

**Universidad Autónoma “Gabriel Rene Moreno”
Facultad de Ingeniería en Ciencias de la Computación y
Telecomunicaciones**



**PRIMER PARCIAL
CALCULADORA DE SALARIOS – 3 CAPAS**

Estudiantes: Zeballos Carvallo Oscar Daniel 218060556

Materia: Arquitectura de Software

Grupo: SA

Docente: Ing. Josue Obed Veizaga

Santa Cruz - Bolivia

Contenido

1.	Descripción del Problema	3
1.1	Objetivos	3
1.1.1	Objetivo General	3
1.1.2	Objetivo Específico	3
2.	Patrones de diseño	3
2.1	Plantilla.....	3
2.2	Memento.....	3
3.	Requerimientos	4
3.1	Identificar Actores y casos de uso.....	4
3.1.1	Actores.....	4
3.1.2	Casos de Uso.....	4
3.2	Detallar casos de uso	4
3.2.1	Gestionar empleado	4
3.2.2	Gestionar salarios.....	6
3.3	Diagrama general de casos de uso	8
4.	Identificación de módulos.....	8
5.	Diseño de la Arquitectura Lógica	8
6.	Diseño de la Base de Datos	9
6.1	Diseño Conceptual.....	9
6.2	Diseño lógico de la base de datos	9
6.3	Diseño Físico de la base datos	9
7.	Diseño de detalle procedimental.....	10
7.1	Diseño de clases dinámico.....	10

1. Descripción del Problema

La empresa X, no tiene una forma de poder calcular el salario de sus trabajadores, los cuales se diferencian entre seniors y juniors. El pago es por hora trabajada y por antigüedad. Hay una diferencia notable en el precio de hora trabajada de un senior a un junior.

1.1 Objetivos

1.1.1 Objetivo General

Desarrollar una calculadora de salarios la cual pueda hacer el cálculo del salario tanto para seniors y para juniors.

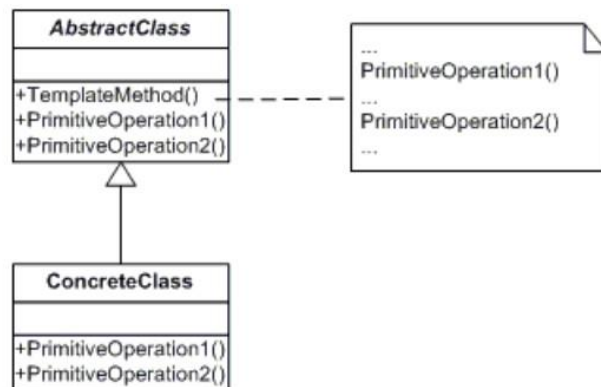
1.1.2 Objetivo Específico

- Permitir al administrador calcular el salario mensual de un trabajador, sea senior o sea junior.

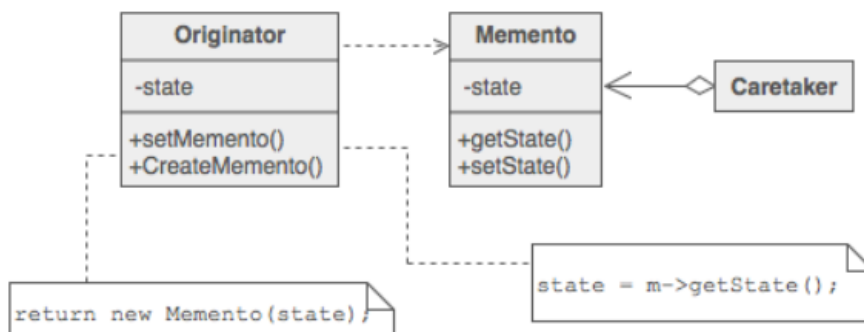
2. Patrones de diseño

2.1 Plantilla

UML Class Diagram



2.2 Memento



3. Requerimientos

3.1 Identificar Actores y casos de uso

3.1.1 Actores

Actor – Administrador: El único actor del sistema. Es el responsable del uso del sistema.

