CARATULA

**INDICE**

CASOS DE USO 4

CU005: Registrar nuevo cliente 4

Diagrama del caso de uso 4

Especificación del caso de uso 4

Prototipo 4

Diagrama de Secuencia 4

Diagrama ER del caso de uso 4

Análisis de la caja negra del SP 4

Código del SP 5

Prueba del SP 6

CU006: Registrar un deposito 7

Diagrama del caso de uso 7

Especificación del caso de uso 7

Prototipo 7

Diagrama de Secuencia 7

Diagrama ER del caso de uso 7

Análisis de la caja negra del SP 7

Código del SP 7

Prueba del SP 7

CU007: Registrar un retiro 8

Diagrama del caso de uso 8

Especificación del caso de uso 8

Prototipo 8

Diagrama de Secuencia 8

Diagrama ER del caso de uso 8

Análisis de la caja negra del SP 8

Código del SP 8

Prueba del SP 8

CU008: Registrar una transferencia 8

Diagrama del caso de uso 8

Especificación del caso de uso 8

Prototipo 9

Diagrama de Secuencia 9

Diagrama ER del caso de uso 9

Análisis de la caja negra del SP 9

Código del SP 9

Prueba del SP 9

# CASOS DE USO

## CU001: Registrar nuevo cliente

### Diagrama del caso de uso

### Especificación del caso de uso

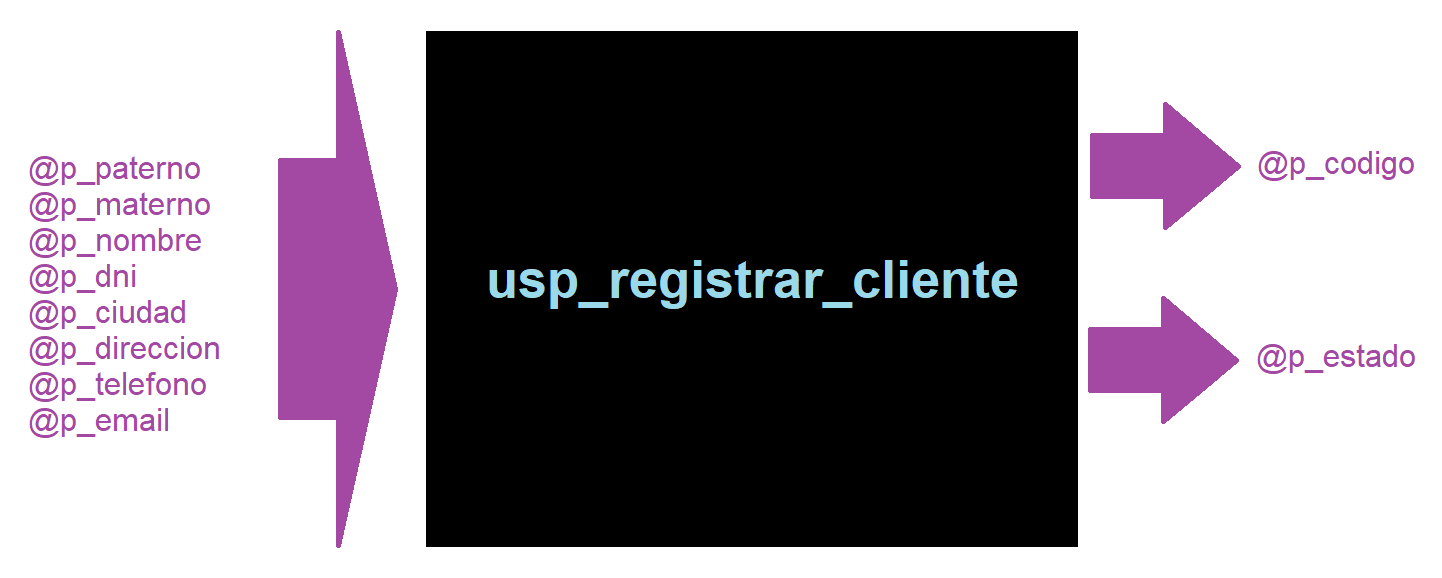
### Prototipo

### Diagrama de Secuencia

### Diagrama ER del caso de uso



### Análisis de la caja negra del SP



### Código del SP

create procedure usp\_registra\_cliente

(

@p\_paterno varchar (25),

@p\_materno varchar (25),

@p\_nombre varchar (30),

@p\_dni char (8 ),

@p\_ciudad varchar (30),

@p\_direccion varchar (50),

@p\_telefono varchar (20),

@p\_email varchar (50),

@p\_codigo char (5) out,

@p\_estado int out

)

as

begin

-- Declaracion de variables

declare @contador int, @tamanio int;

-- Validacion

if(@p\_paterno is null or @p\_materno is null or @p\_nombre is null)

begin

set @p\_estado = -3;

return; -- Fin del programa

end;

-- Proceso

set @p\_estado = 1;

BEGIN TRY

-- Inicio de la TX

BEGIN TRANSACTION;

