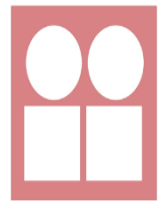




Universidad Nacional
Autónoma de México

RED UNIVERSITARIA DE COLABORACIÓN
EN INGENIERÍA DE SOFTWARE Y BASES DE DATOS



INSTITUTO
DE INGENIERÍA
UNAM

Documentos y herramientas para el proceso de ingeniería de software

Ing. Julio Alfonso De León Razo

Agenda

- Antecedentes
- Proceso de Ingeniería de Software
- Conclusión
- Preguntas

Proceso de Ingeniería de Software

- Toma de requerimientos
- Planeación
- Análisis
- Diseño
- Construcción
- Pruebas
- Implementación
- Mantenimiento



Toma de requerimientos

➤ Herramientas

➤ Documento para especificar los requerimientos de Software

SRS.

Objetivo

[Indicar el objetivo de la toma de requerimientos. El problema que se pretende resolver.]

[El Sistema nombre completo del sistema (siglas del sistema) permitirá proceso que realizará el sistema así como también, trabajará en conjunto con otros sistemas, tal es el caso de otro(s) sistema(s).]

Requerimientos Funcionales

[Con base en las necesidades identificadas con la toma de requerimientos, describa aquí los requerimientos funcionales que deberá satisfacer la solución.]

ID	Nombre	Descripción
RF1		
RF1.1		
RF1.2		
RF2		
RF3		

Planeación

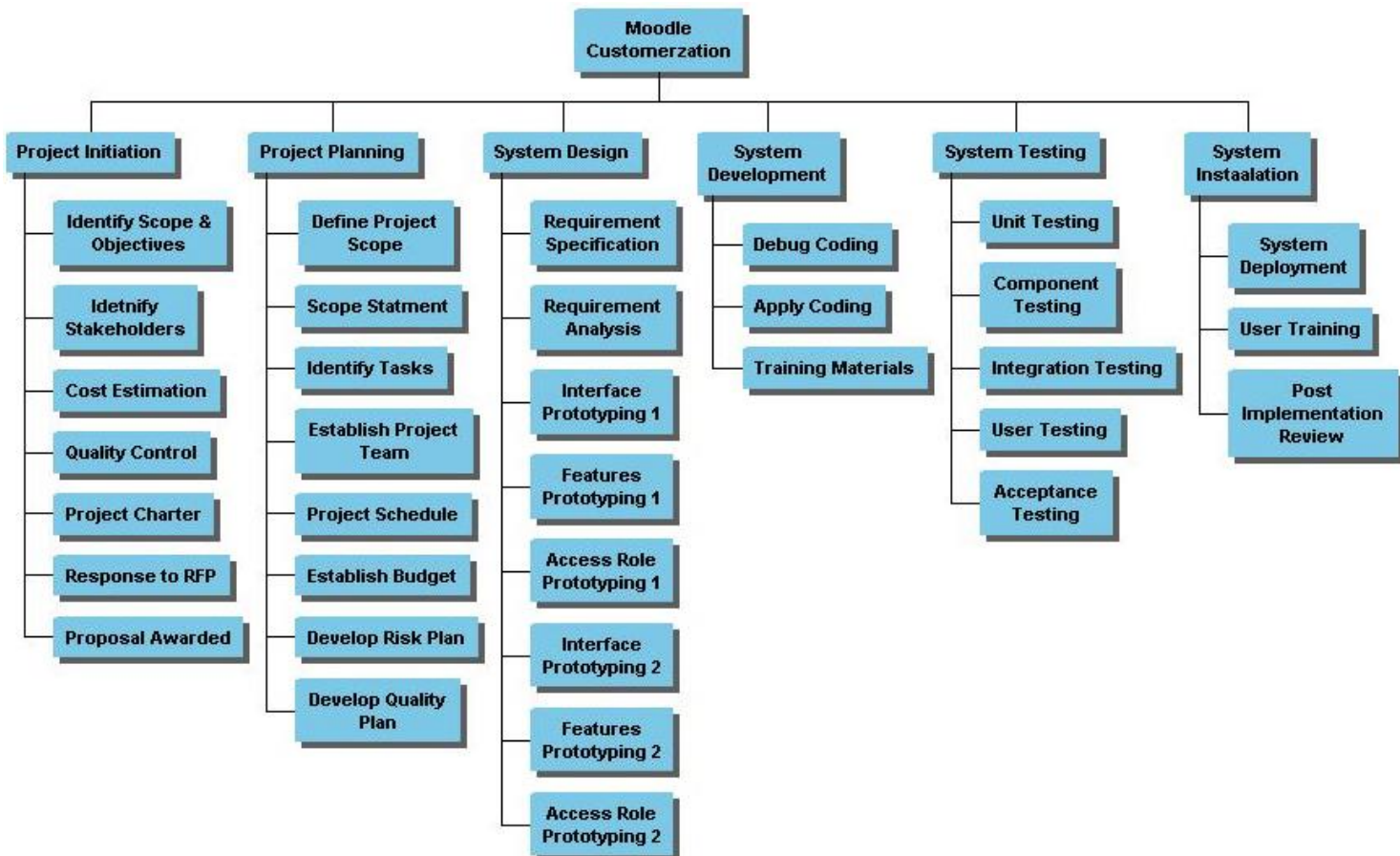
- Herramientas
 - EDT
 - Tablero KanBan
 - Project Libre
 - Plan de Riesgos