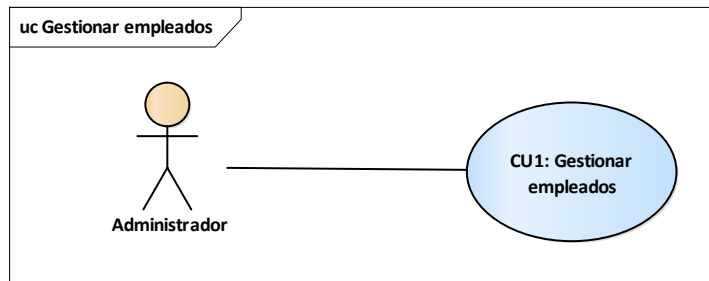
3.1.2 Casos de Uso

CU1 – Gestionar empleados: Gestiona los empleados de la empresa.

CU2 – Gestionar salarios: Gestiona los salarios de los empleados.

3.2 Detallar casos de uso

3.2.1 Gestionar empleado



CASO DE USO	GESTIONAR EMPLEADOS
Proposito	Crear los empleados
Descripcion	Gestionar la creacion
Actores	Administrador
Actor iniciador	Administrador
Precondicion	Ninguna
Postcondicion	Ninguna
Proceso	1. Crear empleado a. Insertar datos b. Guardar datos

▼

Nombre

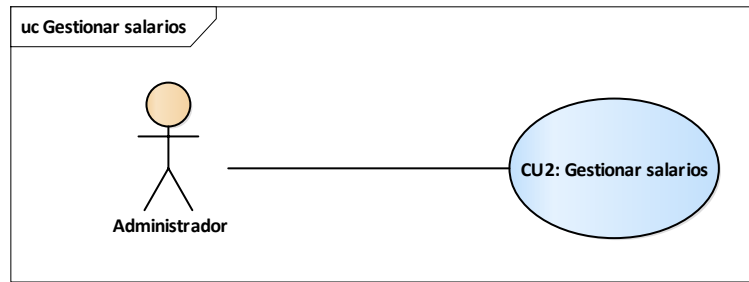
Apellidos

Cargo (S - J)

Crear

Cerrar

3.2.2 Gestionar salarios



CASO DE USO		GESTIONAR SALARIOS
Proposito	Gestionar los salarios de los empleados	
Descripcion	Gestionar la creacion planilla de salarios	
Actores	Administrador	
Actor iniciador	Administrador	
Precondicion	Creacion de empleados	
Postcondicion	Ninguna	
Proceso	1. Crear planilla de salarios a. Insertar datos b. Guardar datos	

Empleado

1 - Anabel - S

Horas

Bonificacion

Observacion

Crear

Cerrar

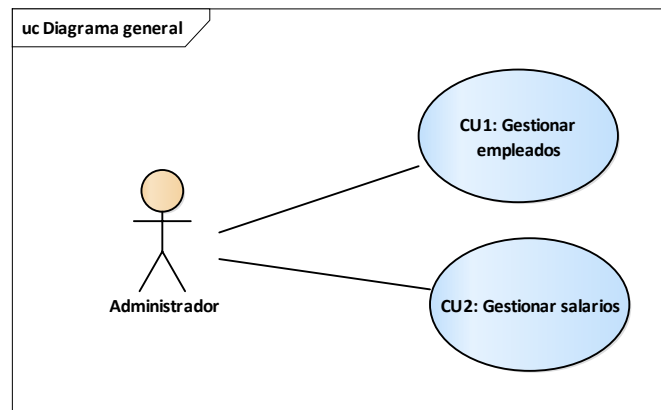
Nombre	Apellido	Horas	Bonificacion	Total	Observaciones

