Proyecto Monitoreo

de

Humedad

Manual Tecnico

Hecho por Johan Castro / Harol Olivera / Elkin Alexis

Proyecto monitoreo de humedad UML 1

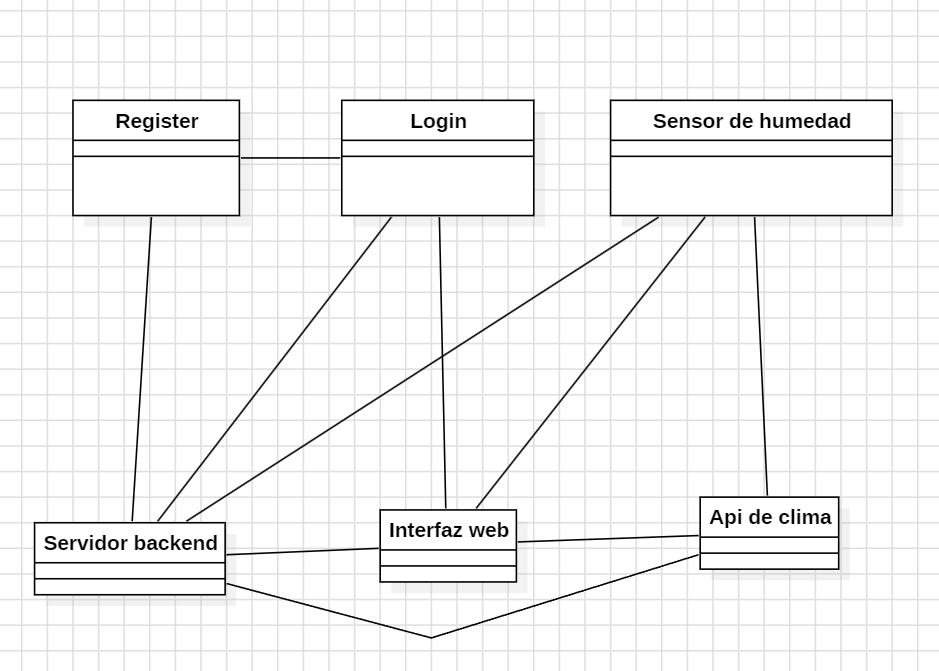
Hecho por Elkin Alexis / Johan Castro / Harol Olivera

