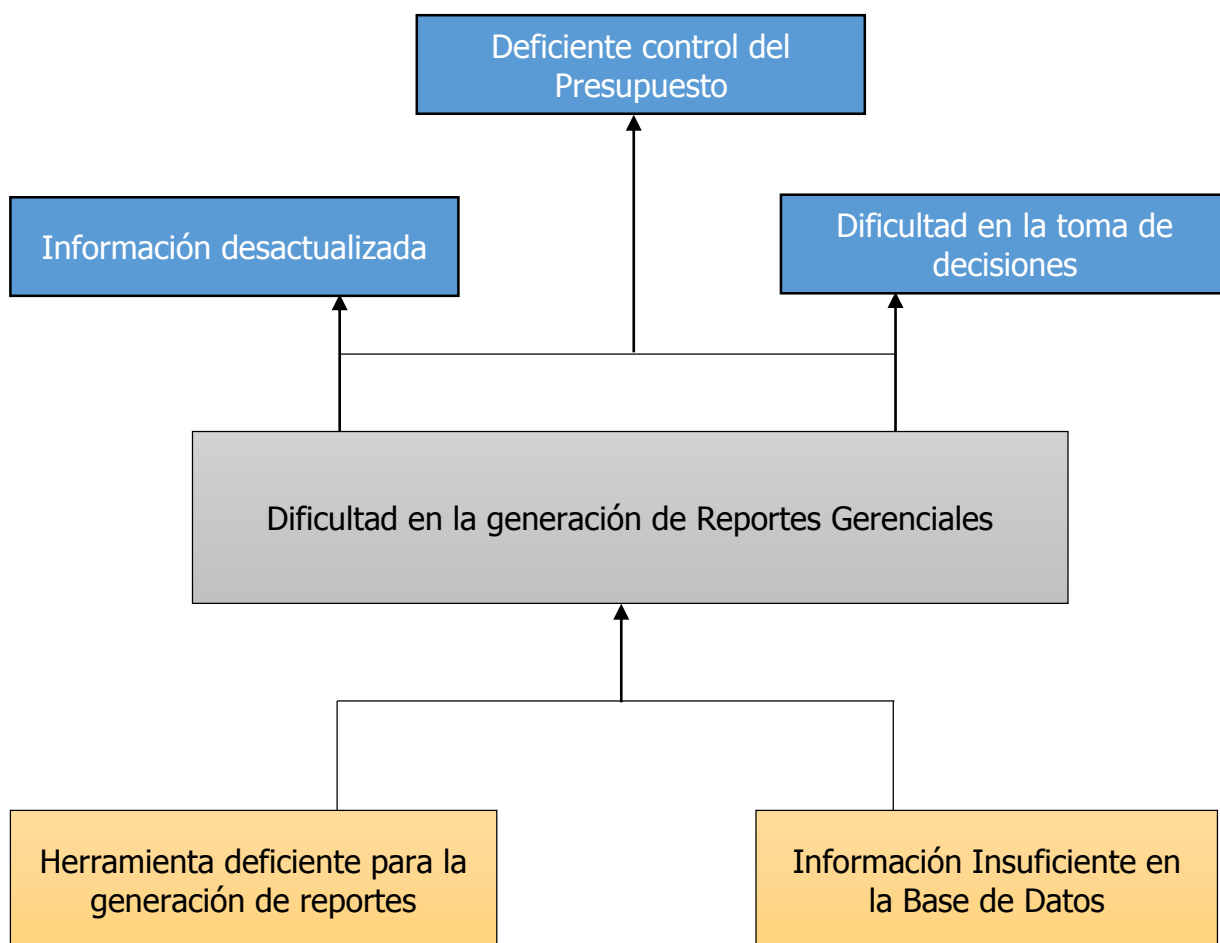
Con lo que por experiencia se conoce y basado en entrevistas que se han realizado al personal que está involucrado directa e indirectamente en el proyecto a realizar; se presenta la siguiente tabla, en la cual se pondera el apoyo que cada uno de estos pueda brindar.

La columna “expectativa” indica el interés que pueda tener la entidad para poder brindar ayuda al proyecto. La columna “fuerza” indica la influencia que pueda tener la entidad para ayudar a resolver problemas. La columna “resultado” muestra el producto de las dos columnas anteriormente mencionadas para poder ponderar que entidad tiene un mayor aporte al proyecto.

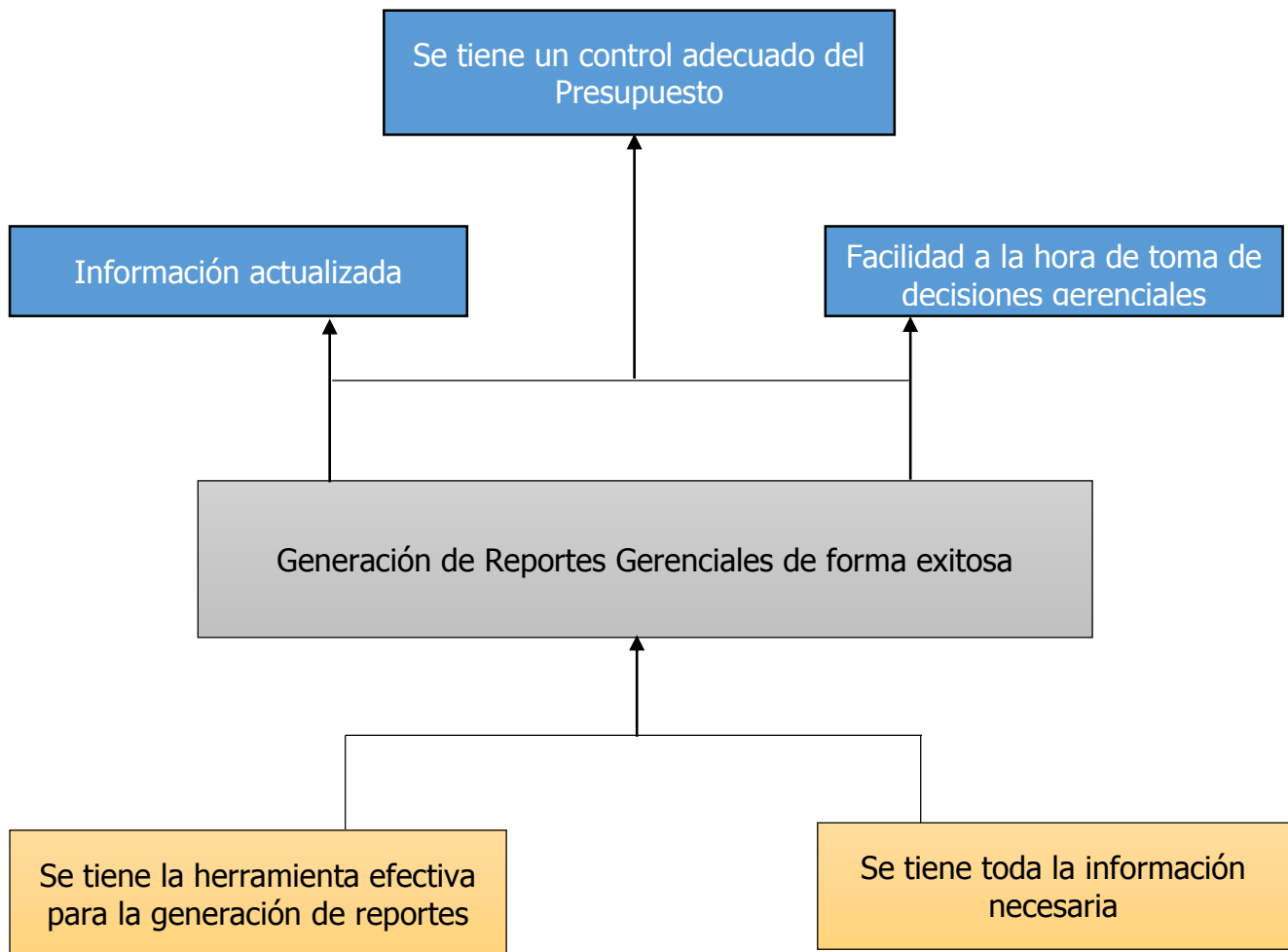
Se utiliza una escala de 1 a 5, donde 1 es el nivel de impacto más bajo y 5 es el más alto.

