Información del Documento

|  |  |
| --- | --- |
| **Resumen:** | El presente documento describe todos los aspectos relacionados con una Arquitectura de la Solución para el sistema de cálculo REMETFU (Registro Metropolitano de Fauna Urbana) |
| **Alcance:** | El presente documento contiene conceptos clave en torno al desarrollo de una Arquitectura de la Solución, la definición de Dominios de Arquitectura, Diagramas o Vistas de Arquitectura, Principio y Declaraciones de Arquitectura. |
| **Audiencia:** | * Arquitectos de Software * Analistas e Ingenieros de Desarrollo del Proyecto * Gerente de Proyecto * Direcciones y Jefaturas Tecnológicas |

Historial del Documento

|  |  |  |
| --- | --- | --- |
| **Versión/Fecha** | **Autor** | **Descripción** |
| 1.0.0 / 2022-08-15 | Segundo Cisneros | Versión Inicial del documento |
|  |  |  |

Revisión del Documento

|  |  |  |
| --- | --- | --- |
| **Versión/Fecha** | **Autor** | **Descripción** |
|  |  |  |
|  |  |  |

CONTENIDO

[1. ANTECEDENTES Y JUSTIFICACIÓN 3](#_Toc20795)

[2. REQUERIMIENTOS FUNCIONALES 3](#_Toc20301)

[3. REQUERIMIENTOS NO FUNCIONALES 4](#_Toc21820)

[4. ARQUITECTURA BASADA EN SERVICIOS 5](#_Toc14576)

[5. VISTAS DE ARQUITECTURA 5](#_Toc2941)

[5.1 Diagramas de Casos de Uso 5](#_Toc16456)

[5.1.1 Administración de usuarios 5](#_Toc27717)

[5.1.2 Parametrización 6](#_Toc23073)

[5.1.3 Cálculo 6](#_Toc2119)

[5.1.4 Generación de órdenes de pago 7](#_Toc28436)

[5.1.5 Consultas 7](#_Toc7586)

[5.2 Diagramas de Clases 7](#_Toc1286)

[5.3 Diagramas Entidad Relación 7](#_Toc25059)

[5.4 Diagramas de Actividades 8](#_Toc3490)

[PROCESOS DEL SITEMA COD (SICOD) 8](#_Toc4366)

[5.4.1 Administración de usuarios 8](#_Toc18881)

[5.4.2 Parametrización 8](#_Toc4600)

[5.4.3 Cálculo 8](#_Toc28880)

[5.4.4 Generación de Ordenes de pago 8](#_Toc17023)

[5.4.5 Consultas 8](#_Toc1211)

[5.6 Diagrama de N Capas 8](#_Toc21084)

[6. CONCLUSIONES 10](#_Toc15850)

[7. RECOMENDACIONES 10](#_Toc2464)

[8. GLOSARIO 10](#_Toc6215)

# ANTECEDENTES Y JUSTIFICACIÓN

|  |
| --- |
| Se expide la Ordenanza Metropolitana 019-2020, con el fin de generar una base de datos de animales y tutores registrados, información de establecimientos prestadores de servicios selacionados a la fauna urbana (consultorios, clínicas, hospitales, centros de acogida, albergues, criaderos, peluquería, tiendas, escuela de adiestramiento, entre otras), animales, personas sanciones por causales de maltrato animal, animales inscritos fallecidos y alertas por pérdidas o extravíos de animales de compañía.  La Ordenanza Metropolitana N.- 019-2020 – Disposición Transitoria cuarta cita: “En el término de noventa días contados desde la publicación en el Registro Oficial de la presente ordenanza metropolitana sustitutiva, la Dirección Metropolitana de Informática de conformidad con el régimen jurídico aplicable procederá a la elaboración de la herramienta tecnológica, base del Registro Metropolitano de Fauna Urbana denominado REMETFU debiendo contar con los requerimientos de la Secretaría Metropolitana de Salud” |

Según normativa de control interna, modelo de gestión de la DMI, y la Políticas de Tecnología, para todo sistema se requiere una contraparte funcional, la cual al momento se encuentra definida que es la Secretaría de Salud y el área de Fauna Urbana.

Para la ejecución del mencionado desarrollo es requerimiento indispensable tener un Diseño Arquitectónico de la Solución que identifique los componentes lógicos y físicos, internos y externos que son necesarios para la implementación adecuada del Sistema REMETFU.

Finalmente es necesario resaltar que para esta versión del presente documento (1.0.0) se ha trabajado meramente en base al requerimiento de software enviado mediante SITRA GADDMQ-SS-2022-0184-OF, y denominado RSW\_REMETFU\_2022\_001.

# REQUERIMIENTOS FUNCIONALES

|  |  |  |  |
| --- | --- | --- | --- |
| **MÓDULO** | **TIPO** | **CÓDIGO** | **RESUMEN** |
| **INTEGRACIÓN CON PAM** | INTERNET | REQ # 1 | Se integrará con el PAM para el acceso al Sistema REMETFU desde el internet para los usuarios ciudadanos |
| **ADMINISTRACIÓN DEL SISTEMA** | INTRANET | REQ # 2 | Desarrollar la funcionalidad para el manejo de opciones, roles y perfiles |
| **ADMINISTRACION DE USUARIOS** | INTRANET | REQ # 3 | Desarrollar la funcionalidad para la Administración de usuarios para el Aplicativo en lo que refiere al REMETFU, integrándolo al proyecto de Autenticación Centralizada KEYCLOAK |
| **PARAMETRIZACION** | INTRANET | REQ # 4 | Administración de variables que intervienen en el sistema REMETFU y que podrán ser administradas por el área funcional |
| **REGISTRO DE VETERINARIOS** | INTERNET | REQ # 5 | Implementar el módulo para el registro, revisión y aprobación de veterinarios. |
| **REGISTRO DE MICROCHIPS** | INTERNET | REQ # 6 | Implementar el módulo para el registro de los microchips colocados a los animales de compañía |
| **REGISTRO DE ESPECIALISTA EN COMPORTAMIENTO** | INTERNET | REQ # 7 | Implementar el módulo para el registro, actualización, revisión y aprobación de prestadores de servicio: Especialista en comportamiento – conducta animal. |
| **REGITRO DE TUTORES** | INTERNET | REQ # 8 | Implementar el módulo para registro de tutores |
| **REGISTRO DE ANIMALES DE COMPAÑÍA** | INTERNET | REQ # 9 | Implementar el módulo para el pre registro y registro de animales de compañía con tutor |
| **REGISTRO DE ANIMALES DE COMPAÑÍA SIN TUTOR** | INTRANET | REQ # 10 | Implementar el módulo para el pre registro y registro de animales de compañía sin tutor o tutores en condición de calle. |
| **REGISTRO DEL COMITÉ DE ETICA** | INTRANET | REQ # 11 | Implementar el módulo de comité de Ética |
| **CONSULTAS** | INTRANET  INTERNET | REQ # 12 | Implementación del módulo de consulta del estado de los trámites ingresados en el Sistema REMETFU |