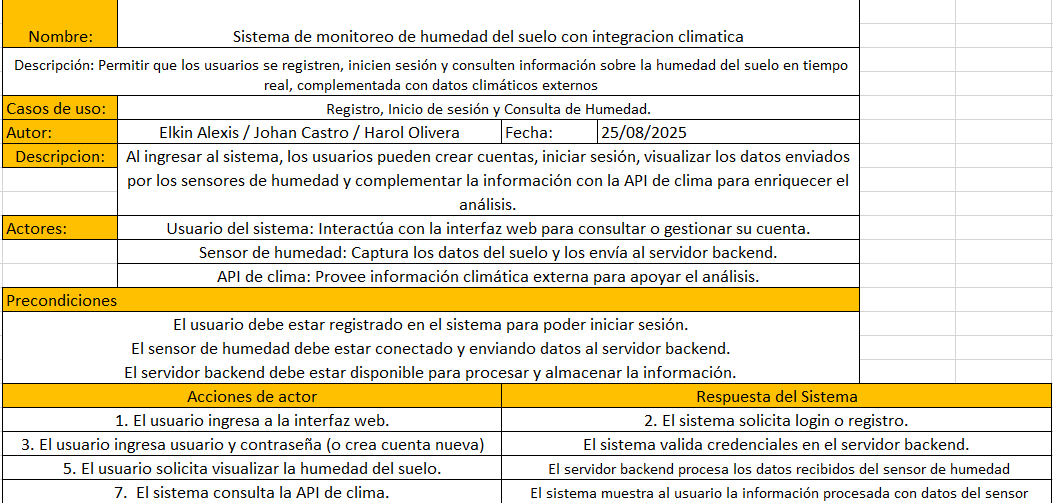
## Contextualizacion

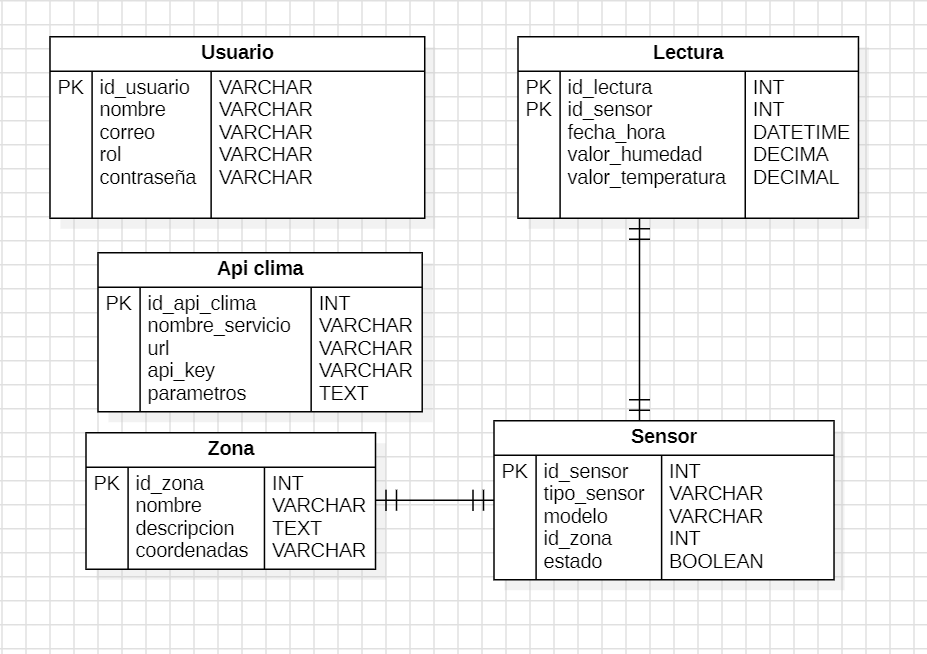
* **Register/Login**: Permiten que los usuarios creen cuentas y accedan de forma segura al sistema, conectándose al backend para validar y almacenar la información.
* **Sensor de humedad**: Mide la humedad del suelo y envía los datos al servidor backend, para ser procesados y almacenados, además de enviar información relevante al dashboard y, si es necesario, consumir datos externos (como el clima) para enriquecer el análisis.
* **Servidor backend**: Es el núcleo lógico que recibe información de los sensores y de los módulos de usuario (register/login); procesa, almacena y distribuye datos tanto a la interfaz web como hacia o desde la API de clima.
* **Interfaz web**: Es el dashboard donde el usuario visualiza y analiza los datos. Se conecta al backend para recibir información y gestionarla. Consulta tanto registros internos como datos de la API de clima para mostrar análisis completos.
* **API de clima**: Permite obtener información climática externa, la cual puede ser usada para comparar con las mediciones, generar alertas o recomendaciones. El backend y la interfaz consultan esta API para enriquecer la experiencia del usuario.

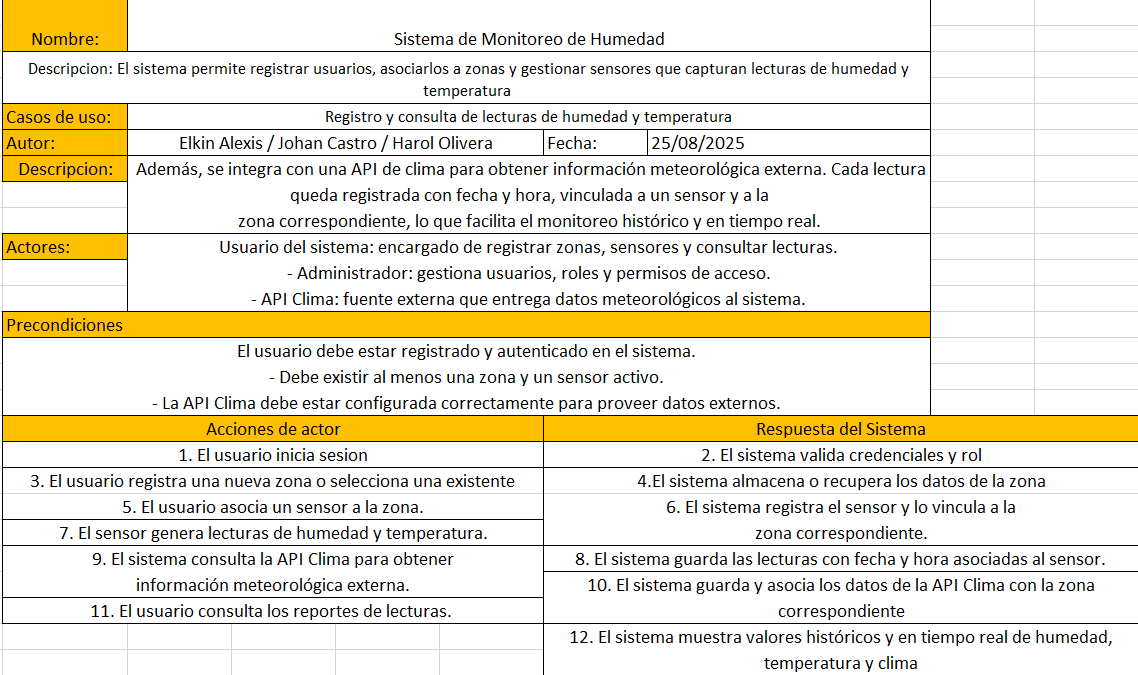
Las conexiones existen para garantizar que:

* Toda autenticación y almacenamiento sea centralizado (backend).
* Los sensores solo interactúan con lo estrictamente necesario (backend y clima).
* El dashboard pueda usar toda la información del sistema, tanto interna como externa, para análisis y visualización.





Proyecto Monitoreo de humedad UML 2



La base de datos soporta la aplicación almacenando usuarios, zonas, sensores, lecturas y datos climáticos de forma organizada y vinculada. Las entidades fueron seleccionadas porque representan los elementos clave del sistema: usuarios para controlar accesos y roles, zonas para organizar geográficamente los sensores, sensores como dispositivos físicos que generan datos, lecturas con los registros históricos de humedad y temperatura y API Clima para integrar información meteorológica externa.

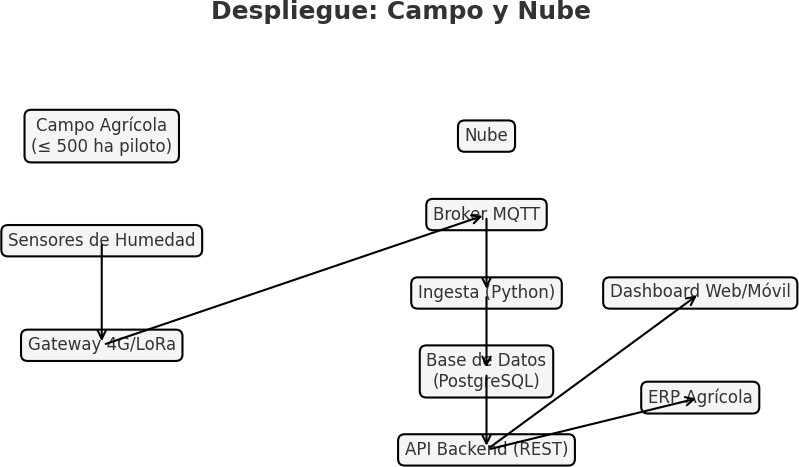
Las relaciones reflejan las necesidades reales:

Una zona puede tener muchos sensores. Un sensor puede generar muchas lecturas.

Usuarios pueden asignarse a varias zonas (y viceversa) para gestión. La API de clima complementa datos para análisis.

Proyecto Monitoreo de humedad UML 3

**Diagrama de Despliegue**



El despliegue separa el entorno de campo (sensores y gateway con conectividad LoRa/4G) de la nube (broker MQTT, ingesta, base de datos y API). El dashboard web/móvil y el ERP consumen la API de forma segura.