Involucrados	Expectativa	Fuerza	Resultado
Gerente Administrativo	4	4	16
Sub-gerente de Presupuesto	4	3	12
Sub-gerente de RR-HH	4	3	12
Sub-gerente de Recursos Materiales	4	3	12
Director de TI	5	3	15
Secretario de Estado	2	5	10

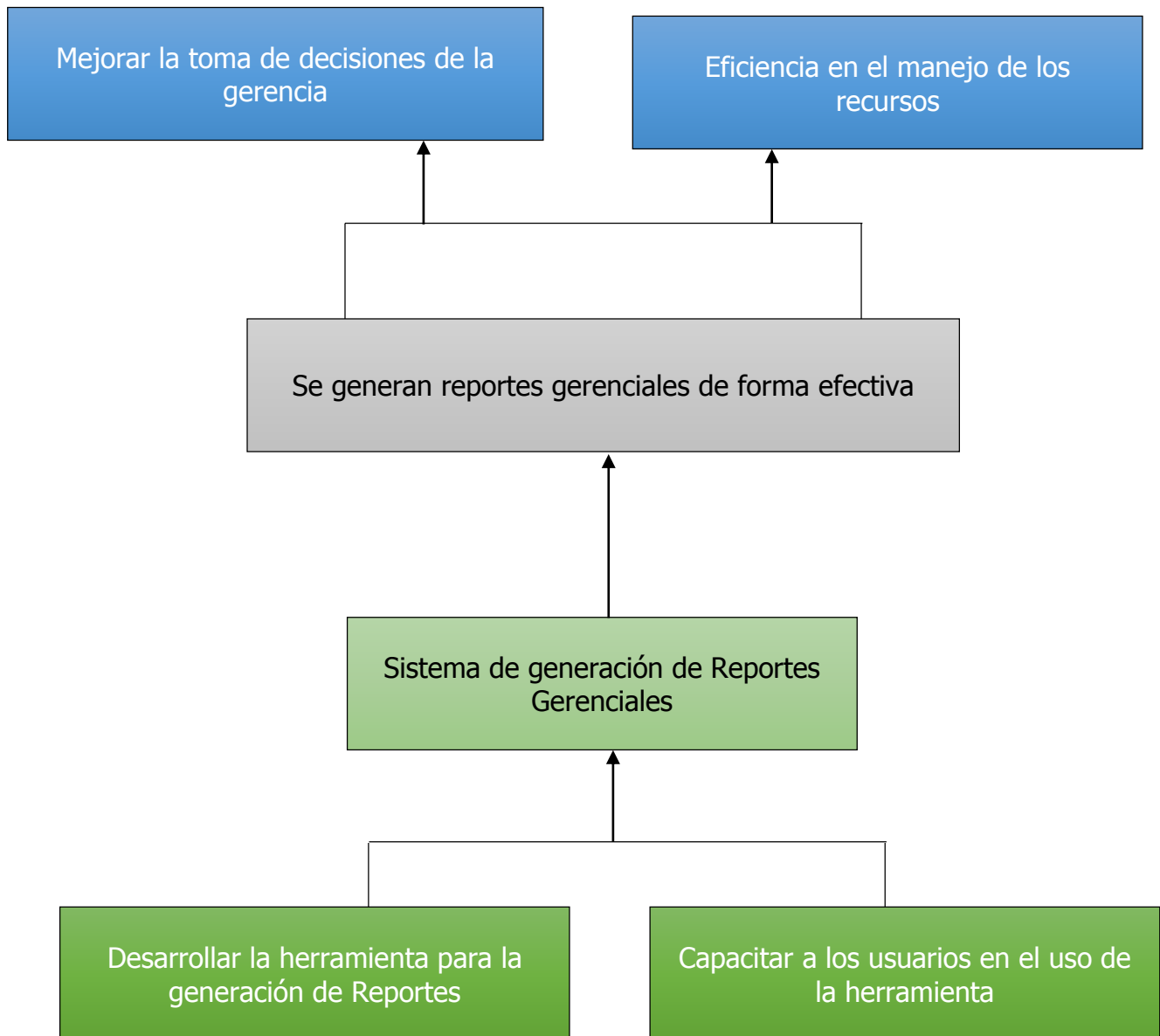
## Árbol de Problema



# Árbol de Soluciones



# Estructura Analítica del Proyecto





## Construcción de matriz de marco lógico

Nivel	Enunciado de nivel	Indicadores		Modo de verificación	Supuestos
		Enunciado	Calculo		
<b>FIN</b>	F.1: Contribuir a la toma de decisiones de la gerencia  F.2: Contribuir a la eficiencia en el manejo de los recursos	El consumo se acerca en un 90% a lo presupuestado.	Consumo $\geq$ $0.9 \times$ Presupuesto y Consumo $\leq$ Presupuesto	Reportes anuales de presupuestos (Gastos, etc.)	Se tiene un registro de todo lo que se ha comprado o todo en lo que se ha invertido el presupuesto.
<b>PROPOSITO</b>	Se generan reportes gerenciales de forma efectiva	El 90% de los reportes que necesita la gerencia se pueden generar de forma exitosa	(Reportes generados/ total generados solicitados por gerencia) $\geq 0.9$	Registro de reportes emitidos por el sistema	Lleva un control y copia de los reportes que se han emitido
<b>COMPONENTES</b>	C.1 La herramienta de generación de reportes está disponible la mayoría del tiempo	C.1 El 90% de las ocasiones se pueden generar los reportes sin contratiempos (errores de la herramienta).	C.1).- (Reportes generados/ total generados solicitados por gerencia) $\geq 0.9$	Registros de reportes emitidos	C.1).- se lleva un control de fallas presentadas y de los reportes generados y no generados.
<b>ACTIVIDADES</b>	A.1.1 Desarrollar una herramienta para generación de reportes.  A.1.2 Capacitar a los usuarios en el uso de la herramienta	A.1.1).- Lps. 0.00  A.1.2).- Lps. 2500.00	A.1.1).- Limite: Lps. 0.00  A.1.2).- Limite: Lps. 2500.00	Presupuesto de costos	A1.1)Se cuenta con presupuesto para el desarrollo de la herramienta  A1.2) Se cuenta con presupuesto para capacitaciones

# Especificaciones de escenarios de uso

## Caso de Uso 01: Ingreso al sistema

**Actores:** usuarios del sistemas.

**Precondición:**

- El usuario y la contraseña fueron ingresados correctamente.