*Tabla 1: Requerimientos funcionales*

# REQUERIMIENTOS NO FUNCIONALES

|  |  |  |
| --- | --- | --- |
| **GRUPO** | **CÓDIGO** | **REQUERIMIENTO** |
| **ADMINISTRACIÓN DE USUARIOS** | RNF001 | Integración Keyclock mediante API GATEWAY para la adminsitración de usuarios en el aplicativo |
| **MANEJO DE CONTENEDORES** | RNF002 | Aplicativo funcional en contenedores |
| **SEGURIDAD** | RNF003 |  |
| **DISPONIBILIDAD** | RNF004 | Toda la solución debe tener una disponibilidad 8x5 por lo que se sugiere que se disponga de Alta Disponibilidad a nivel de la infraestructura. |
| **MANTENIBILIDAD** | RNF005 | La solución debe contener pistas de auditoría de la siguiente manera:   * Cuatro columnas en todas las tablas transaccionales que muestre la fecha, hora, usuario y máquina donde se creó el registro de la tabla. * Tablas complementarias a ciertas tablas transaccionales para almacenar el log de los servicios consumidos con fecha hora de consumo y respuesta. Estas tablas complementarias de log deben ser respaldadas y borradas cada 7 años. |
| **MANTENIBILIDAD** | RNF006 | La solución deber tener un log de errores diseñado de la siguiente manera:   * Log (archivo texto) de excepciones que almacene:   + Fecha de la excepción   + Hora de la excepción   + Máquina donde se dio la excepción   + Clase donde se produjo la excepción   + Mensaje de la excepción * Archivo de Excepciones que se almacene en una ruta específica tanto en del lado Servidor, formará parte del nombre del archivo de log la fecha y tendrá la misma estructura de la tabla de Excepciones. |
| **ESCALABILIDAD** | RNF007 | La solución debe tener la posibilidad de escalar a nivel horizontal de manera obligatoria por lo que se debe desarrollar de manera que pueda ser desplegada copias de los componentes de la solución en varios servidores que funciones como un clúster. Ante esto se sugiere que se utilice un esquema de balanceo de carga entre los servidores donde se desplegará la solución. |
| **ESCALABILIDAD** | RNF008 | La solución debe tener escalabilidad vertical por lo que se sugiere que la firma de todos los métodos de la solución utilicen entidades y listas de entidades en lugar de parámetros individuales con el fin de que cualquier nueva versión del aplicativo no se vea afectada por el cambio en la firma de los métodos y que la nueva funcionalidad se desplegada con facilidad porque su afectación es solo a nivel interno de cada método. |
| **USABILIDAD** | RNF009 | La solución debe seguir el esquema de diseño de interfaces acorde a los estándares de Imagen de la DMI y secretaría de comunicación. |
| **INTEROPERABILIDAD** | RNF010 | La solución debe enlazarse con otros aplicativos mediante un componente que actúe como middleware. En este caso se sugiere que se haga uso de un ESB y que sea un ESB del mercado; preferentemente de tipo Open Source. Esto basado en el análisis comparativo que se entregó como parte del Informe de Factibilidad de Arquitectura. |
| **VERIFICABILIDAD** | RNF010 | La solución debe ser construida de tal forma que favorezca el proceso de Integración Continua con ejecución de pruebas unitarias y de integración. |

*Tabla 2: Requerimientos no funcionales*

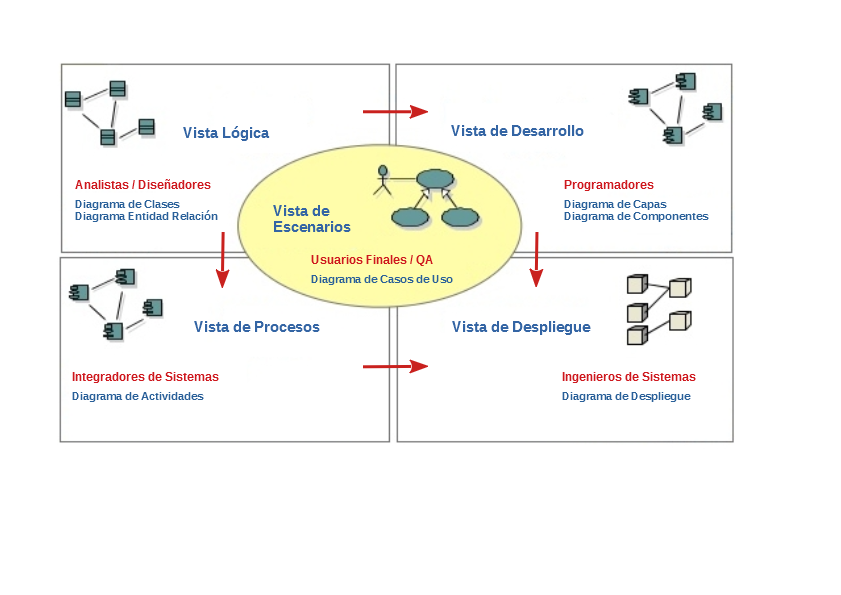
# ARQUITECTURA BASADA EN SERVICIOS

Para el proceso de desarrollo, adquisición y mantenimiento de aplicaciones de software dentro del GAD del DMQ, es necesario considerar que su arquitectura se desarrolle de manera estándar y apegada a un conjunto de buenas prácticas o marcos de referencia ya probados y de uso generalizado en ambientes tecnológicos competitivos, con el objetivo de permitir una rápida respuesta a un ambiente tecnológico cambiante y demandante durante la construcción de la solución, se utilizará la arquitectura basada en servicios; la cual abarca una serie de componentes lógicos que aseguran la construcción de soluciones desacopladas, escalables, seguras y fáciles de gobernar.

En el caso del Sistema REMETFU (Registro Metropolitano de Fauna Urbana), va a tener servicios externos SOAP (de los sistemas que así los expongan), y servicios internos que en base al avance de la tecnología van a ser REST.

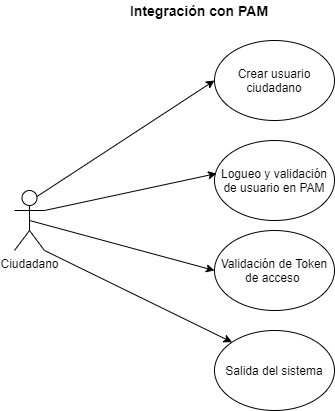
# VISTAS DE ARQUITECTURA

A continuación, se muestran los diagramas de Lenguaje de Modelado Unificado (UML) que forman parte de cada una de las vistas requeridas por el modelo 4+1.