Project
Libre™

Planeación

➔ Estructura de desglose de trabajo (EDT)



Planeación

➤ Tablero KanBan

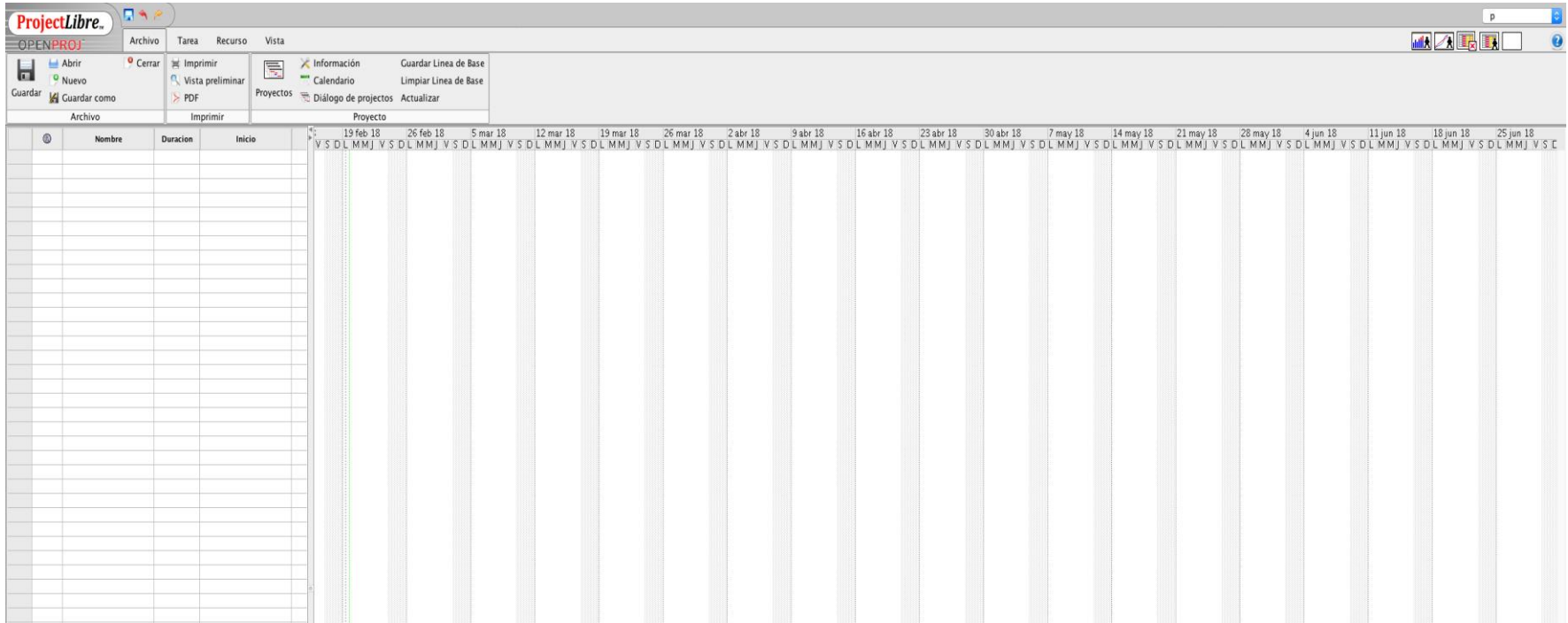
The screenshot displays a Kanban board interface. At the top left is the MeisterTask logo. To its right are navigation icons: a rocket icon labeled 'REDIS' with a dropdown arrow, and an information icon 'i'. The board consists of three columns:

- Abrir (2 items):** A teal header with a lightbulb icon. It contains two task cards: 'Cotización de infraestructura' and 'Comparativa de soluciones'. A plus sign is at the bottom.
- En Progreso (1 item):** A blue header with a pulse icon. It contains one task card: 'Pruebas de integración'. A plus sign is at the bottom.
- Listo (1 item):** A purple header with a checkmark icon. It contains one task card: 'Adquisición de equipo de red'. A plus sign is at the bottom.