**Post-condición:**

- Se muestra una nueva página con diferentes opciones que dependerá de los privilegios otorgados al usuario que ha ingresado.

**Flujo Básico:**

El usuario introduce su nombre de usuario y su contraseña. Se realiza la conexión a la base de datos para verificar que los datos ingresados por el usuario existen y pertenecen al usuario que desea ingresar de ser así se permite el ingreso del usuario al sistema de generación de reportes mostrándole una nueva página con diferentes opciones que dependerá de los privilegios otorgados al usuario que ha ingresado.

**Excepción 1:** Ingreso no exitoso al sistema

- El usuario intenta ingresar al sistema con su nombre de usuario y contraseña pero el usuario no fue encontrado en los registros de la bases de datos.
- Se le muestra al usuario un mensaje informando que los valores ingresados no existen en el sistema.
- El administrador del sistema deberá crear una cuenta para el usuario si este está autorizado para tener acceso al sistema.

**Excepción 2:** Ingreso no exitoso al sistema por inconsistencia entre usuario y contraseña.

- El usuario ha sido encontrado en los registros de las bases de datos, pero la contraseña ingresada no coincide o viceversa.
- Se le debe informar al usuario de la inconsistencia.
- El usuario deberá ingresar sus datos de nuevo correctamente.

## Caso de Uso 02: Registro de usuarios

**Actores:** administrador del sistemas.

**Precondición:**

- El usuario no está registrado en el sistema.
- El administrador debe recibir una solicitud de registro del usuario.

**Post-condición:**

- Se cree la cuenta de usuario en el sistema.

**Flujo Básico:**

El administrador debe verificar que el usuario no existe en el sistema, si es el caso deberá ingresar todos los datos del empleado necesarios para registrarlo.

**Excepción 1:** Conexión fallida a la base de datos

**Contexto:** El usuario no posee una cuenta en el sistema para generar reportes

## Caso de Uso 03: Generación reportes de presupuesto

**Actores:** usuarios del sistema

**Precondiciones:**

- El usuario ingreso correctamente al sistema de reportes
- El usuario ingreso correctamente los datos para los filtros utilizados para generar los reportes.

**Flujo Básico:**

- El usuario seleccionara el tipo de reporte que desea generar, especificando los campos que necesita para generar un reporte específico.
- Se muestran los resultados para el reporte solicitado.
- Se exportan los resultados para el reporte en formato Excel.

**Post-condiciones:**

- Se generan correctamente los reportes deseados.

**Excepciones:** No se muestran los resultados debido a que no existen resultados para los filtros.

**Asunciones:**

- No se encuentran registros a partir de los filtros ingresados.

**Resultados:**

- Se muestra un mensaje informando que se deben intentar otros filtros para buscar registros.

**Contexto:** El usuario desea generar algún tipo de reporte.

# Reportes

<b>Nombre del Reporte:</b>	<b>Reporte empleados con cuenta presupuestaria individual</b>		
<b>Descripción General del Reporte:</b>			
<b>Campos</b>	<b>Tablas</b>	<b>Descripción</b>	<b>Notas</b>
Código	prempy.cempno		
Identidad	Prempy.cfedid		
Nombres	Prempy.cfname		
Apellidos	Prempy.clname		
Cargos	HRJobs.cdsc		
Código depto	Prempy.cdeptno		
Depto	Prdept.cdeptname		
Sueldo	Prempy.nmonthpay		
Fecha contrato	Prempy.dcntrct		
Acuerdo	Prempy.dhire		
fechaNacimiento	Prempy.dbirth		
ctaxstate	Prempy.ctaxstate		
Tipo de Planilla	Prempy.cpaytype		
Código presupuestaria	Prempg.cwageacc		
Estado	Prempy.cstate		
Sexo	Prempy.csex		
<b>Filtro del Reporte:</b>			
<b>Campo</b>	<b>Criterio</b>	<b>Descripción</b>	<b>Notas</b>
Cargo			
Depto			
Código Presupuestario			
Estado			
Fecha contrato y acuerdo			

## Procedimientos Almacenados:

```
create procedure [dbo].[sp_slFContrato]
    @f_inicio datetime, @f_fin datetime
AS
begin

    set nocount on;
    Select emp.cempno, cfedid, cfname, clname, csex, dbirth,
    emp.cdeptno, cdeptname, jobs.cDesc, nmonthpay, dcntrect, dhire,
    ctaxstate, emp.cpaytype, cstate, emp.cwageacc
    from mpsiafi.dbo.prempy emp, mpsiafi.dbo.HRJobs jobs,
        mpsiafi.dbo.prdept dept, mpsiafi.dbo.prempg empg
    where emp.cdeptno = dept.cdeptno
        and emp.cjobtitle = jobs.cJobTitlNO
        and emp.cempno = empg.cempno
        and dcntrect between isnull(@f_inicio,'19000101') and isnull(@f_fin,
'29001231')
end
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
create procedure [dbo].[sp_slFAcuerdo]
    @f_inicio datetime, @f_fin datetime
AS
begin

    set nocount on;
    Select emp.cempno, cfedid, cfname, clname, csex, dbirth,
    emp.cdeptno, cdeptname, jobs.cDesc, nmonthpay, dcntrect, dhire,
    ctaxstate, emp.cpaytype, cstate, emp.cwageacc
    from mpsiafi.dbo.prempy emp, mpsiafi.dbo.HRJobs jobs,
        mpsiafi.dbo.prdept dept, mpsiafi.dbo.prempg empg
    where emp.cdeptno = dept.cdeptno
        and emp.cjobtitle = jobs.cJobTitlNO
        and emp.cempno = empg.cempno
        and dhire between isnull(@f_inicio,'19000101') and isnull(@f_fin,
'29001231')
end
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```

CREATE procedure [dbo].[sp_slCargo]
    @cargo varchar(40)
AS
begin
    Select mpsiafi.dbo.prempy.cempno, cfedid, cfname, clname, csex, dbirth,
    mpsiafi.dbo.prempy.cdeptno, cdeptname, cDesc, nmonthpay, dcntrect, dhire,
    ctaxstate, mpsiafi.dbo.prempy.cpaytype, cstate, mpsiafi.dbo.prempg.cwageacc
    from mpsiafi.dbo.prempy, mpsiafi.dbo.HRJobs,
        mpsiafi.dbo.prdept, mpsiafi.dbo.prempg
    where mpsiafi.dbo.prempy.cdeptno = mpsiafi.dbo.prdept.cdeptno
        and mpsiafi.dbo.prempy.cjobtitle = mpsiafi.dbo.HRJobs.cJobTitlNO
        and mpsiafi.dbo.prempy.cempno = mpsiafi.dbo.prempg.cempno
        and cDesc LIKE '%' + @cargo + '%';

end
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

create procedure [dbo].[sp_slDpto]
    @dpto char(30)
AS
begin

    Select empy.cempno, cfedid, cfname, clname, csex, dbirth,
    empy.cdeptno, cdeptname, jobs.cDesc, nmonthpay, dcntrect, dhire,
    ctaxstate, empy.cpaytype, cstate, empg.cwageacc
    from mpsiafi.dbo.prempy empy, mpsiafi.dbo.HRJobs jobs,
        mpsiafi.dbo.prdept dept, mpsiafi.dbo.prempg empg
    where empy.cdeptno = dept.cdeptno
        and empy.cjobtitle = jobs.cJobTitlNO
        and empy.cempno = empg.cempno
        and cdeptname = @dpto

end
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

create procedure [dbo].[sp_slCodPresupuestario]
    @codP char(30)
AS
begin

    set nocount on;
    Select empy.cempno, cfedid, cfname, clname, csex, dbirth,
    empy.cdeptno, cdeptname, jobs.cDesc, nmonthpay, dcntrect, dhire,
    ctaxstate, empy.cpaytype, cstate, empq.cwageacc
    from mpsiafi.dbo.prempy empy, mpsiafi.dbo.HRJobs jobs,
        mpsiafi.dbo.prdept dept, mpsiafi.dbo.prempq empq
    where empy.cdeptno = dept.cdeptno
        and empy.cjobtitle = jobs.cJobTitlNO
        and empy.cempno = empq.cempno
        and empq.cwageacc = @codP

end
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

create procedure [dbo].[sp_slEstado]
    @estado char(2)
AS
begin

    set nocount on;
    Select empy.cempno, cfedid, cfname, clname, csex, dbirth,
    empy.cdeptno, cdeptname, jobs.cDesc, nmonthpay, dcntrect, dhire,
    ctaxstate, empy.cpaytype, cstate, empq.cwageacc
    from mpsiafi.dbo.prempy empy, mpsiafi.dbo.HRJobs jobs,
        mpsiafi.dbo.prdept dept, mpsiafi.dbo.prempq empq
    where empy.cdeptno = dept.cdeptno
        and empy.cjobtitle = jobs.cJobTitlNO
        and empy.cempno = empq.cempno
        and cstate = @estado

end
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

create procedure [dbo].[sp_slCargo_Dpto]
    @cargo char(40), @dpto char(30)
AS
begin