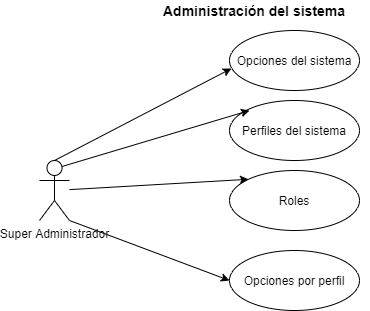
**

## 5.1 Vista de Escenarios: Diagramas de Casos de Uso

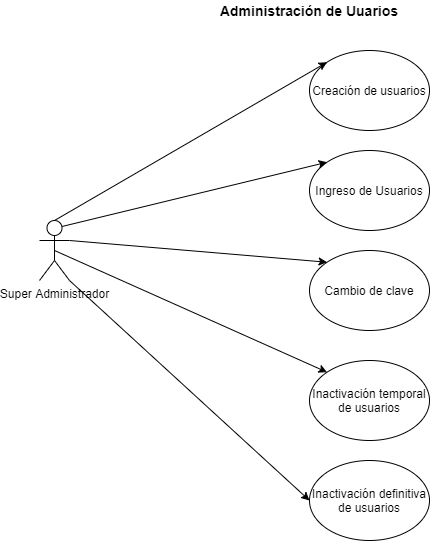
### 5.1.1 Integración con PAM



### 5.1.2 Administración del Sistema

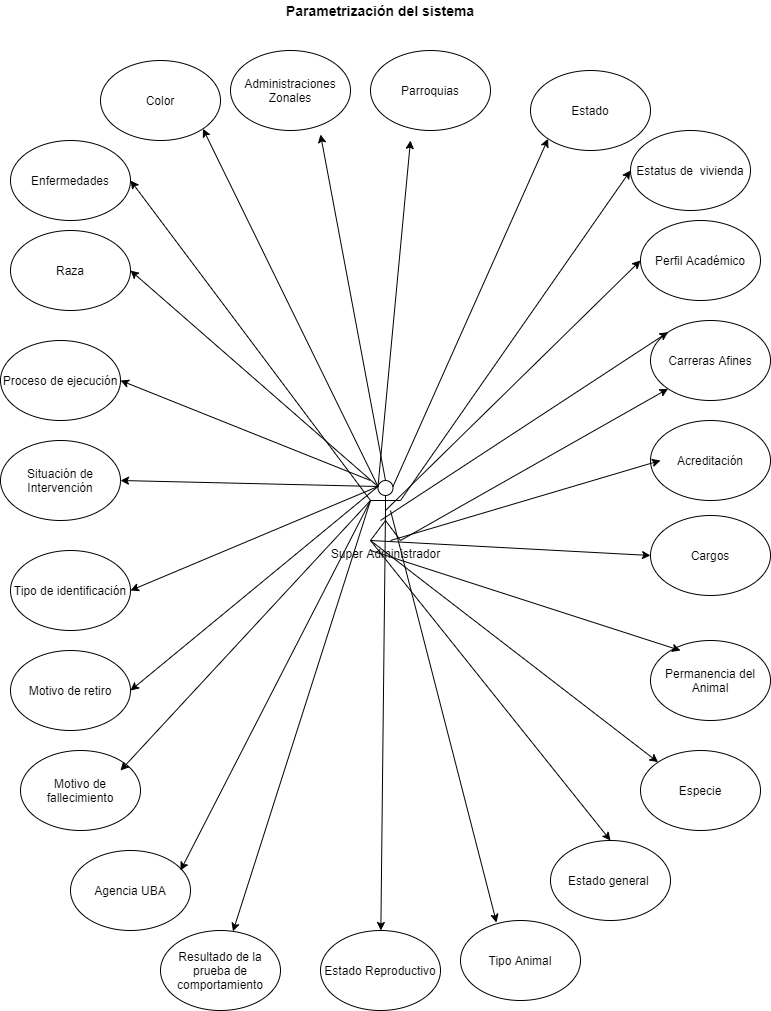


### 5.1.1 Administración de Usuarios

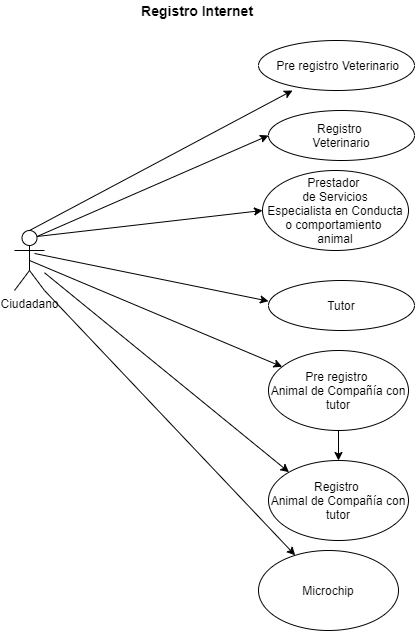


%3CmxGraphModel%3E%3Croot%3E%3CmxCell%20id%3D%220%22%2F%3E%3CmxCell%20id%3D%221%22%20parent%3D%220%22%2F%3E%3CmxCell%20id%3D%222%22%20value%3D%22Super%20Administrador%22%20style%3D%22shape%3DumlActor%3BverticalLabelPosition%3Dbottom%3BlabelBackgroundColor%3D%23ffffff%3BverticalAlign%3Dtop%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22100%22%20y%3D%22330%22%20width%3D%2250%22%20height%3D%2270%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%223%22%20value%3D%22Creaci%C3%B3n%20de%20Usuarios%22%20style%3D%22ellipse%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22250%22%20y%3D%2290%22%20width%3D%22140%22%20height%3D%2270%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%224%22%20value%3D%22Bloqueo%20de%20Usuarios%22%20style%3D%22ellipse%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22380%22%20y%3D%22150%22%20width%3D%22140%22%20height%3D%2270%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%225%22%20value%3D%22Creaci%C3%B3n%20de%20Roles%22%20style%3D%22ellipse%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22460%22%20y%3D%22220%22%20width%3D%22140%22%20height%3D%2270%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%226%22%20value%3D%22Asignaci%C3%B3n%20de%20Roles%22%20style%3D%22ellipse%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22500%22%20y%3D%22310%22%20width%3D%22140%22%20height%3D%2270%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%227%22%20value%3D%22Creaci%C3%B3n%20de%20Opciones%22%20style%3D%22ellipse%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22400%22%20y%3D%22379%22%20width%3D%22140%22%20height%3D%2270%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%228%22%20value%3D%22Asignaci%C3%B3n%20de%20Opciones%20al%20Rol%22%20style%3D%22ellipse%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22280%22%20y%3D%22449%22%20width%3D%22140%22%20height%3D%2270%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%229%22%20value%3D%22%22%20style%3D%22endArrow%3Dclassic%3Bhtml%3D1%3B%22%20edge%3D%221%22%20source%3D%222%22%20parent%3D%221%22%3E%3CmxGeometry%20width%3D%2250%22%20height%3D%2250%22%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22410%22%20y%3D%22320%22%20as%3D%22sourcePoint%22%2F%3E%3CmxPoint%20x%3D%22460%22%20y%3D%22270%22%20as%3D%22targetPoint%22%2F%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2210%22%20value%3D%22%22%20style%3D%22endArrow%3Dclassic%3Bhtml%3D1%3BexitX%3D0.75%3BexitY%3D0.1%3BexitDx%3D0%3BexitDy%3D0%3BexitPerimeter%3D0%3B%22%20edge%3D%221%22%20source%3D%222%22%20target%3D%223%22%20parent%3D%221%22%3E%3CmxGeometry%20width%3D%2250%22%20height%3D%2250%22%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22110%22%20y%3D%22340%22%20as%3D%22sourcePoint%22%2F%3E%3CmxPoint%20x%3D%22160%22%20y%3D%22290%22%20as%3D%