Ver salarios

Eliminar

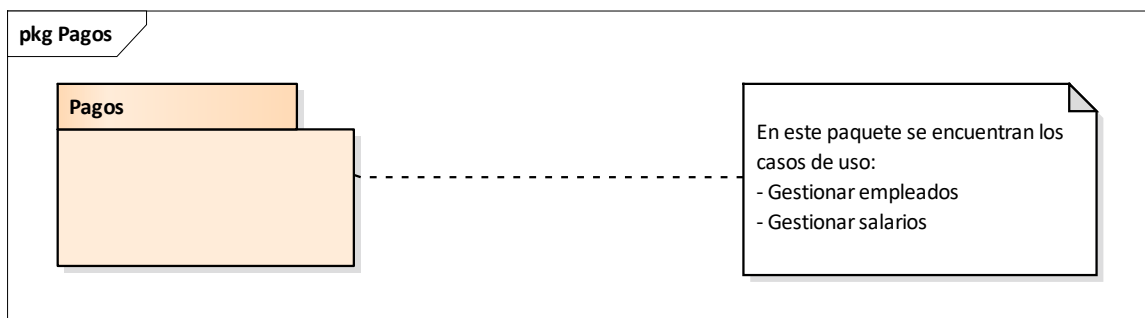
Cerrar

3.3 Diagrama general de casos de uso



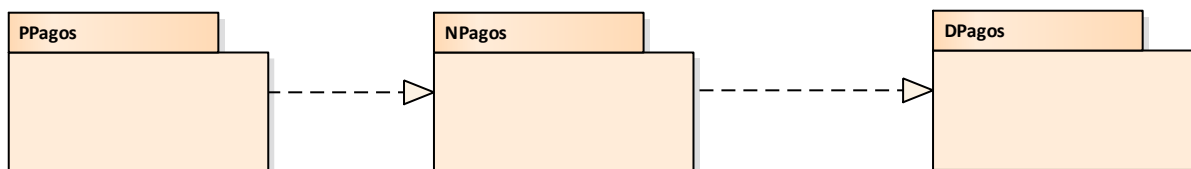
4. Identificación de módulos

Modulo pagos



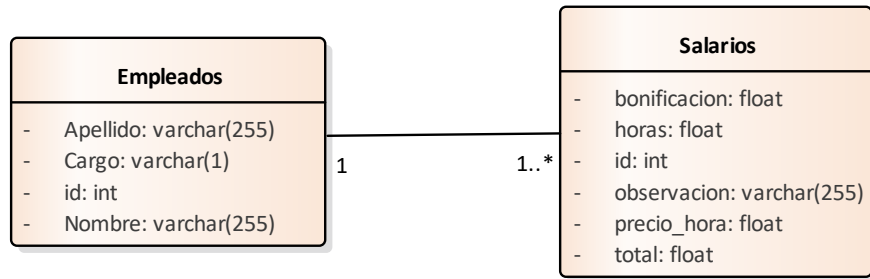
5. Diseño de la Arquitectura Lógica

Módulo pagos



6. Diseño de la Base de Datos

6.1 Diseño Conceptual



6.2 Diseño Lógico de la base de datos

empleados

Column	Type	Comment	PK	FK	Nullable	Default
id	integer		YES		NO	
nombre	varchar(255)				YES	
apellido	varchar(255)				YES	
cargo	varchar(1)				YES	

salarios

Column	Type	Comment	PK	FK	Nullable	Default
id	integer		YES		NO	
horas	double precision				YES	
precio_hora	double precision				YES	
bonificacion	double precision				YES	
total	double precision				YES	
observacion	varchar(255)				YES	
id_empleados	integer			YES	YES	