```

```

    set nocount on;
    Select empy.cempno, cfedid, cfname, clname, csex, dbirth,
    empy.cdeptno, cdeptname, jobs.cDesc, nmonthpay, dcntrect, dhire,
    ctaxstate, empy.cpaytype, cstate, empg.cwageacc
    from mpsiafi.dbo.prempy empy, mpsiafi.dbo.HRJobs jobs,
        mpsiafi.dbo.prdept dept, mpsiafi.dbo.prempg empg
    where empy.cdeptno = dept.cdeptno
        and empy.cjobtitle = jobs.cJobTitlNO
        and empy.cempno = empg.cempno
        and jobs.cDesc = @cargo and cdeptname = @dpto

```

```

end
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

create procedure [dbo].[sp_slAcuerdo_Contrato]
    @aIn datetime, @aFin datetime, @cIn datetime, @cFin datetime
AS
begin

```

```

    set nocount on;
    Select empy.cempno, cfedid, cfname, clname, csex, dbirth,
    empy.cdeptno, cdeptname, jobs.cDesc, nmonthpay, dcntrect, dhire,
    ctaxstate, empy.cpaytype, cstate, empg.cwageacc
    from mpsiafi.dbo.prempy empy, mpsiafi.dbo.HRJobs jobs,
        mpsiafi.dbo.prdept dept, mpsiafi.dbo.prempg empg
    where empy.cdeptno = dept.cdeptno
        and empy.cjobtitle = jobs.cJobTitlNO
        and empy.cempno = empg.cempno
        and dhire between isnull(@aIn, '19000101') and isnull(@aFin, '29001231')
        and dcntrect between isnull(@cIn, '19000101') and isnull(@cFin,

```

```

'29001231')
end
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```



```
CREATE procedure [dbo].[sp_slCargo1]
    @cargo varchar(40)
AS
begin
    Select mpsiafi.dbo.prempy.cempno, cfedid, cfname, clname, csex, dbirth,
    mpsiafi.dbo.prempy.cdeptno, cdeptname, cDesc, nmonthpay, dcntrct, dhire,
    ctaxstate, mpsiafi.dbo.prempy.cpaytype, cstate, mpsiafi.dbo.prempg.cwageacc
    from mpsiafi.dbo.prempy, mpsiafi.dbo.HRJobs,
        mpsiafi.dbo.prdept, mpsiafi.dbo.prempg
    where mpsiafi.dbo.prempy.cdeptno = mpsiafi.dbo.prdept.cdeptno
        and mpsiafi.dbo.prempy.cjobtitle = mpsiafi.dbo.HRJobs.cJobTitlNO
        and mpsiafi.dbo.prempy.cempno = mpsiafi.dbo.prempg.cempno
        and mpsiafi.dbo.HRJobs.cDesc LIKE '%' + @cargo + '%';
end
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

<b>Nombre del Reporte:</b>	<b>Reporte de Ingresos de empleados</b>		
<b>Descripción General del Reporte:</b>			
<b>Campos</b>	<b>Tablas</b>	<b>Descripción</b>	<b>Notas</b>
Codigo	prempy.cempno		
identidad	Prempy.cfedid		
nombres	Prempy.cfname		
apellidos	Prempy.clname		
cargo	HRJobs.cDesc		
Cod. depto	Prempy.cdeptno		
depto	Prdept.cdeptname		
referencia	Prmisc.cref		
banco	Prmisc.cbankno		
cheque	Prmisc.cchkno		
planilla	Prmisc.cpayno		
Cod. ingreso	Prmisc.cpaycode		
Valor ingreso	Prmisc.nqty * prmisc.nrate		

## Procedimiento Almacenado:

```

CREATE PROCEDURE [dbo].[sp_reporte_ingreso_empleados]
AS
BEGIN
SET NOCOUNT ON;
    SELECT E.cempno AS "CODIGO", E.cfedid AS "IDENTIDAD", E.cfname AS "NOMBRES",
    E.clname AS "APELLIDOS", J.cDesc AS "CARGO", E.cdeptno AS "COD. DEPTO.",
    D.cdeptname AS "DEPTO.", M.cref AS "REFERENCIA", M.cbankno AS "BANCO",
    M.cchkno AS "CHEQUE", M.cpayno AS "PLANILLA", M.cpaycode AS "COD. INGR.",
    (M.nqty*M.nrate) AS "VALOR INGR."
    FROM mpsiafi.dbo.prempy AS E, mpsiafi.dbo.prmisc AS M, mpsiafi.dbo.prdept AS
    D, mpsiafi.dbo.HrJobs AS J
    WHERE E.cdeptno = D.cdeptno AND E.cjobtitle = J.cJobTitlNO AND E.cempno =
    M.cempno
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

<b>Nombre del Reporte:</b>	<b>Reporte de Deducciones de empleados</b>		
<b>Descripción General del Reporte:</b>			
<b>Campos</b>	<b>Tablas</b>	<b>Descripción</b>	<b>Notas</b>
Codigo	prempy.cempno		
identidad	Prempy.cfedid		
nombres	Prempy.cfname		
apellidos	Prempy.clname		
cargo	HRJobs.cDesc		
Cod. depto	Prempy.cdeptno		
depto	Prdept.cdeptname		
planilla	Prmisd.cpayno		
Cod Ingreso (Deduccion)	Prmisd.cdeditcode		
descripcion	Prmisd.cdedit		
Valor deducciones	Prmisd.ndeditamt		

## Procedimientos Almacenados:

```

CREATE PROCEDURE [dbo].[sp_reporte_deducciones_empleados]
AS
BEGIN

    SET NOCOUNT ON;

    SELECT empl.cempno AS '# EMPLEADO', empl.cfedid AS 'IDENTIDAD',
           empl.cfname AS NOMBRES, empl.clname AS APELLIDOS,
           jobs.cDesc AS CARGO, empl.cdeptno AS 'COD. DEPTO.',
           dept.cdeptname AS 'DEPTO.', misd.cpayno AS PLANILLA,
           misd.cdeditno AS 'COD. DED.', misd.cdedit AS DESCRIPCION,
           misd.ndeditamt AS 'VALOR DED.'
    FROM mpsiafi.dbo.prempy AS empl
        INNER JOIN mpsiafi.dbo.HRJobs AS jobs
            ON empl.cjobtitle = jobs.cJobTitlNO
        INNER JOIN mpsiafi.dbo.prdept AS dept
            ON empl.cdeptno = dept.cdeptno
        INNER JOIN mpsiafi.dbo.prmisd AS misd
            ON empl.cempno = misd.cempno

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

Nombre del Reporte:	Reporte Presupuestario por Programa		
Descripción General del Reporte:			
Campos	Tablas	Descripción	Notas
Programa	(Bgbudt.cbudno)	Es el grupo de 3 caracteres a partir del primer guion	110-101-01-0101-11100
Presupuesto	Bgdudt.cbudid		
Transferencia	Bgdudt.namtmod - Bgdudt.namt		
Modificado	Bgdudt.namtmod		
Utilizado	Bgdudt.namtused		
Disponible	Bgdudt.namtmod – bgdudt.namtused		
%Utilizado	Bgdudt.namtused / Bgdudt.namtmod		
%Disponible	(Bgdudt.namtmod – bgdudt.namtused) / Bgdudt.namtmod		

## Procedimientos Almacenados:

```

CREATE PROCEDURE [dbo].[sp_reporte_presupuesto_programa]
AS
BEGIN

    SET NOCOUNT ON;

    SELECT SUBSTRING(budget.cbudno,5,3) AS PROGRAMA ,budget.cbudid AS
PRESUPUESTO,
        (SUM(budget.namtmod) - SUM(budget.namt)) AS TRANSFERENCIA,
        SUM(budget.namtmod) AS MODIFICADO, SUM(budget.namtused) AS
UTILIZADO,
        SUM(budget.namtmod - budget.namtused) AS DISPONIBLE,
        (SUM(budget.namtused) / SUM(budget.namtmod)) AS '% UTILIZADO',
        ((SUM(budget.namtmod) - SUM(budget.namtused)) /
SUM(budget.namtmod)) AS '% DISPONIBLE'
        FROM mpsiafi.dbo.bgbudt AS budget
        WHERE SUBSTRING(budget.cbudid,5,4) = YEAR(GETDATE())
        GROUP BY SUBSTRING(budget.cbudno,5,3), budget.cbudid

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

<b>Nombre del Reporte:</b>	<b>Reporte Presupuestario por Grupo</b>		
<b>Descripción General del Reporte:</b>			
<b>Campos</b>	<b>Tablas</b>	<b>Descripción</b>	<b>Notas</b>
Grupo	(Bgbudt.cbudno)	Son los digitos 1 del ultimo grupo de 5 caracteres	110-101-01-0101- 11100
Presupuesto	Bgdudt.cbudid		
Transferencia	Bgdudt.namtmod - Bgdudt.namt		
Modificado	Bgdudt.namtmod		
Utilizado	Bgdudt.namtused		
Disponible	Bgdudt.namtmod – bgdudt.namtused		
%Utilizado	Bgdudt.namtused / Bgdudt.namtmod		
%Disponible	(Bgdudt.namtmod – bgdudt.namtused) / Bgdudt.namtmod		

## Procedimientos Almacenados:

```

CREATE PROCEDURE [dbo].[sp_reporte_presupuesto_grupo]

AS
BEGIN

    SET NOCOUNT ON;

    SELECT SUBSTRING(budget.cbudno,17,1) AS GRUPO ,budget.cbudid AS PRESUPUESTO,
           (SUM(budget.namtmod) - SUM(budget.namt)) AS TRANSFERENCIA,
           SUM(budget.namtmod) AS MODIFICADO, SUM(budget.namtused) AS
UTILIZADO,
           SUM(budget.namtmod - budget.namtused) AS DISPONIBLE,
           (SUM(budget.namtused) / SUM(budget.namtmod)) AS '% UTILIZADO',
           ((SUM(budget.namtmod) - SUM(budget.namtused)) /
SUM(budget.namtmod)) AS '% DISPONIBLE'
    FROM mpsiafi.dbo.bgbudt AS budget
    WHERE SUBSTRING(budget.cbudid,5,4) = YEAR(GETDATE())
    GROUP BY SUBSTRING(budget.cbudno,17,1), budget.cbudid

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

<b>Nombre del Reporte:</b>	<b>Reporte Presupuestario por Subgrupo</b>		
<b>Descripción General del Reporte:</b>			
<b>Campos</b>	<b>Tablas</b>	<b>Descripción</b>	<b>Notas</b>
SubGrupo	Bgbudt.cbudno	Son los digitos 1 y 2 del ultimo grupo de 5 caracteres	110-101-01-0101-11100
Presupuesto	Bgdudt.cbudid		
Transferencia	Bgdudt.namtmod - Bgdudt.namt		
Modificado	Bgdudt.namtmod		
Utilizado	Bgdudt.namtused		
Disponible	Bgdudt.namtmod – bgdudt.namtused		
%Utilizado	Bgdudt.namtused / Bgdudt.namtmod		
%Disponible	(Bgdudt.namtmod – bgdudt.namtused) / Bgdudt.namtmod		

## Procedimientos Almacenados:

```

CREATE PROCEDURE [dbo].[sp_reporte_presupuesto_subgrupo]
AS
BEGIN
SELECT SUBSTRING(budget.cbudno,17,2) AS "SUB GRUPO",
       (budget.cbudid) AS "PRESUPUESTO",
       (SUM(budget.namtmod) - SUM(budget.namt)) AS "TRANSFERENCIA",
       SUM(budget.namtmod) AS "MODIFICADO",
       SUM(budget.namtused) AS "UTILIZADO",
       (SUM(budget.namtmod) - SUM(budget.namtused)) AS "DISPONIBLE",
       (SUM(budget.namtused) / SUM(budget.namtmod)) AS "% UTILIZADO",
       ((SUM(budget.namtmod) - SUM(budget.namtused)) / SUM(budget.namtmod)) AS "%
DISPONIBLE"
FROM mpsiafi.dbo.bgbudt as budget
WHERE SUBSTRING(budget.cbudid,5,4) = YEAR(GETDATE())
GROUP BY SUBSTRING(budget.cbudno,17,2), budget.cbudid;
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

<b>Nombre del Reporte:</b>	<b>Reporte Comparación Presupuesto Original</b>		
<b>Descripción General del Reporte:</b>	Presupuestos Originales		
<b>Campos</b>	<b>Tablas</b>	<b>Descripción</b>	<b>Notas</b>
Cuenta	Bgbudt.cbudno		
Presupuesto	Bgdudt.cbudid		
Original	Bgdudt.namt		
Modificado	Bgdudt.namtmod		
Utilizado	Bgdudt.namtused		
Disponible	Bgdudt.namtmod – bgdudt.namtused		
%Utilizado	Bgdudt.namtused / Bgdudt.namtmod		
%Disponible	(Bgdudt.namtmod – bgdudt.namtused) / Bgdudt.namtmod		

## Procedimientos Almacenados:

```

CREATE PROCEDURE [dbo].[sp_reporte_presupuesto_compara]
AS
BEGIN

    SET NOCOUNT ON;

    SELECT budget.cbudno AS CUENTA ,budget.cbudid AS PRESUPUESTO,
           (budget.namt) AS ORIGINAL,
           (budget.namtmod) AS MODIFICADO, (budget.namtused) AS UTILIZADO,
           (budget.namtmod – budget.namtused) AS DISPONIBLE,
           ((budget.namtused) / (budget.namtmod)) AS '% UTILIZADO',
           (((budget.namtmod) – (budget.namtused)) / (budget.namtmod)) AS '%
DISPONIBLE'
    FROM mpsiafi.dbo.bgbudt AS budget
    WHERE SUBSTRING(budget.cbudid,5,4) = YEAR(GETDATE())

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER OFF
GO

```

<b>Nombre del Reporte:</b>	<b>Reporte Transacciones</b>		
<b>Descripción General del Reporte:</b>	Presupuestos Originales		
<b>Campos</b>	<b>Tablas</b>	<b>Descripción</b>	<b>Notas</b>
Tipo		Es el tipo de transacción (Sea de transferencia o de ajuste)	
Desde	bgtrfm.cfbudno	Para la tabla de ajustes no va, para la tabla de transferencias es bgtrfm.cfbudno	
Hasta	bgtrfm.ctbudno bgadjm.cbudno		
Cantidad	bgadjm.nadjamt bgtrfm.ntrfamt		
Fecha	bgtrfm.tmodrec bgadjm.tmodrec		

## Procedimientos Almacenados:

```

CREATE PROCEDURE [dbo].[sp_reporte_presupuesto_transacciones]
AS
BEGIN

    SET NOCOUNT ON;

    ((SELECT 'TRANS.' AS TIPO, CONVERT(VARCHAR, trn.cfbudno) AS DESDE,
        trn.ctbudno AS HASTA, trn.ntrfamt AS 'CANT.',
        trn.tmodrec AS FECHA
        FROM mpsiafi.dbo.bgtrfm AS trn
        WHERE YEAR(trn.tmodrec) = YEAR(GETDATE()))
    UNION
    (SELECT 'AJUST.' AS TIPO, 'N/A' AS DESDE,
        aju.cbudno AS HASTA, aju.nadjamt AS 'CANT.',
        aju.tmodrec AS FECHA
        FROM mpsiafi.dbo.bgadjm AS aju
        WHERE YEAR(aju.tmodrec) = YEAR(GETDATE()))
    ORDER BY FECHA
END

```



# Diccionario de Datos

## Base de Datos: mpsiafireportes

### Nombre Tabla: tbl\_rol

No	Nombre Columna	PK	FK	M	Tipo de Dato
1	id_rol	P		Y	SMALLINT
2	descripcion				VARCHAR (45)

### Procedimiento Almacenado:

```
CREATE PROCEDURE [dbo].[sp_getroles]
AS
BEGIN
    SET NOCOUNT ON;

    SELECT r.id_rol AS ROL,r.descripcion AS 'DESC'
        FROM tbl_rol AS r
END
GO
```

### Nombre Tabla: tbl\_rol\_usuario

No	Nombre Columna	PK	FK	M	Tipo de Dato
1	id_usuario	P	F	Y	VARCHAR (45)
2	id_rol	P	F	Y	SMALLINT

## Nombre Tabla: tbl\_usuario

No	Nombre Columna	PK	FK	M	Tipo de Dato
1	id_usuario	P		Y	VARCHAR (45)
2	prnombre			Y	VARCHAR (50)
3	sgnombre				VARCHAR (50)
4	prapellido			Y	VARCHAR (50)
5	sgapellido				VARCHAR (50)
6	clave			Y	VARBINARY
7	fecha_creacion				Datetime
8	fecha_ultimo_ingreso				Datetime
9	email				VARCHAR (100)
10	id_departamento			Y	Integer

## Procedimientos Almacenados:

```
CREATE PROCEDURE [dbo].[sp_nuevousuario]
    @idusuario VARCHAR(45), @correo VARCHAR(100),
    @prnombre VARCHAR(50), @sgnombre VARCHAR(50)=NULL,
    @prapellido VARCHAR(50), @sgapellido VARCHAR(50)=NULL,
    @clave VARCHAR(50)=NULL, @rol TINYINT=3
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    DECLARE @error INT;

    BEGIN TRANSACTION
        INSERT INTO tbl_usuarios (id_usuario, email, prnombre,
            sgnombre, prapellido, sgapellido, clave, fecha_creacion)
            VALUES (@idusuario, @correo, @prnombre,
                @sgnombre, @prapellido, @sgapellido,
                PWDENCRYPT(ISNULL(@clave, '123456')), GETDATE())

        INSERT INTO rol_usuarios (id_usuario, id_rol)
            VALUES (@idusuario, @rol)

    SET @error = @@ERROR;
    IF @error <> 0 BEGIN
        COMMIT
        RETURN 0;
    END
    ELSE BEGIN
        ROLLBACK
        RETURN @error;
    END
END
GO
```

```

CREATE PROCEDURE [dbo].[sp_eliminausuario]
    @correo VARCHAR(100)
AS
BEGIN

    SET NOCOUNT ON;

    DELETE tbl_usuarios
        WHERE email = @correo
END
GO

CREATE PROCEDURE [dbo].[sp_getusuarios]
    @id VARCHAR(45)='',@correo VARCHAR(100)=''
AS
BEGIN
    SET NOCOUNT ON;
    SELECT u.email AS CORREO,u.id_usuario AS ID
        FROM tbl_usuarios AS u
        WHERE u.email LIKE '%' + @correo + '%'
            AND u.id_usuario LIKE '%' + @id + '%'
END
GO

CREATE PROCEDURE [dbo].[sp_login]
    @correo VARCHAR(100),@clave VARCHAR(50)
AS
BEGIN

    SET NOCOUNT ON;

    SELECT TOP 1 (usr.pnombre + ' ' + usr.prapellido + ' - ' + rol.descripcion)
AS NOMBRE, rol.id_rol AS ROL
        FROM mpsiafireportes.dbo.tbl_usuarios AS usr
            INNER JOIN mpsiafireportes.dbo.tbl_rol_usuario AS rolusr
                ON usr.id_usuario = rolusr.id_usuario
            INNER JOIN mpsiafireportes.dbo.tbl_rol AS rol
                ON rol.id_rol = rolusr.id_rol
        WHERE usr.email = @correo AND PWDCOMPARE(@clave,usr.clave)=1