22targetPoint%22%2F%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2211%22%20value%3D%22%22%20style%3D%22endArrow%3Dclassic%3Bhtml%3D1%3BexitX%3D1%3BexitY%3D0.3333333333333333%3BexitDx%3D0%3BexitDy%3D0%3BexitPerimeter%3D0%3B%22%20edge%3D%221%22%20source%3D%222%22%20target%3D%224%22%20parent%3D%221%22%3E%3CmxGeometry%20width%3D%2250%22%20height%3D%2250%22%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22230%22%20y%3D%22310%22%20as%3D%22sourcePoint%22%2F%3E%3CmxPoint%20x%3D%22280%22%20y%3D%22260%22%20as%3D%22targetPoint%22%2F%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2212%22%20value%3D%22%22%20style%3D%22endArrow%3Dclassic%3Bhtml%3D1%3BexitX%3D0.5%3BexitY%3D0.5%3BexitDx%3D0%3BexitDy%3D0%3BexitPerimeter%3D0%3B%22%20edge%3D%221%22%20source%3D%222%22%20target%3D%226%22%20parent%3D%221%22%3E%3CmxGeometry%20width%3D%2250%22%20height%3D%2250%22%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22410%22%20y%3D%22320%22%20as%3D%22sourcePoint%22%2F%3E%3CmxPoint%20x%3D%22460%22%20y%3D%22270%22%20as%3D%22targetPoint%22%2F%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2213%22%20value%3D%22%22%20style%3D%22endArrow%3Dclassic%3Bhtml%3D1%3BexitX%3D0.6%3BexitY%3D0.8%3BexitDx%3D0%3BexitDy%3D0%3BexitPerimeter%3D0%3B%22%20edge%3D%221%22%20source%3D%222%22%20target%3D%227%22%20parent%3D%221%22%3E%3CmxGeometry%20width%3D%2250%22%20height%3D%2250%22%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22410%22%20y%3D%22320%22%20as%3D%22sourcePoint%22%2F%3E%3CmxPoint%20x%3D%22460%22%20y%3D%22270%22%20as%3D%22targetPoint%22%2F%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2214%22%20value%3D%22%22%20style%3D%22endArrow%3Dclassic%3Bhtml%3D1%3BexitX%3D0%3BexitY%3D1%3BexitDx%3D0%3BexitDy%3D0%3BexitPerimeter%3D0%3BentryX%3D0%3BentryY%3D0%3BentryDx%3D0%3BentryDy%3D0%3B%22%20edge%3D%221%22%20source%3D%222%22%20target%3D%228%22%20parent%3D%221%22%3E%3CmxGeometry%20width%3D%2250%22%20height%3D%2250%22%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22410%22%20y%3D%22320%22%20as%3D%22sourcePoint%22%2F%3E%3CmxPoint%20x%3D%22460%22%20y%3D%22270%22%20as%3D%22targetPoint%22%2F%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3C%2Froot%3E%3C%2FmxGraphModel%3E

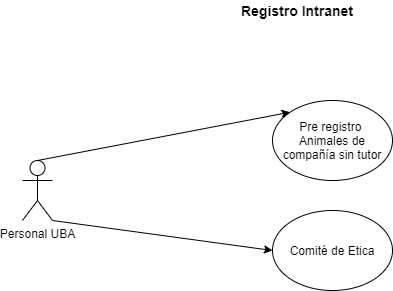
### 5.1.2 Parametrización



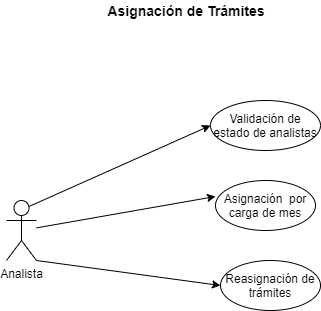
### 5.1.3 Registro Internet



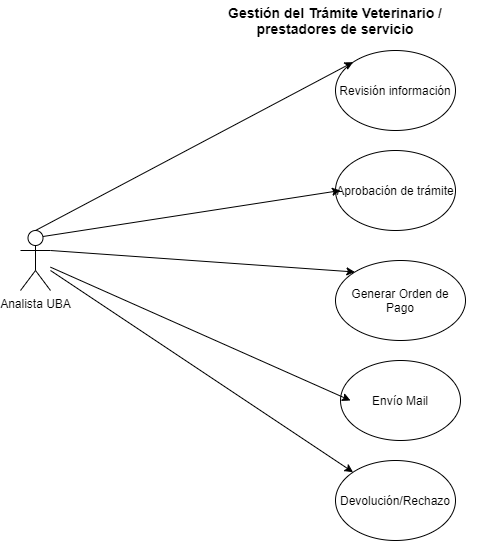
### 5.1.4 Registro Intranet



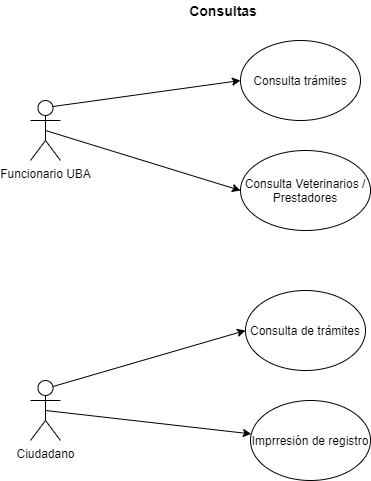
### 5.1.5 Asignación de Trámites



### 5.1.6 Gestión del trámite Veterinario / prestadores de servicio



### 5.1.7 Consultas

****

## 5.2 Vista Lógica: Diagramas de Clases

Los diagramas correspondientes a esta sección son actualizados constantemente por varios desarrolladores, por lo que pueden ser encontrados en cada proyecto del Team Foundation Server (TFS) y pueden ser encontrados en la dirección: XXXXXXXXXXXXXXXXXX

## 5.3 Vista Lógica: Diagramas Entidad Relación

Los diagramas correspondientes a esta sección son actualizados constantemente por varios desarrolladores y pueden ser encontrados en la dirección: XXXXXXXXXXXXXXXXXX