Each column header also features a count of items and a plus sign on the far right of the 'Listo' column.

Planeación

➔ ProjectLibre



Planeación

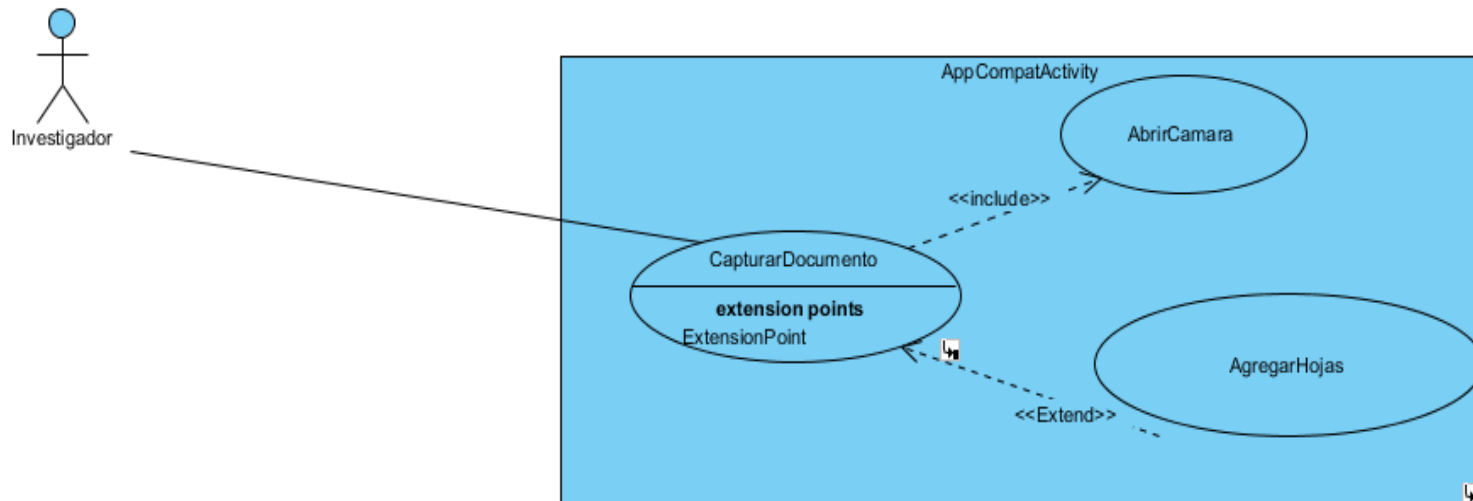
➔ Plan de Riesgos

ID	Actividad	Descripción	Causa Raíz	Entregables afectados	Probabilidad de ocurrencia	Impacto	Respuesta Planificada	Responsable de la respuesta
R1	Servidor	Atraso en la entrega por falta de disponibilidad	Atenerse a un solo proveedor	Equipo Servidor	50 %	Alto	Considerar más de un proveedor y saber que este puede dar respuesta en tiempo y costo. (Previo acuerdo)	PM
R2	Contratación de servicio	El proveedor de Data Center no puede otorgar el servicio	Atenerse a un solo proveedor	DataCenter	10%	Alto	Considerar más de un proveedor.	PM
R3	Proceso de compras	Proceso para realizar la adquisición de bienes automatizado.	Depto. Compras no es colaborativo.	Módulo de Compras.	40%	Alto	Dirección gestionar la entrega de documentación o un responsable a cargo. Que conozca todo el proceso.	IS.