6.3 Diseño Físico de la base datos

```
CREATE DATABASE salarios_bd;  
USE salarios_bd;
```

```
CREATE TABLE empleados (  
  id SERIAL PRIMARY KEY,  
  nombre varchar(255),
```

```

    apellido varchar(255),
    cargo varchar(1)
)

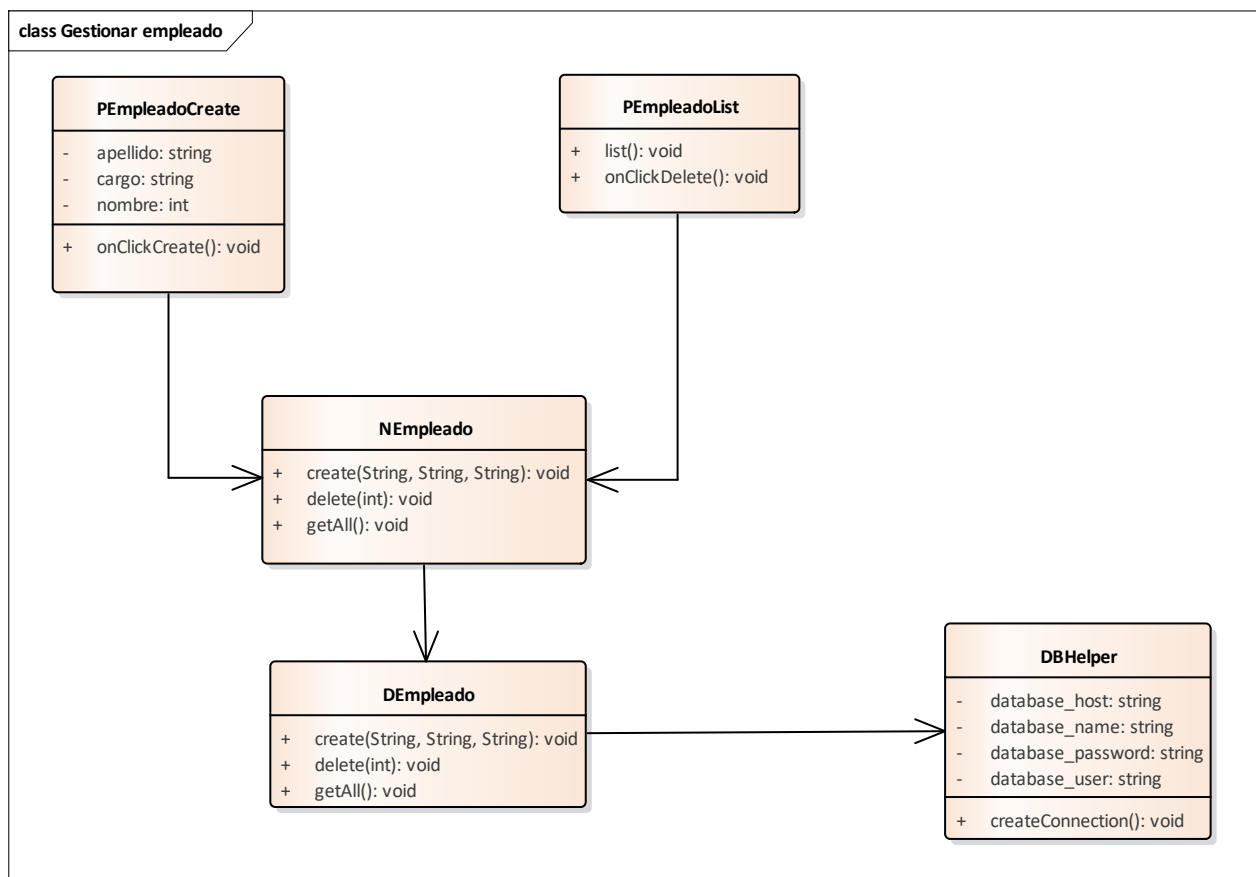
CREATE TABLE salarios (
    id SERIAL PRIMARY KEY,
    horas float,
    bonificacion float,
    total float,
    observacion varchar(255),
    id_empleados int,
    FOREIGN KEY (id_empleados)
    REFERENCES empleados(id)
)

```

7. Diseño de detalle procedimental

7.1 Diseño de clases dinámico

CU1 Gestionar empleados



CU2: Gestionar salarios