Nombrado de esquemas:

1. El “Instructivo Para la Creación de Ambientes Bases de Datos en RDBMS SQL SERVER”, Numeral 3 “ Sintaxis:[NombreBaseDatos] “, no indica el nombre que se debe colocar cuando existe más de un esquema en la misma base de datos.
2. La base de datos pre existente se denomina MDMQ\_REMETFU.
3. Anteponer el nombre de la base de datos para el nombre del esquema no agrega valor puesto que la sentencia quedaría:

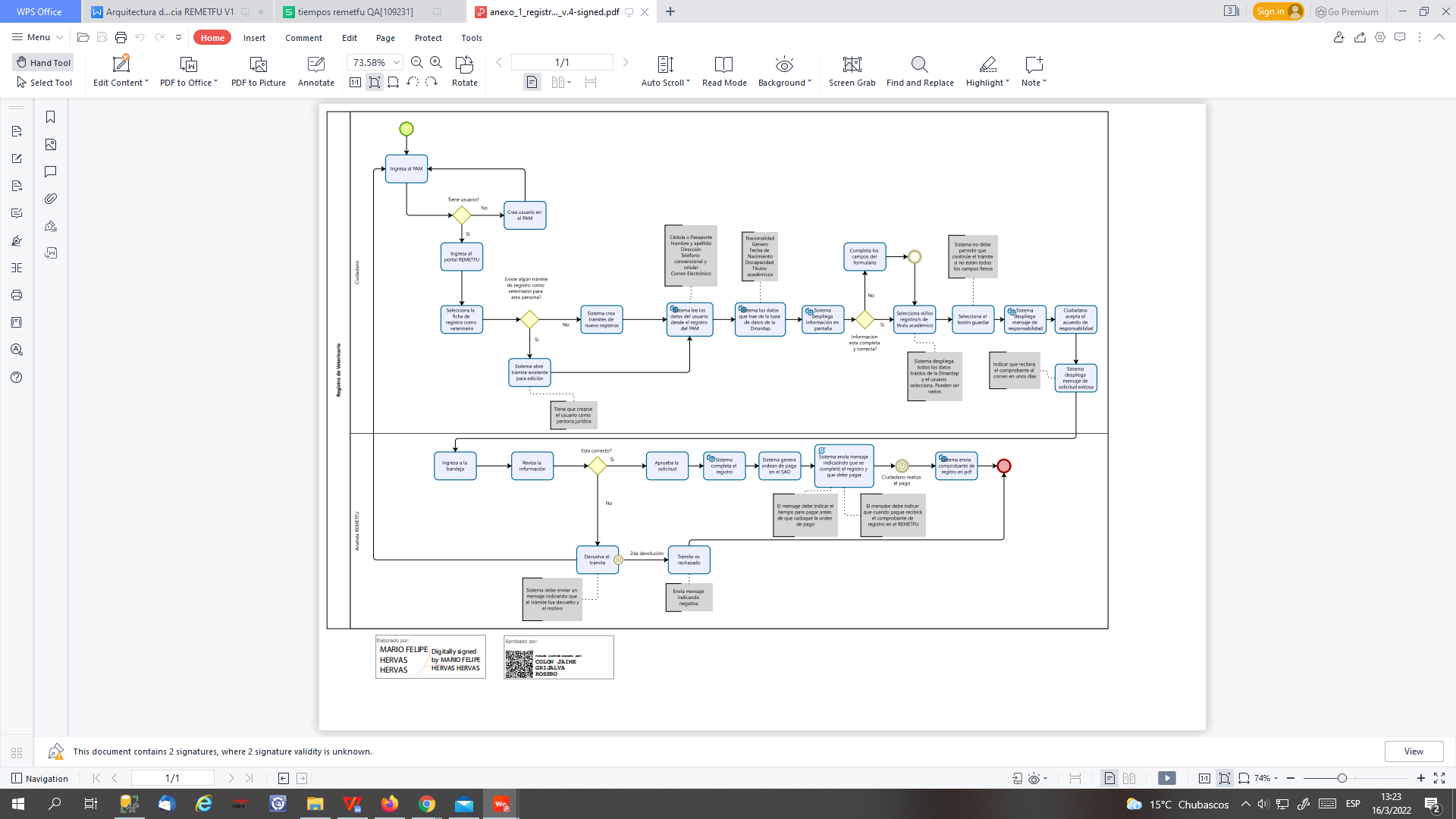
MDMQ\_REMETFU.NOMBRE\_TABLA

## 5.4 Vista de Procesos: Diagramas de Actividades / Procesos

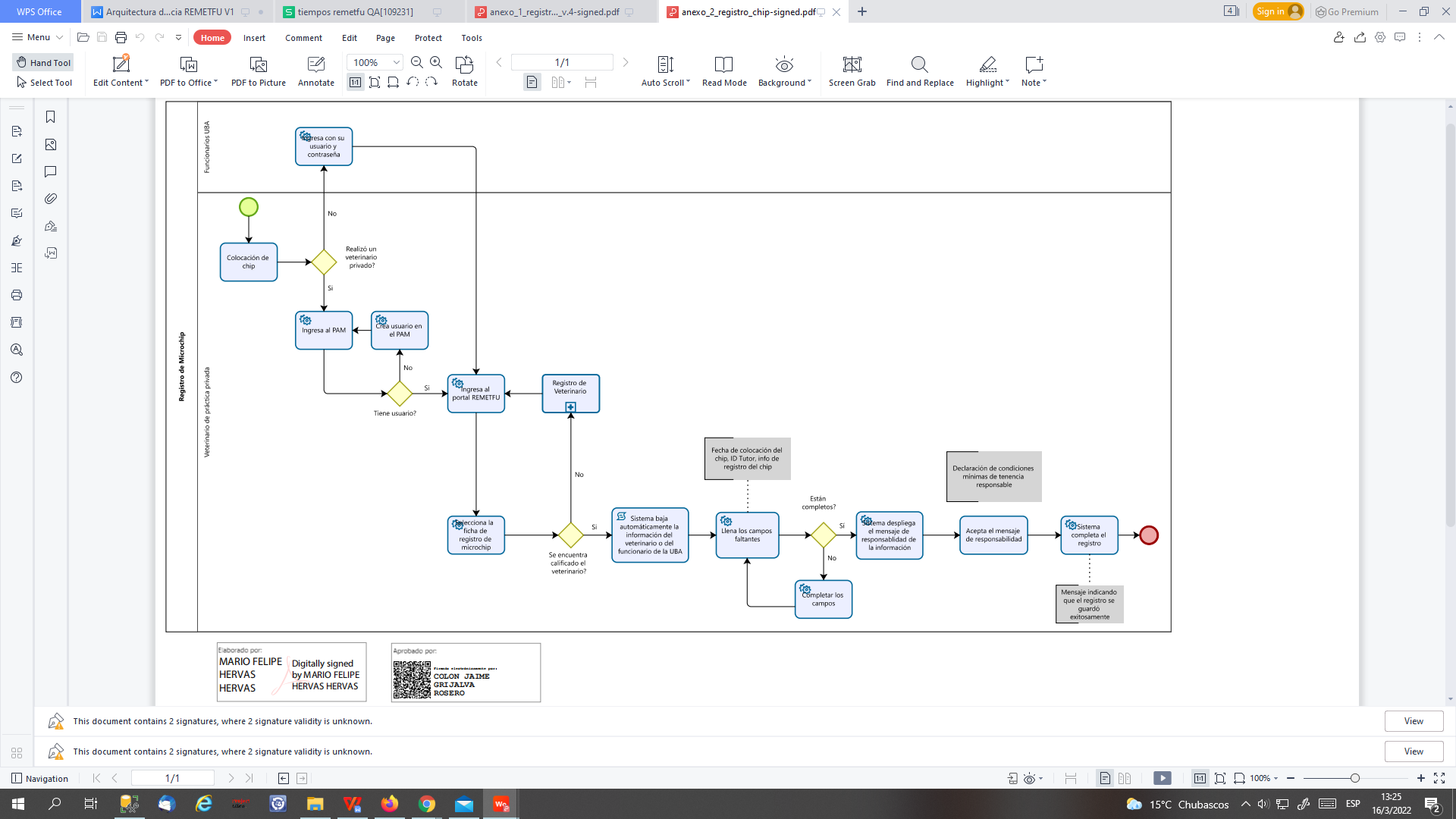