-- Incrementar el contador

update contador set int\_contitem = int\_contitem + 1

where vch\_conttabla = 'Cliente';

if(@@ROWCOUNT <> 1)

begin

set @p\_estado = -2;

throw 51000, 'No se puede crear el codigo del cliente.', 1;

end;

select @contador=int\_contitem, @tamanio=int\_contlongitud

from Contador

where vch\_conttabla = 'Cliente';

set @p\_codigo = RIGHT(concat('00000',@contador),@tamanio);

-- Insertar el registro

INSERT INTO CLIENTE(chr\_cliecodigo,vch\_cliepaterno,vch\_cliematerno,

vch\_clienombre,chr\_cliedni,vch\_clieciudad,vch\_cliedireccion,

vch\_clietelefono,vch\_clieemail) VALUES(@p\_codigo,@p\_paterno,

@p\_materno,@p\_nombre,@p\_dni,@p\_ciudad,@p\_direccion,@p\_telefono,@p\_email);

-- Confirmar la TX

COMMIT TRANSACTION;

END TRY

BEGIN CATCH

-- Cancelar la TX

ROLLBACK TRANSACTION;

if(@p\_estado = 1) set @p\_estado = -1;

END CATCH;

end;

go

### Prueba del SP

**Prueba 1**

Esta es una prueba exitosa.

declare @codigo char(5), @estado int;

exec usp\_registra\_cliente 'GARCIA','ROJAS','ADRIANA','76345698','Lima',

'Los Olivos','963476892','carlos@gmail.com', @codigo out, @estado out;

print concat('Estado: ', @estado);

print concat('Codigo: ', @codigo);

go

Estado: 1

Codigo: 00023

**Prueba 2**

Esta se trata de una prueba con error, debido a que se esta intentando un NULL en el nombre del cliente.

declare @codigo char(5), @estado int;

exec usp\_registra\_cliente 'VASQUEZ','VALVERDE',NULL,'12345678','Lima',

'Los Olivos','963476892','carlos@gmail.com', @codigo out, @estado out;

print concat('Estado: ', @estado);

print concat('Codigo: ', @codigo);

go

Estado: -3

Codigo:

## CU002: Crear cuenta

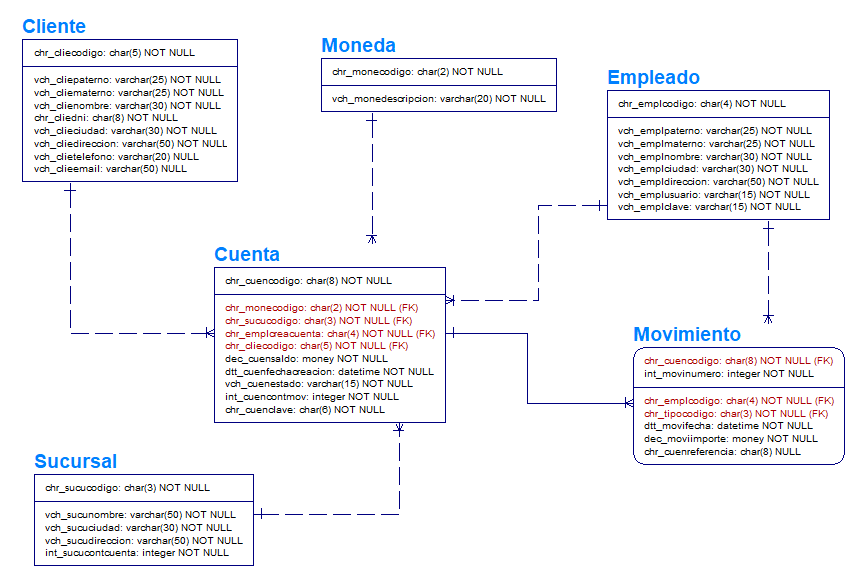
### Diagrama del caso de uso

### Especificación del caso de uso

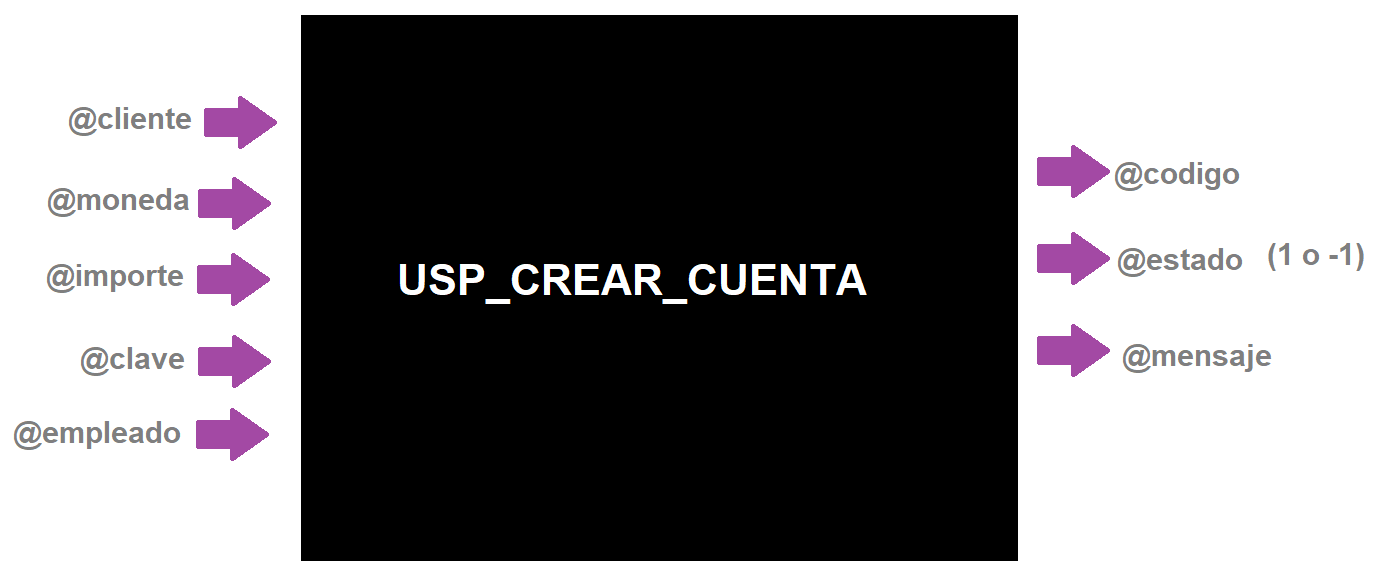
### Prototipo

### Diagrama de Secuencia

### Diagrama ER del caso de uso



### Análisis de la caja negra del SP



### Código del SP

CREATE PROCEDURE USP\_CREAR\_CUENTA(

@CLIENTE CHAR(5),

@MONEDA CHAR(2),

@IMPORTE MONEY,

@CLAVE CHAR(6),

@EMPLEADO CHAR(4),

@CODIGO CHAR(8) OUT,

@ESTADO INT OUT,

@MENSAJE VARCHAR(1000) OUT

) AS

BEGIN

DECLARE @cont int, @sucursal char(3);

SET @ESTADO = 1;

BEGIN TRY

-- Inicia Tx

BEGIN TRANSACTION;

-- Generar codigo de la cuenta

SELECT @sucursal=chr\_sucucodigo FROM Asignado

WHERE chr\_emplcodigo=@EMPLEADO AND dtt\_asigfechabaja IS NULL;

SELECT @cont=int\_sucucontcuenta FROM SUCURSAL WHERE chr\_sucucodigo=@sucursal;

SET @CODIGO = @sucursal + RIGHT(CONCAT('00000',@cont),5);

--Proceso

INSERT INTO CUENTA(chr\_cuencodigo,chr\_monecodigo,chr\_sucucodigo,

chr\_emplcreacuenta,chr\_cliecodigo,dec\_cuensaldo,dtt\_cuenfechacreacion,

vch\_cuenestado,int\_cuencontmov,chr\_cuenclave)

VALUES(@CODIGO,@MONEDA,@sucursal,@EMPLEADO,@CLIENTE,@IMPORTE,GETDATE(),'ACTIVO',1,@CLAVE);

INSERT INTO MOVIMIENTO(chr\_cuencodigo,int\_movinumero,dtt\_movifecha,chr\_emplcodigo,

chr\_tipocodigo,dec\_moviimporte)

VALUES(@CODIGO,1,getdate(),@EMPLEADO,'001',@IMPORTE);

-- Confirma Tx

COMMIT TRANSACTION;

END TRY

BEGIN CATCH

if(@ESTADO=1)

begin

set @ESTADO = -1;

set @MENSAJE = 'Error en el proceso, intentelo nuevamente.';

end;

print ERROR\_MESSAGE();

ROLLBACK TRANSACTION;

END CATCH;

END;

GO

### Prueba del SP

DECLARE @CODIGO CHAR(8), @ESTADO INT,@MENSAJE VARCHAR(1000);

EXEC USP\_CREAR\_CUENTA '00014','01',6500.0,'NOSELODIGASANADIE','0005',

@CODIGO OUT, @ESTADO OUT, @MENSAJE OUT;

PRINT 'CODIGO: ' + @CODIGO;

PRINT CONCAT('ESTADO: ',@ESTADO);

PRINT CONCAT('MENSAJE: ', @MENSAJE);

GO

CODIGO: 00400002

ESTADO: 1

MENSAJE: Todo ok.

## CU006: Registrar un deposito

### Diagrama del caso de uso

### Especificación del caso de uso

### Prototipo

### Diagrama de Secuencia

### Diagrama ER del caso de uso

### Análisis de la caja negra del SP

### Código del SP

### Prueba del SP

## CU007: Registrar un retiro

### Diagrama del caso de uso

### Especificación del caso de uso

### Prototipo

### Diagrama de Secuencia

### Diagrama ER del caso de uso

### Análisis de la caja negra del SP

### Código del SP

### Prueba del SP

## CU008: Registrar una transferencia

### Diagrama del caso de uso

### Especificación del caso de uso

### Prototipo

### Diagrama de Secuencia

### Diagrama ER del caso de uso

### Análisis de la caja negra del SP

### Código del SP

### Prueba del SP