## Decisiones de Arquitectura e Integración

* + Tipo de arquitectura: orientación a servicios (microservicios ligeros). Elegimos separar cuatro dominios: (1) Ingesta, (2) Almacenamiento/Consulta, (3) Alertas/Notificaciones, (4) Integraciones. Esto facilita escalar de forma independiente la ingesta y las alertas ante picos, sin complejizar en exceso el despliegue.
  + Justificación: el piloto está limitado a ~500 hectáreas; con microservicios ligeros aseguramos calidad de datos, baja latencia en el dashboard (< 2 s) y mantenemos independencia tecnológica. Si en el futuro se expande, cada servicio puede escalar horizontalmente.
  + Tecnologías: Python para ingesta/análisis; JavaScript para el dashboard; PostgreSQL para almacenamiento relacional y series temporales simples. MQTT como protocolo de mensajería eficiente en enlaces de campo.
  + Integraciones: ERP agrícola vía webhooks/REST; notificaciones por SMS/Email/WhatsApp. Seguridad con HTTPS y tokens; control de acceso por perfiles (Administrador y Agricultor).
  + Operación y calidad: pruebas unitarias, integrales y de aceptación; revisión estática de código. Monitoreo de colas MQTT, métricas de ingesta, tiempos de respuesta y tasa de alertas.

Proyecto Monitoreo de Humedad UML 4

Nombre: Sistema de Monitoreo Agrícola IoT

Descripción: Diagrama que representa los principales componentes del sistema y su interacción, para

la recolección, análisis y visualización de datos de humedad del suelo.

Caso de uso: Diseño e implementación de un sistema de monitoreo de humedad en cultivos con

sensores IoT.

Autor: Equipo de desarrollo – Sprint 2

Fecha: 15/08/2025

Descripción detallada:

Este diagrama de componentes muestra la estructura básica del sistema:

• Sensores IoT: Dispositivos encargados de recolectar las mediciones de humedad en campo.

• Backend (Lógica de Negocio): Procesa los datos recibidos, ejecuta reglas de negocio y gestiona las

alertas.

• Base de Datos PostgreSQL: Almacena lecturas, usuarios y parámetros de control, garantizando

persistencia.

• Dashboard Web: Interfaz gráfica para que los agricultores consulten datos y gestionen parámetros de

forma sencilla.

Actores: Agricultores y administradores del sistema.

Precondiciones:

• Los sensores deben estar conectados correctamente a la red IoT.

• El backend debe tener acceso a la base de datos y estar disponible en el servidor.

Flujo Normal:

1. Los sensores capturan la humedad en tiempo real.

2. El backend recibe y procesa las lecturas.

3. El backend almacena la información en la base de datos.

4. El dashboard consulta al backend y muestra los resultados al usuario.

Flujo Alternativo:

• Si los sensores fallan en el envío, el backend genera alertas de error.

• Si la base de datos no está disponible, el backend almacena temporalmente los datos en caché.

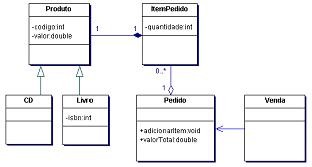
Postcondiciones:

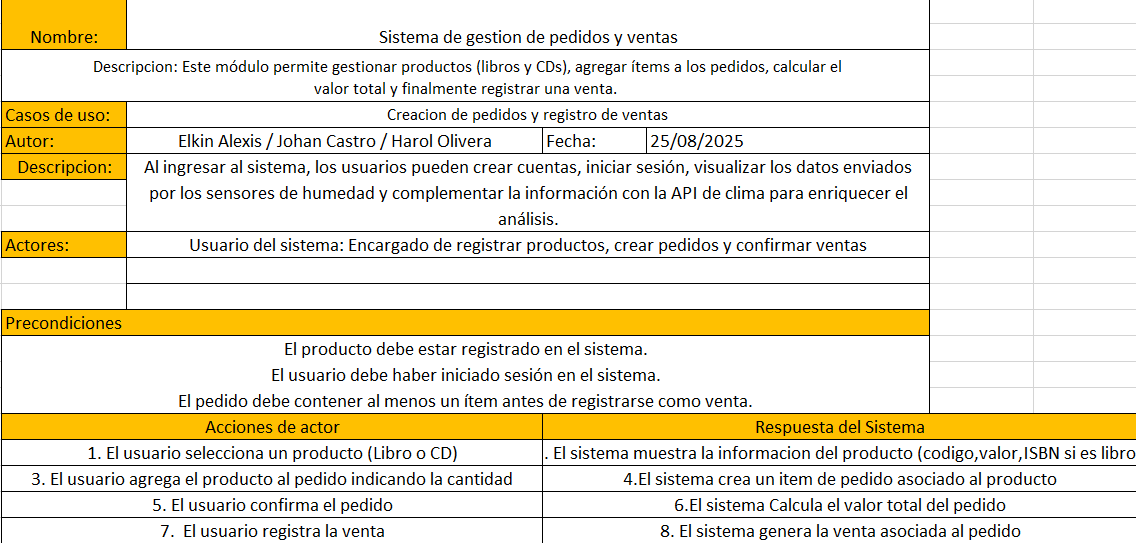
• Los datos quedan registrados en la base de datos.