### PROCESOS DEL SISTEMA REMETFU

### 5.4.1 Administración de usuarios

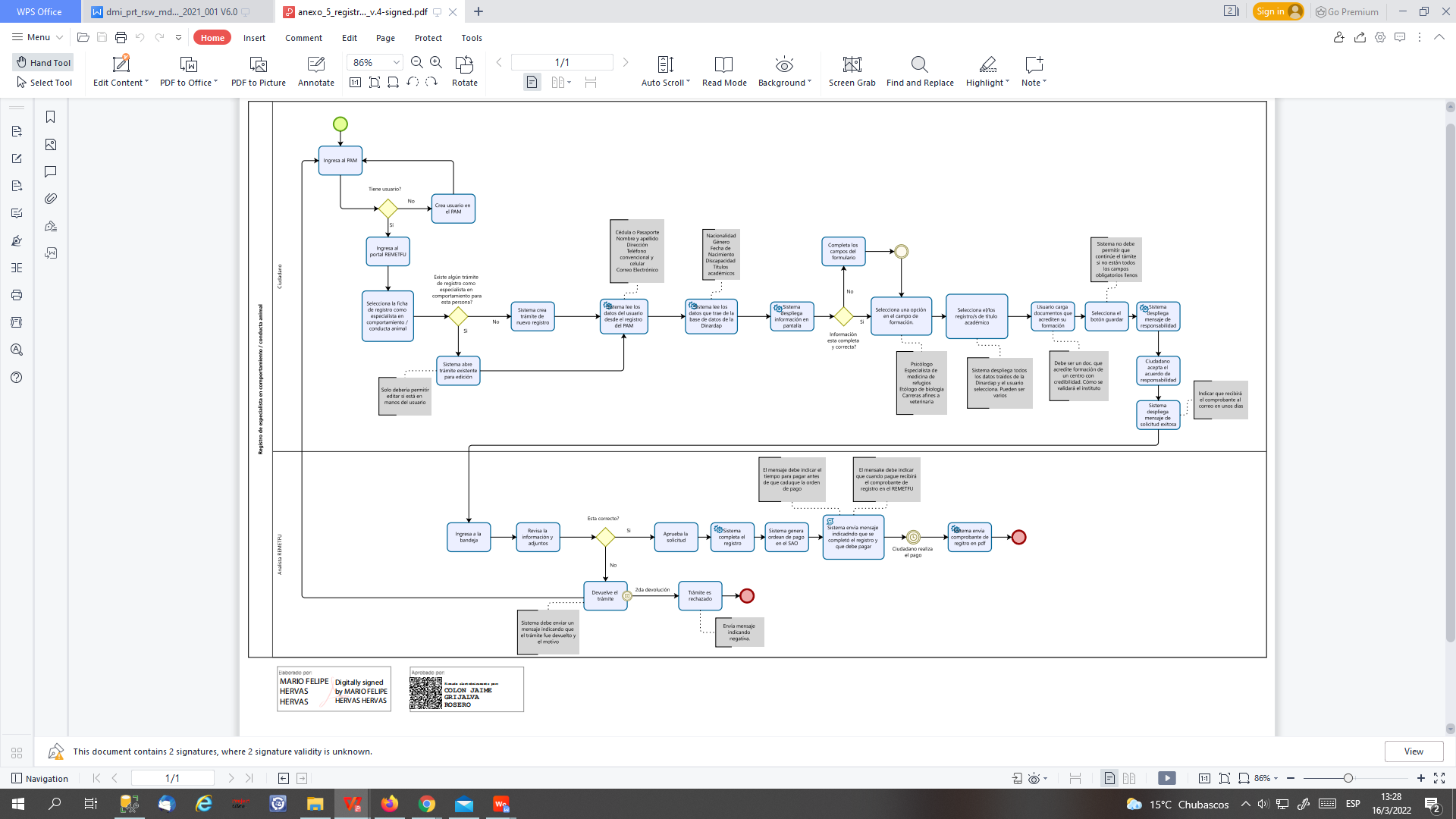
### 5.4.3 Registro de Veterinarios



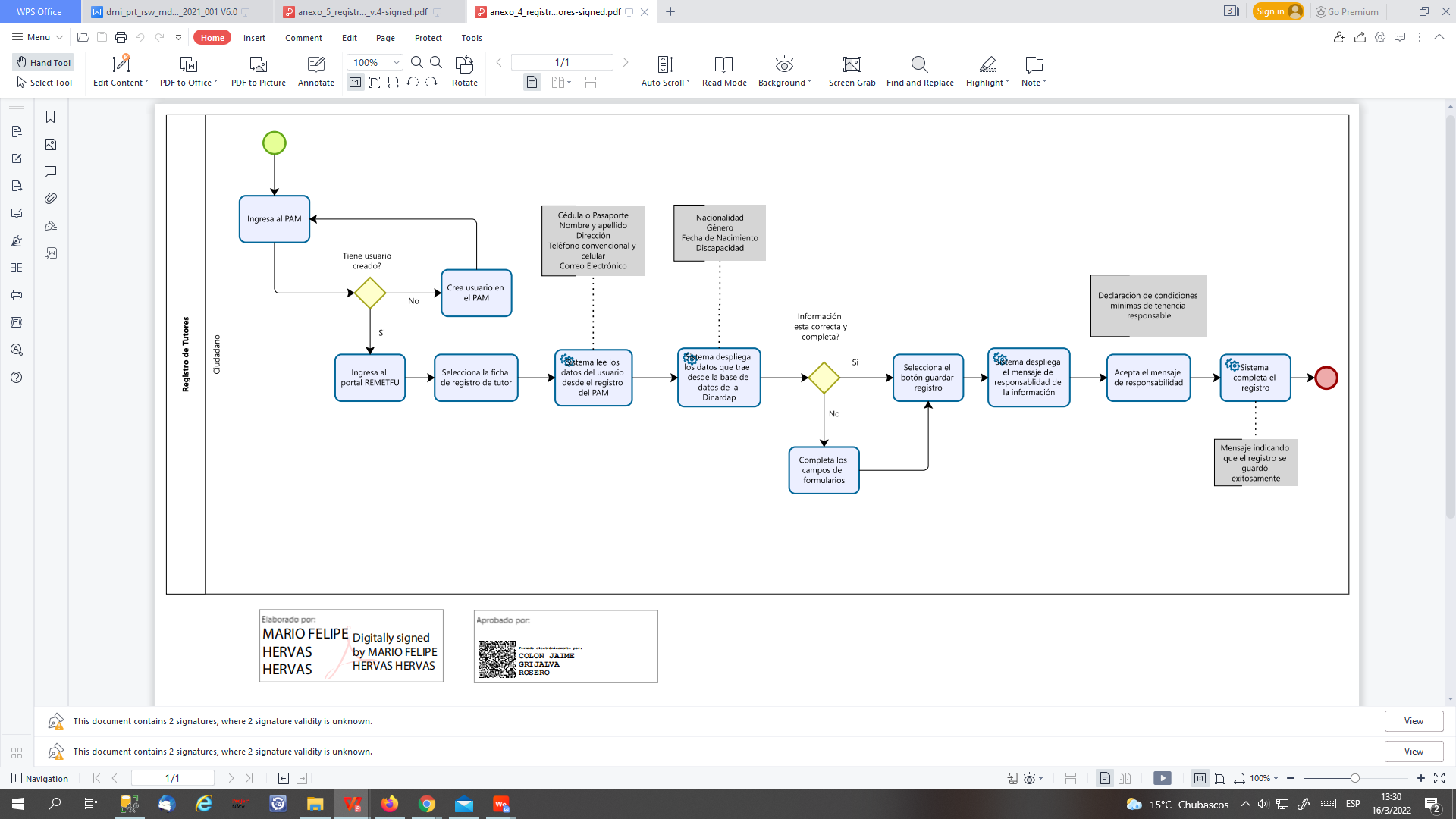
### 5.4.4 Registro de los microchip



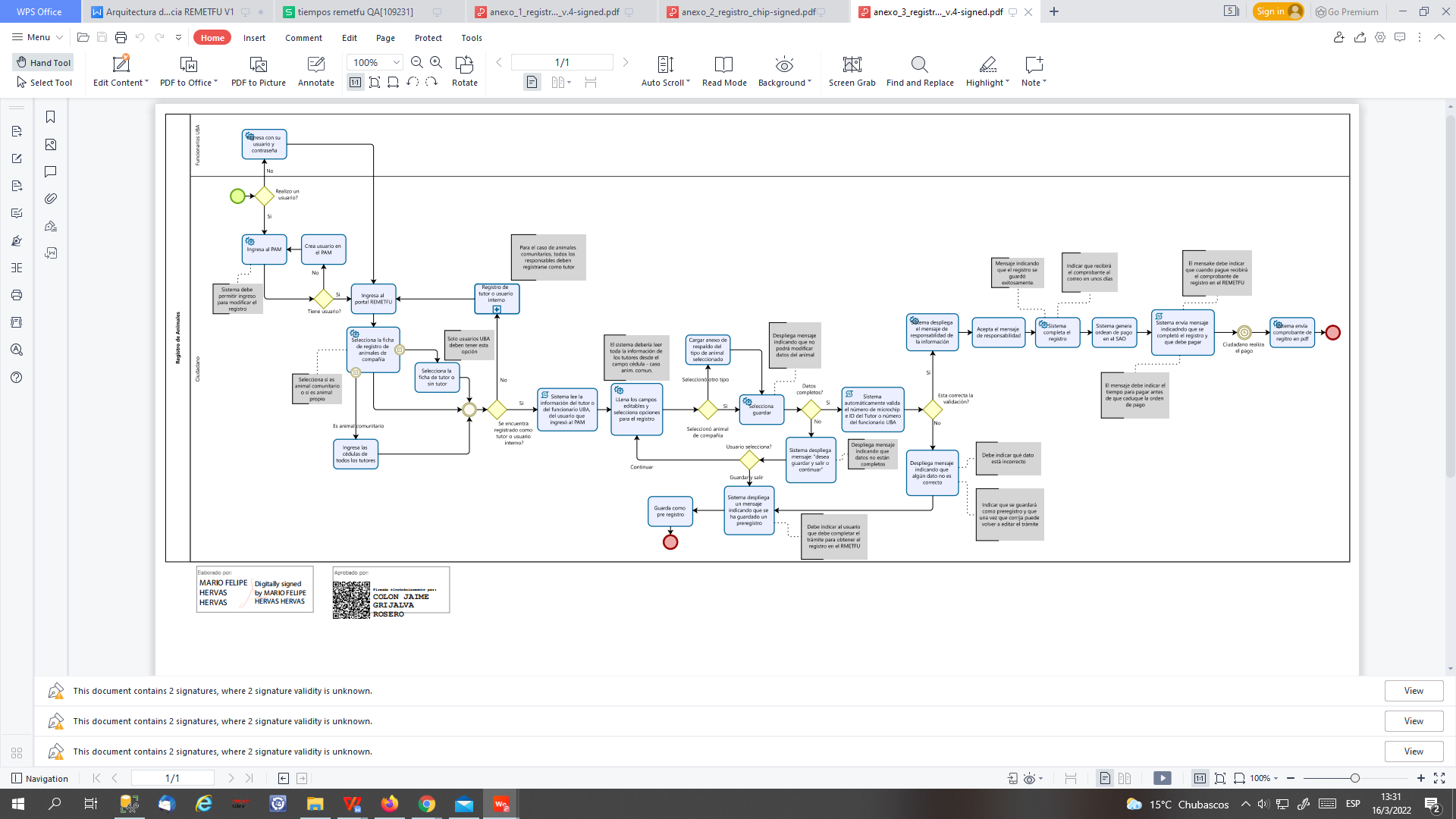
### 5.4.4 Registro de especilista en conducta animal