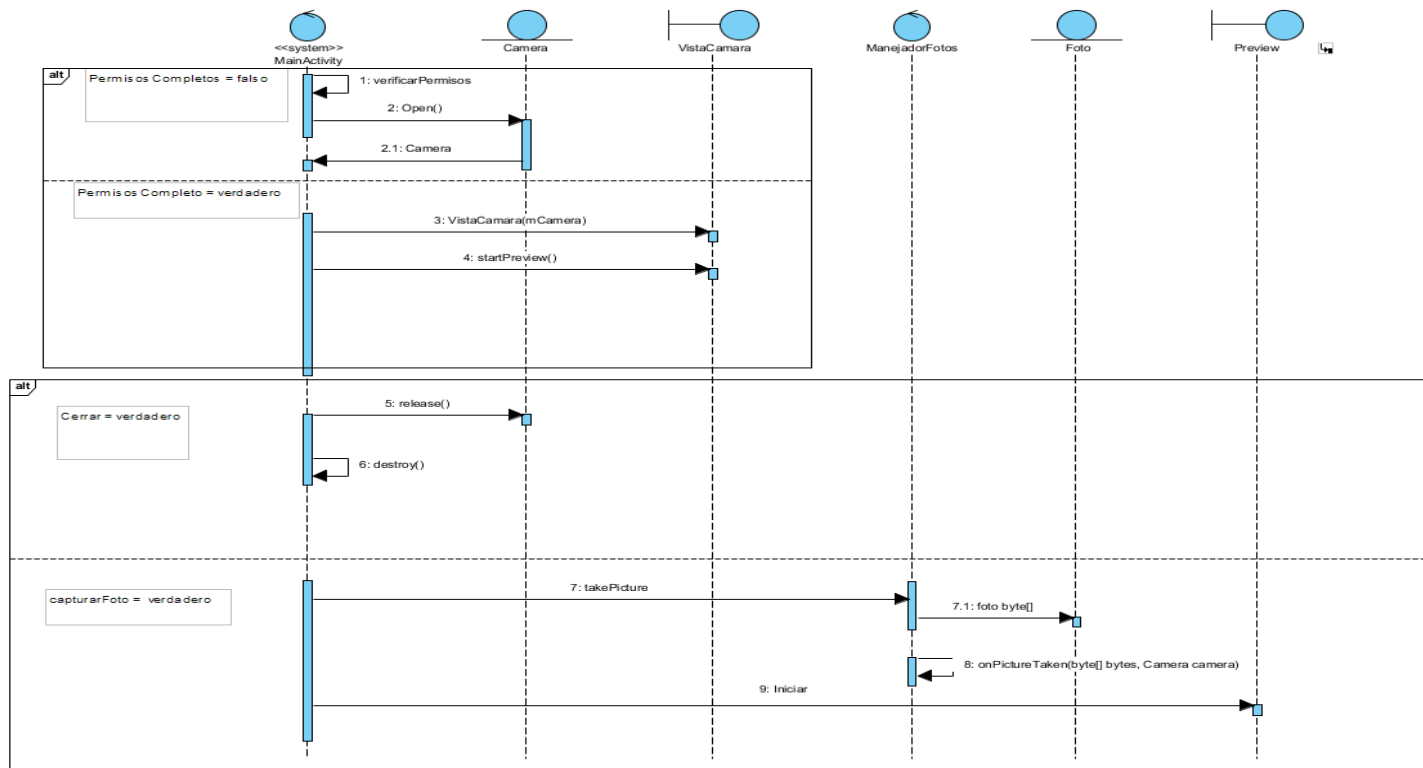
- Herramientas
 - UML
 - Diagramas de casos de uso
 - Diagramas de secuencia