• El dashboard refleja la última información disponible.

Frecuencia: Constante (cada lectura de los sensores).

Tiempo de Respuesta: Menor a 5 segundos.





Proyecto de Monitoreo de Humedad UML 5

Nombre: Sistema de Monitoreo Agrícola IoT

Descripción: Diagrama que representa las entidades principales del sistema y sus relaciones,

sirviendo de base para el modelo de datos que soporta el almacenamiento y análisis de la información

recolectada por los sensores.

Caso de uso: Identificación de entidades y relaciones para la construcción del modelo de datos.

Autor: Equipo de desarrollo – Sprint 1

Fecha: 11/08/2025

Entidades definidas:

• Usuario: Representa a los actores del sistema que acceden a la plataforma.

• Registro: Contiene información complementaria de cada usuario.

• Riego: Almacena las solicitudes de riego generadas.

• Funcionalidad: Registra datos ambientales y de operación asociados al monitoreo.

Relaciones principales:

• Un Usuario puede estar vinculado a múltiples registros en la tabla Registro (1:N).

• Cada Registro corresponde de manera directa a un único Usuario (1:1).

• Una solicitud en la tabla Riego se relaciona con la tabla Funcionalidad (1:N).

• El campo de enlace asegura coherencia temporal entre Riego y Funcionalidad.

Claves primarias y foráneas:

• Se definieron claves primarias en cada entidad para garantizar unicidad.

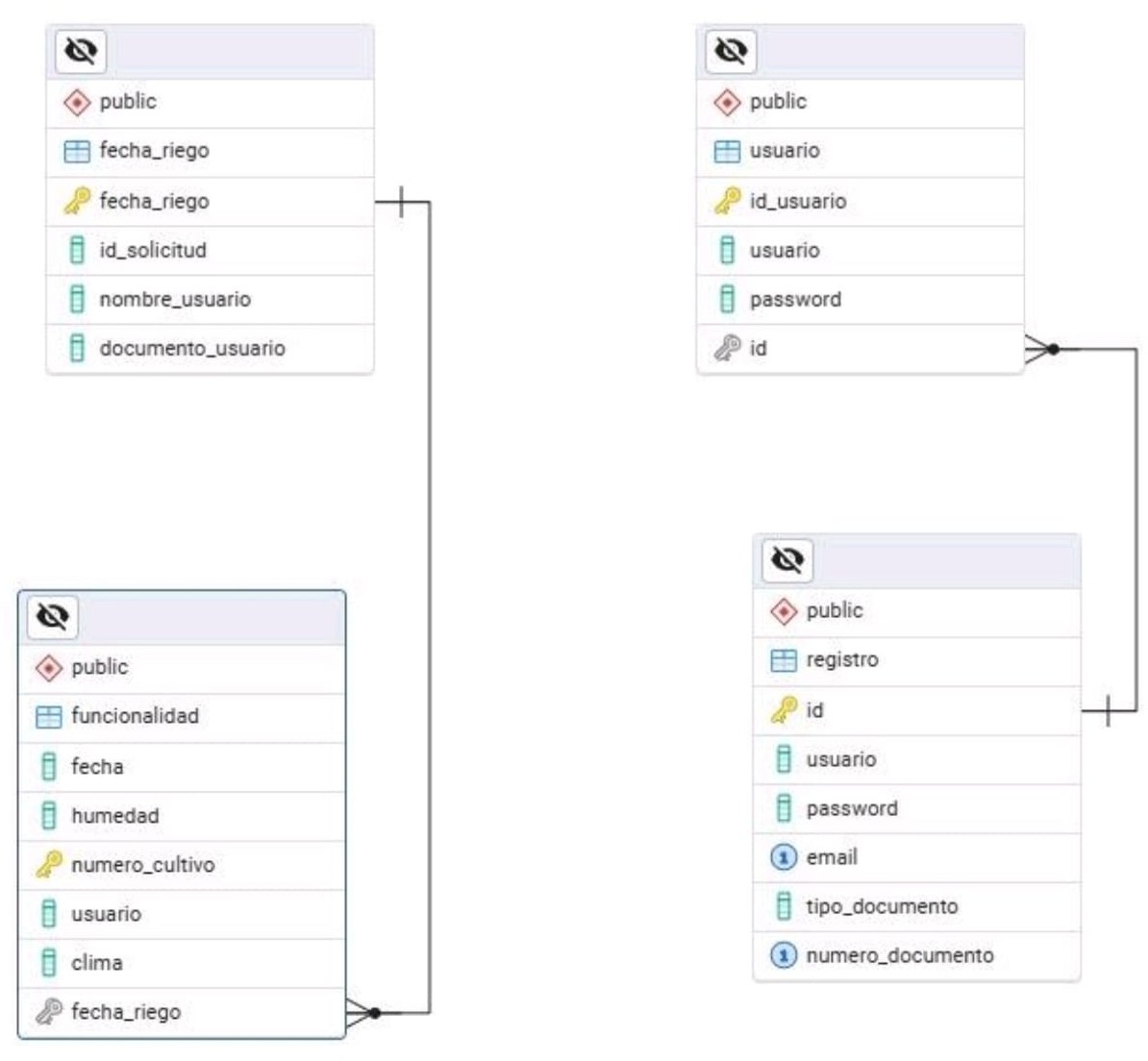
• Se establecieron claves foráneas, como la de Funcionalidad hacia Riego, para mantener integridad