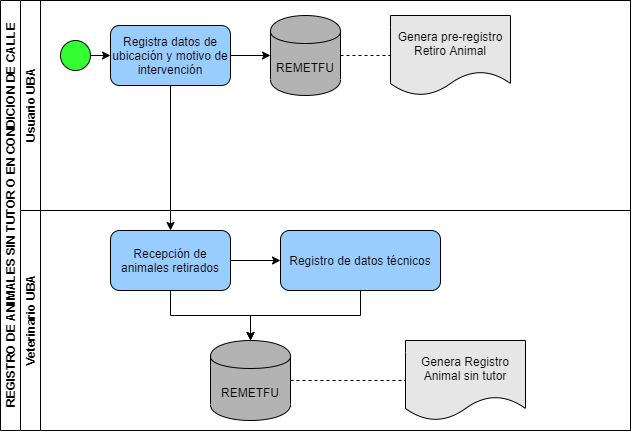
**5.4.5 Registro de tutores**



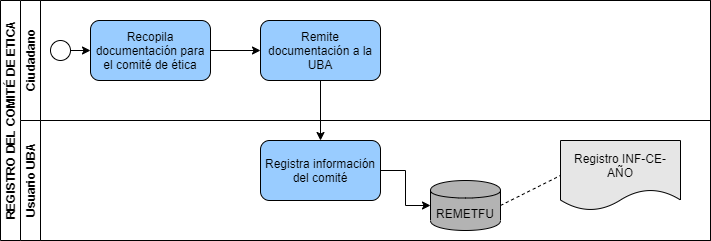
**5.4.6 Registro de animales de compañía con tutor**



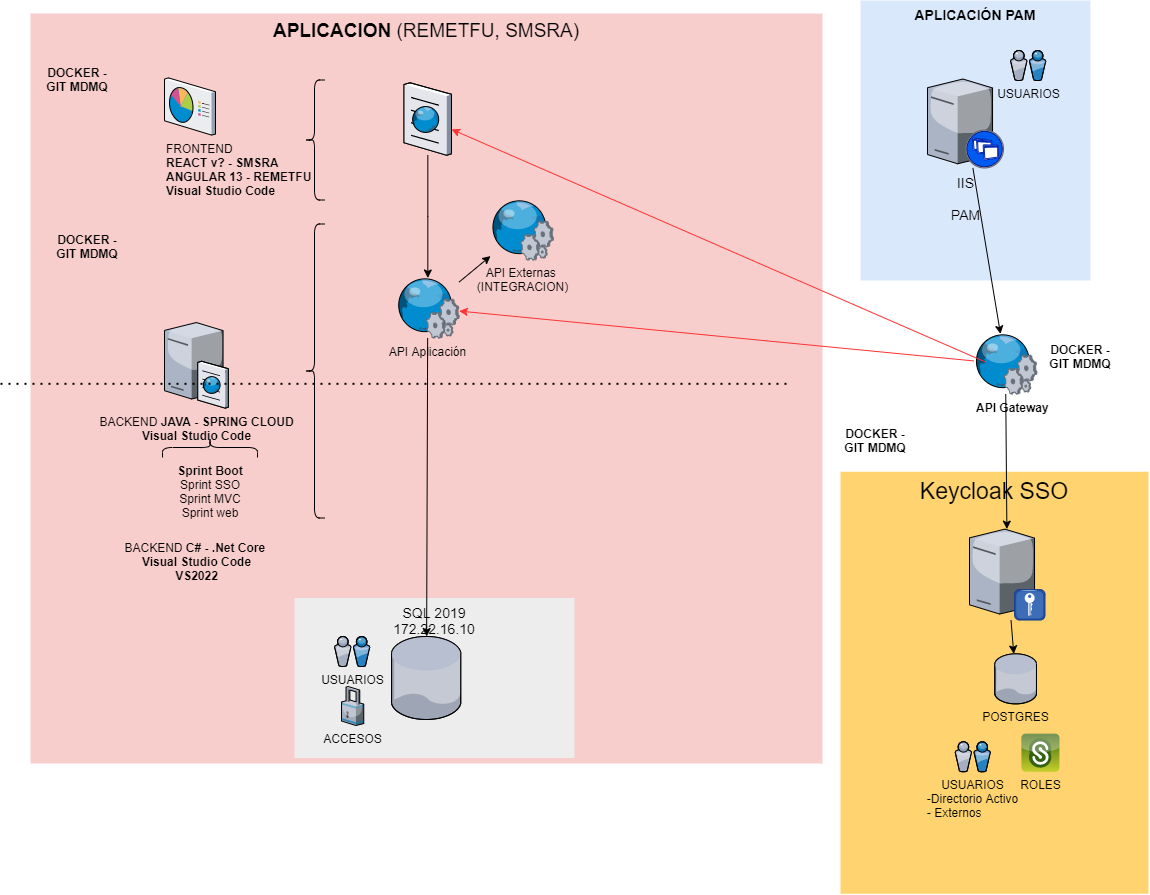
### 5.4.7 Registro de animales sin tutor



**5.4.8 Registro de comité de ética**



## 5.6 Vista de Desarrollo: Diagrama de N Capas



*Figura 2: Diagrama de Capas*

En la figura 2 se aprecia la arquitectura orientada a la construcción de APIS mediante Spring boot con integración a servicios SOAP, API GATEWAY y a la base de datos SQL SERVER

1. FRONTEND: en ésta capa se utiliza Angular 13, y se integra al API GATEWAY del Keycloak para el loguin, y el PAM.
2. BACKEND:
   * 1. API de aplicación: integra APIS/SERVICIOS de sistemas externos y servicios propios del aplicativo
3. El Controlador solicita los datos al Modelo mediante el Data Transfer Object (DTO), para lo cual el modelo está conformado por las siguientes capas:
   * DTO: objetos que sirven para el intercambio de información entre las capas de presentación, servicios y negocio.
   * Servicio: clases y referencias a servicios.
   * Negocio: clases con la lógica de negocio.
   * Datos: clases que ejecutan operaciones CRUD sobre la base de datos.
   * Entidades: sirven para la interacción entre la capa de negocio, datos y base de datos.
4. El Modelo devuelve los datos.
5. El controlador selecciona una vista.
6. Se devuelve la vista seleccionada.
7. El controlador devuelve una página aspx que carga los datos del modelo seleccionado.

# CONCLUSIONES

* La Arquitectura de la Solución del Sistema de REMETFU sigue las recomendaciones, principios y marco conceptual que consta en el Documento de Arquitectura General de Referencia.
* Al ser un documento con información cambiante existe la posibilidad de hacer ajustes a las vistas una vez que se entreguen los documentos formales de Requerimientos Funcionales.

# RECOMENDACIONES

* Se recomienda detallar las características físicas de los componentes según la Arquitectura planteada mediante un Capacity Plan elaborado en por el área de Infraestructura de la DMI.
* Se recomienda la entrega total y formal de los Requerimientos Funcionales y Anexos.

# GLOSARIO

* DMI: Dirección Metropolitana de Informática
* Capacity Plan: herramienta de análisis en donde se plasman las características físicas, versiones, licencias y valores de los componentes de hardware de la solución.