    UPDATE mpsiafireportes.dbo.tbl_usuarios
        SET fecha_ultimo_ingreso = GETDATE()
        WHERE email = @correo
END
GO

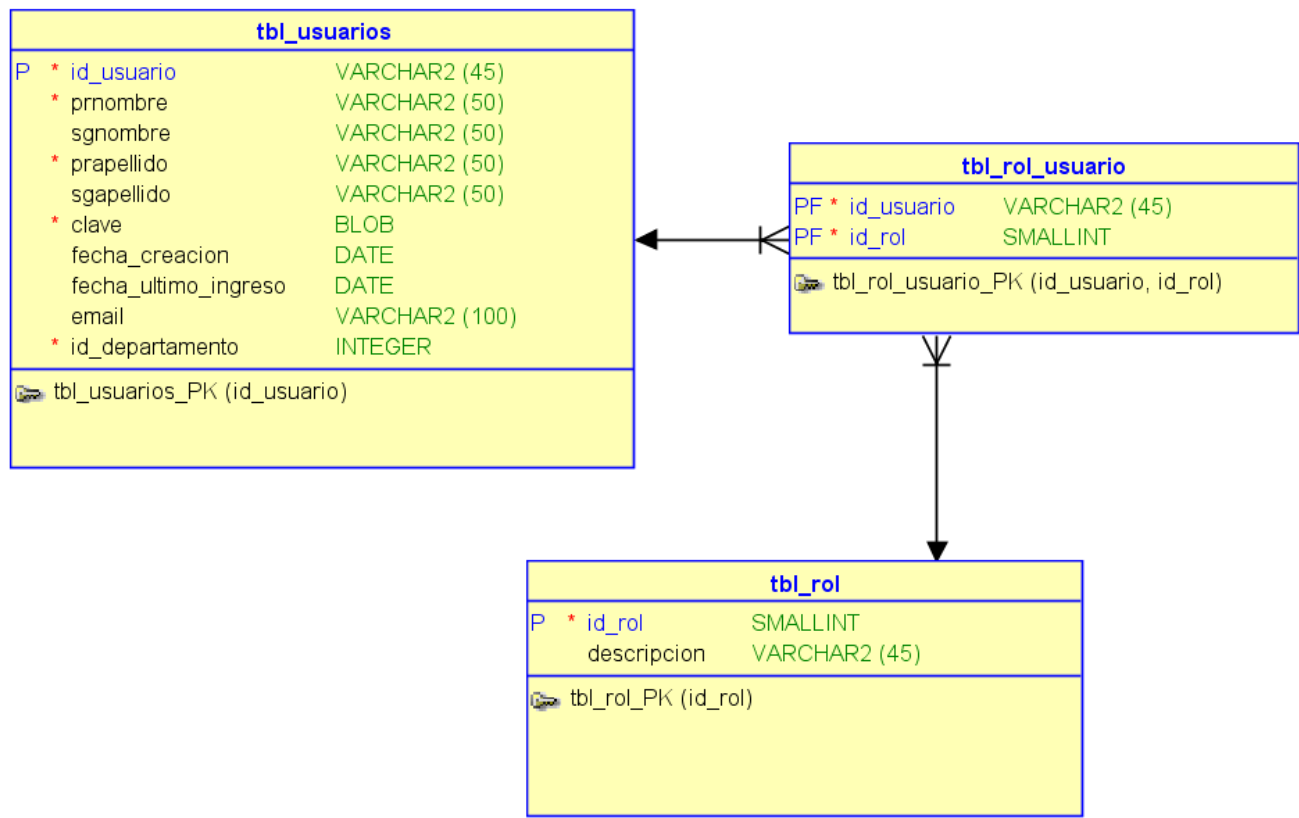
CREATE PROCEDURE [dbo].[sp_reiniciaclave]
    @correo VARCHAR(100),@clave VARCHAR(50)='123456'
AS
BEGIN

    SET NOCOUNT ON;