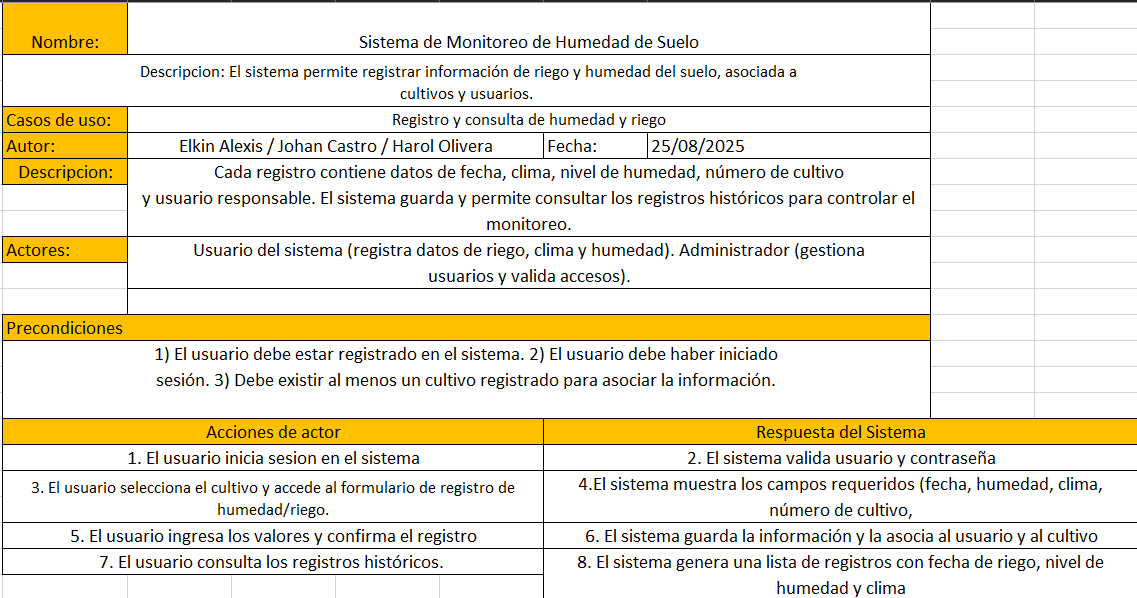
referencial.

Postcondiciones:

• El modelo de datos garantiza coherencia y organización para soportar la lógica de negocio.

Frecuencia: Uso constante en cada transacción del sistema.

Tiempo de Respuesta: Menor a 3 segundos en consultas y registros.

****