➤ Diagrama de casos de uso



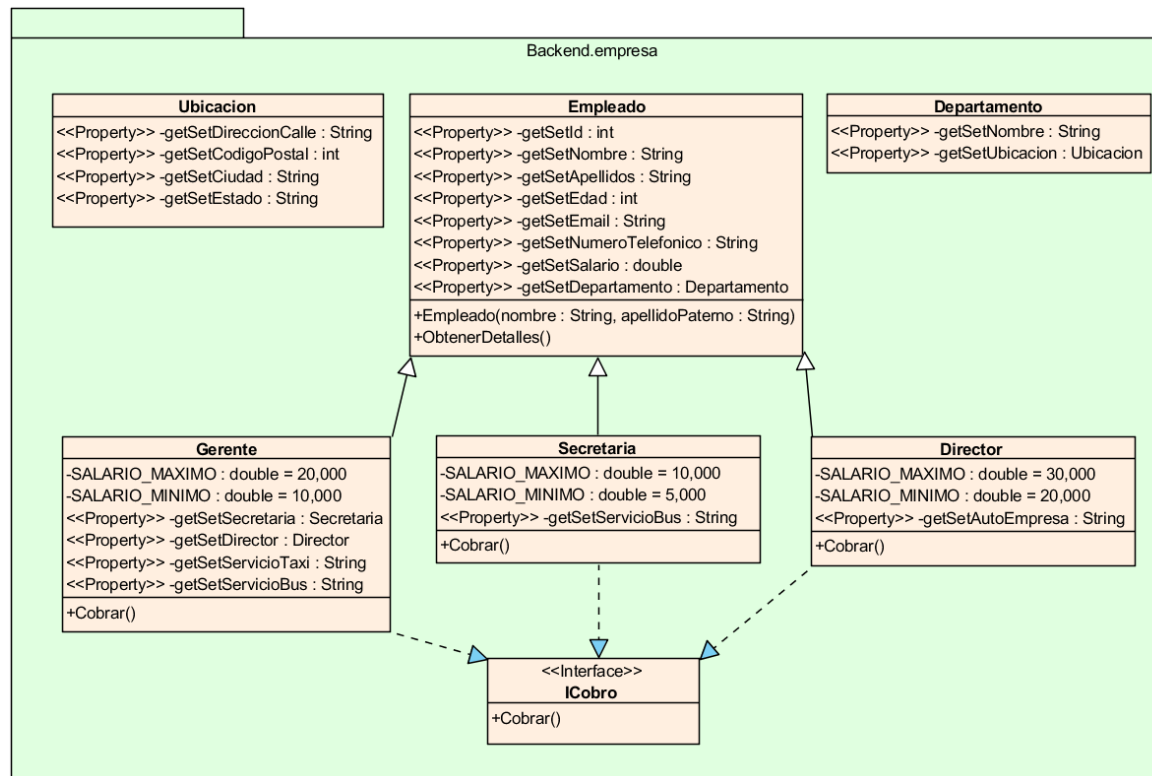
➔ Diagramas de secuencia



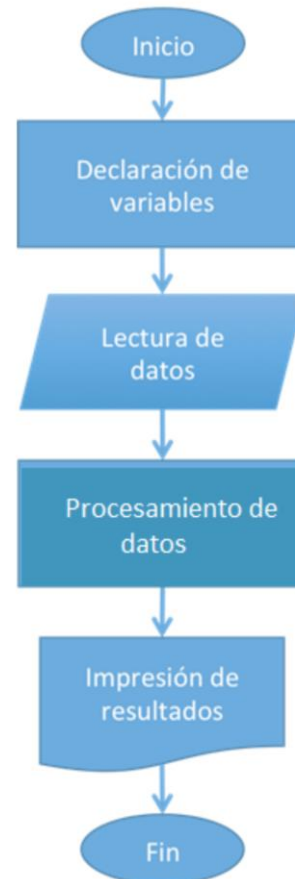
- Herramientas
 - Diagrama de clases
 - Diagramas de flujo
 - Diagrama Entidad relación