    UPDATE tbl_usuarios
        SET clave = PWDENCRYPT(@clave)
        WHERE email=@correo;
END
GO

```

## Diagrama Relacional



## Base de datos: mpsiafi

### Nombre Tabla: bgadjm

No	Nombre Columna	M	Tipo de Dato
1	cadjno	Y	CHAR
2	cbudid	Y	CHAR
3	cbudno	Y	CHAR
4	capppcode	Y	CHAR
5	cdesc	Y	CHAR
6	dtrs		Datetime
7	cuser	Y	CHAR
8	nadjamt	Y	NUMERIC
9	nadjqty	Y	NUMERIC
10	mnotepad	Y	TEXT
11	tmodrec	Y	Datetime
12	lvoid	Y	SMALLINT
13	dvoid		Datetime

## Nombre Tabla: bgtrfm

No	Nombre Columna	M	Tipo de Dato
1	ctrfno		CHAR
2	cbudid		CHAR
3	cfbudno		CHAR
4	ctbudno		CHAR
5	cappcode		CHAR
6	cdesc		CHAR
7	dtrs		Datetime
8	cuser		CHAR
9	ntrfamt		NUMERIC
10	ntrfqty		NUMERIC
11	mnotepad		TEXT
12	tmodrec		Datetime
13	lvoid		SMALLINT
14	dvoid		Datetime

## Nombre Tabla: prempg

No	Nombre Columna	M	Tipo de Dato
1	cuid	Y	CHAR
2	cempno	Y	CHAR
3	cwageacc	Y	CHAR
4	ctaxacc	Y	CHAR
5	ndistpct	Y	NUMERIC
6	cfwtacc	Y	CHAR
7	cficaacc	Y	CHAR
8	cmediacc	Y	CHAR
9	cfutaacc	Y	CHAR
10	csutaacc	Y	CHAR
11	cficaeracc	Y	CHAR
12	cmedieracc	Y	CHAR
13	cfutaeracc	Y	CHAR
14	csutaeracc	Y	CHAR

## Nombre Tabla: HRJobs

No	Nombre Columna	M	Tipo de Dato
1	cJobTitlNO	Y	VARCHAR
2	cDesc	Y	VARCHAR
3	nSalary	Y	DECIMAL
4	nPorPlus	Y	DECIMAL
5	mnotepad		VARCHAR
6	tmodrec	Y	Datetime

## Nombre Tabla: apchck

No	Nombre Columna	M	Tipo de Dato
1	cvendno	Y	CHAR
2	cpayto	Y	CHAR
3	cchktype	Y	CHAR
4	cbankno	Y	CHAR
5	cchkno	Y	CHAR
6	ctogl	Y	CHAR
7	dcheck	Y	Datetime
8	dcreate	Y	Datetime
9	lcancel	Y	SMALLINT
10	lhold	Y	SMALLINT
11	inchkamt	Y	NUMERIC
12	nfchkamt	Y	NUMERIC
13	nxchgrate	Y	NUMERIC
14	nbchkamt	Y	NUMERIC

## Nombre Tabla: bgbudp

No	Nombre Columna	M	Tipo de Dato
1	cuid	Y	CHAR
2	cbudid	Y	CHAR
3	cbudno	Y	CHAR
4	cyear	Y	CHAR
5	cpdno	Y	CHAR
6	namt	Y	NUMERIC
7	namtused	Y	NUMERIC
8	nqty	Y	NUMERIC
9	nqtyused	Y	NUMERIC
10	tmodrec	Y	Datetime
11	namtaccr	Y	NUMERIC
12	nqtyaccr	Y	NUMERIC
13	namtmod	Y	NUMERIC
14	nqtymod	Y	NUMERIC

## Nombre Tabla: bgbudt

No	Nombre Columna	M	Tipo de Dato
1	cuid	Y	CHAR
2	cbudid	Y	CHAR
3	cbudno	Y	CHAR
4	cdesc	Y	CHAR
5	cstmacct	Y	CHAR
6	crsvacct	Y	CHAR
7	cclsid	Y	CHAR
8	ctypid	Y	CHAR
9	cdeptno	Y	CHAR
10	namt	Y	NUMERIC
11	namtused	Y	NUMERIC
12	nqty	Y	NUMERIC
13	nqtyused	Y	NUMERIC
14	tmodrec	Y	Datetime
15	namtaccr	Y	NUMERIC
16	nqtyaccr	Y	NUMERIC
17	namtmod	Y	NUMERIC
18	nqymod	Y	NUMERIC

## Nombre Tabla: prmsid

No	Nombre Columna	M	Tipo de Dato
1	cuid	Y	CHAR
2	cempno	Y	CHAR
3	cdedno	Y	CHAR
4	cdedcode	Y	CHAR
5	cpayno	Y	CHAR
6	ndedamt	Y	NUMERIC
7	tmodrec	Y	Datetime
8	lsystem	Y	SMALLINT
9	cdesc	Y	CHAR
10	ctrsno	Y	CHAR

## Nombre Tabla: prmisc

No	Nombre Columna	M	Tipo de Dato
1	cuid	Y	CHAR
2	cempno		CHAR
3	clname		CHAR
4	cfname		CHAR
5	ctaxstate		CHAR
6	cdeptno		CHAR
7	cpaytype		CHAR
8	cref		CHAR
9	ctrsno		CHAR
10	cbankno		CHAR
11	cchkno		CHAR
12	dtrs		Datetime
13	dcheck		Datetime
14	ltaxable		SMALLINT
15	nohtax		NUMERIC
16	nohtntax		NUMERIC
17	csegid1		CHAR
18	csegid2		CHAR
19	csegid3		CHAR
20	csegid4		CHAR
21	csegid5		CHAR
22	csegmid		CHAR
23	ccycle		CHAR
24	cpayno		CHAR
25	cpaycode		CHAR
26	nqty		NUMERIC
27	nrate		NUMERIC
28	lssystem		SMALLINT
29	nohtertax		NUMERIC
30	nBaseGrTax		NUMERIC
31	cdocno		CHAR
32	cdoctyp		CHAR



## Nombre Tabla: prdept

No	Nombre Columna	M	Tipo de Dato
1	cdeptno	Y	CHAR
2	cdeptname	Y	CHAR
3	cwageacc	Y	CHAR
4	ctaxacc	Y	CHAR
5	cpaytype	Y	CHAR
6	cpayperiod	Y	CHAR
7	lallowot		SMALLINT
8	lepayment		SMALLINT
9	lusetips		SMALLINT
10	nrhr	Y	NUMERIC
11	cficaeracc	Y	CHAR
12	cmedieracc	Y	CHAR
13	cfutaeracc	Y	CHAR
14	csutaeracc	Y	CHAR
15	cficaacc	Y	CHAR
16	cmediacc	Y	CHAR
17	cfutaacc	Y	CHAR
18	csutaacc	Y	CHAR
19	cdeptglseg	Y	CHAR

## Nombre Tabla: prempy

No	Nombre Columna	M	Tipo de Datos
1	cempno	Y	CHAR
2	cfname	Y	CHAR
3	clname	Y	CHAR
4	cinitial	Y	CHAR
5	cssn		CHAR
6	cdeptno	Y	CHAR
7	cstatus	Y	CHAR
8	cfedid		CHAR
9	caddr1	Y	CHAR
10	caddr2	Y	CHAR
11	ccity	Y	CHAR
12	cstate	Y	CHAR
13	czip	Y	CHAR
14	cphone	Y	CHAR
15	cjobtitle	Y	CHAR
16	csex	Y	CHAR
17	ccitizen	Y	CHAR
18	ceeoid	Y	CHAR

No	Nombre Columna	M	Tipo de Datos
19	cethniccd	Y	CHAR
20	cpaytype	Y	CHAR
21	cpayperiod	Y	CHAR
22	cmarry	Y	CHAR
23	cmtype	Y	CHAR
24	ceicstatus	Y	CHAR
25	ctaxstate	Y	CHAR
26	csutastate	Y	CHAR
27	cltaxstate	Y	CHAR
28	cltaxcd	Y	CHAR
29	capplycd	Y	CHAR
30	cprenote	Y	CHAR
31	cw2year	Y	CHAR
32	dhire		Datetime
33	dterminate		Datetime
34	dbirth		Datetime
35	deff		Datetime
36	dnextpf		Datetime
37	dnextsal		Datetime
38	dlastpaid		Datetime
39	dprenote		Datetime
40	lallowot		SMALLINT
41	lepayment		SMALLINT
42	lusetips		SMALLINT
43	lfica		SMALLINT
44	lmedi		SMALLINT
45	lfuta		SMALLINT
46	lsuta		SMALLINT
47	lsdi		SMALLINT
48	leic		SMALLINT
49	lftax		SMALLINT
50	lstax		SMALLINT
51	lsres		SMALLINT
52	lltax		SMALLINT
53	llres		SMALLINT
54	lapply		SMALLINT
55	nbpayrate	Y	NUMERIC
56	napayrate	Y	NUMERIC

No	Nombre Columna	M	Tipo de Datos
57	nrhr	Y	NUMERIC
58	nafwt	Y	NUMERIC
59	nfallow	Y	NUMERIC
60	naswt	Y	NUMERIC
61	nsallow	Y	NUMERIC
62	nalwt	Y	NUMERIC
63	csrcid	Y	CHAR
64	cplnid	Y	CHAR
65	cshfid	Y	CHAR
66	cmssc1id	Y	CHAR
67	cmssc2id	Y	CHAR
68	cmssc3id	Y	CHAR
69	cmssc4id	Y	CHAR
70	cmssc5id	Y	CHAR
71	csegid1	Y	CHAR
72	csegid2	Y	CHAR
73	csegid3	Y	CHAR
74	csegid4	Y	CHAR
75	csegid5	Y	CHAR
76	lPaySlip	Y	SMALLINT
77	ccustno	Y	CHAR
78	cdayoff	Y	CHAR
79	cbatchid	Y	CHAR
80	nmonthpay	Y	NUMERIC
81	n1stmthpay	Y	NUMERIC
82	ltemporal	Y	SMALLINT
83	nfixover	Y	NUMERIC
84	crtn	Y	CHAR
85	dcntrct		Datetime
86	laddtax1	Y	SMALLINT
87	laddtax2	Y	SMALLINT
88	laddtax3	Y	SMALLINT
89	ltax11mth	Y	SMALLINT

# Diagrama Relacional