➔ Diagrama de clases

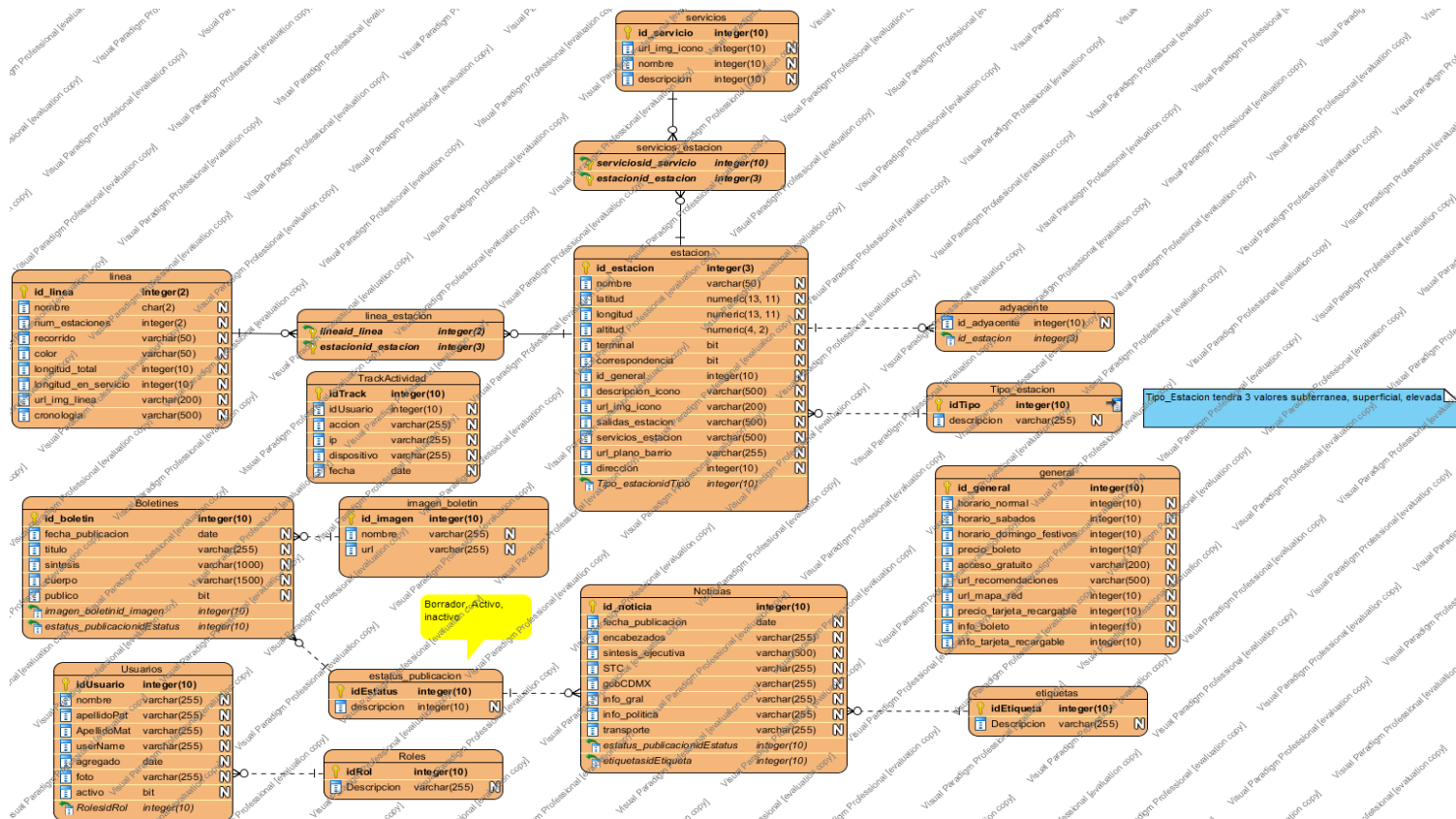


➤ Diagrama de flujo



Diseño

DER



Construcción

➔ Normalizar código

```
class AES extends CI_Controller {
    const M_CBC = 'cbc';
    const M_CFB = 'cfb';
    const M_ECB = 'ecb';
    const M_NOFB = 'nofb';
    const M_OFB = 'ofb';
    const M_STREAM = 'stream';

    protected $key;
    protected $cipher;
    protected $data;
    protected $mode;
    protected $IV;

    /**
     *
     * @param type $data
     * @param type $key
     * @param type $blockSize
     * @param type $mode
     */
    function __construct($data = null, $key = null, $blockSize = null, $mode = null) {
        $this->setData($data);
        $this->setKey($key);
        $this->setBlockSize($blockSize);
        $this->setMode($mode);
        $this->setIV("");
    }
}
```

Integración

➤ Memoria técnica

➤ Contratos

Nombre:	scan_face()
CU:	CU001
Referencias cruzadas:	RF3.1, RF3.3, RF4.1, Contrato Controladores documento_scan()
Clase:	document_model (Camaraweb/models/document_model.php)
Responsabilidades:	Es el modelo que manda la instrucción a la cámara para captar el rostro del usuario.
Notas:	
Entrada:	Imagen del rostro.
Salida:	Patrones faciales en código.
Precondiciones:	El equipo debe contar con una cámara web interna o externa y con suficiente luz para reconocer los patrones faciales.
Postcondiciones:	El equipo debe contar con la memoria suficiente para almacenar los datos obtenidos del escáner.

Conclusión

- Invertir en el ciclo de vida del Software
(Sustentabilidad a tus proyectos)
- Respeto a nuestro trabajo





Universidad Nacional
Autónoma de México

RED UNIVERSITARIA DE COLABORACIÓN
EN INGENIERÍA DE SOFTWARE Y BASES DE DATOS



Preguntas?

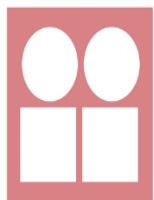
Gracias

Ing. Julio Alfonso De León Razo

Contacto:

Correo:

jdeleonr@iingen.unam.mx



INSTITUTO
DE INGENIERÍA
UNAM