

**Documento de Arquitectura Técnica y Funcional para el Desarrollo de Proyectos o Requerimientos de Software**

****

Versión 1.7

Caracas, Febrero 2020

**TABLA DE CONTENIDO**

[Condiciones Generales 4](#_Toc29990280)

[Consideraciones Para El Llenado Del Documento 5](#_Toc29990281)

[Alcances de los Servicios De la Fabrica 5](#_Toc29990282)

[Datos Generales 6](#_Toc29990283)

[INTRODUCCIÓN 7](#_Toc29990284)

[1. Ficha Del Proyecto o Requerimiento 8](#_Toc29990285)

[2. ALCANCES Y ESPECIFICACIONES FUNCIONALES 9](#_Toc29990286)

[2.1 Requisitos Funcionales 9](#_Toc29990287)

[2.2 Delimitaciones 9](#_Toc29990288)

[2.3 Requisitos NO Funcionales 10](#_Toc29990289)

[3. Diseño Lógico 10](#_Toc29990290)

[3.1 Especificación de módulos 10](#_Toc29990291)

[3.1 Diagramas Del Sistema 10](#_Toc29990292)

[3.3 Pantallas, Notificaciones electrónicas (correos, mensajes, etc.) y Reportes. 15](#_Toc29990293)

[3.4 Interfaces. 15](#_Toc29990294)

[3.5 Entrada / Salidas (Reglas de Negocio). 16](#_Toc29990295)

[3.6 Especificaciones De Programas, Librerías, Copys, Rutinas, Clases y Servicios 16](#_Toc29990296)

[3.7 Definición de archivos de datos (Solo para AS400) 16](#_Toc29990297)

[4. Diseño Físico Detallado 17](#_Toc29990298)

[5. Diseño Lógico Detallado 19](#_Toc29990299)

[6. Análisis Volumétrico 21](#_Toc29990300)

[7. Infraestructura Requerida 21](#_Toc29990301)

[8. Seguridad y Auditoria 24](#_Toc29990302)

[9. Manejo De Datos Externos 24](#_Toc29990303)

[10. Respaldos, Retención Y Recuperación De Información, Logs Y Trazas. 24](#_Toc29990304)

[11. Proceso de Validación de Resultados 24](#_Toc29990305)

[12. Tablas Maestras (Sólo para Core Bancario). 24](#_Toc29990306)

[13. Plan De Despliegue Y Requisitos De Implantación. 25](#_Toc29990307)

[14. Estructura En Aldon. 25](#_Toc29990308)

[15. Roles Y Responsabilidades De La Solución. 25](#_Toc29990309)

[16. Glosario De Términos. 26](#_Toc29990310)

[17. Exclusiones 27](#_Toc29990311)

[18. Metodología De la Fabrica: SCRUM - IterAE© 28](#_Toc29990312)

[19. ANEXO – Workflow Fabrica 29](#_Toc29990313)

[19.1 Procesos De Desarrollo De La Fabrica 30](#_Toc29990314)

[19.2 Procesos Transversales De La Fabrica 35](#_Toc29990315)

**DATOS IDENTIFICATIVOS**

|  |  |
| --- | --- |
| **INFORMACIÓN DEL PROYECTO** | |
| **Nombre del Proyecto o Requerimiento: SONIAT CHAT BOT** |  |
| **Gerente de Proyecto – CLIENTE XXXX:** | VENTAS,PREVENTA y MERCADEO |
| **Director General - SOAINT:** | Alexander Tadeus Méndez |
| **Gerente de Operaciones - SOAINT:** | Luisa Denis Cartaya |
| **Líder Técnico - SOAINT:** | Alex |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CONTROL DE VERSIONES DEL DOCUMENTO** | | | | |
| **Versión** | **Fecha** | **Descripción del Cambio** | **Estado** | **Realizados por :** |
| **1.0** | 09 dic. 19 | Creación del documento |  | José Borrego |
| **1.2** | 16 dic. 19 | Ajustes del Documento |  | Luisa Cartaya |
| **1.3** | 07 ene. 20 | Ajustes del Documento |  | José Borrego  Luisa Cartaya |
| **1.4** | 13 ene. 20 | Ajustes del Documento |  | José Borrego  Luisa Cartaya |
| **1.5** | 14 ene. 20 | Ajustes de Forma |  | José Borrego |
| **1.6** | 15 ene. 20 | Ajustes de Forma |  | José Borrego  Luisa Cartaya |
| **1.7** | 16 ene. 20 | Ajustes de Forma |  | Daniel Barreto  Oscar Marquez |

|  |  |  |
| --- | --- | --- |
| **REVISORES** | | |
| **Nombre** | **Posición** | **Firma** |
| Luisa Cartaya | Gerente de Operaciones |  |
| Alexander Méndez | Director General |  |

# Condiciones Generales

* **SOAINT** se reserva todos los derechos sobre este documento. La información del presente no puede ser reproducida ni total, ni parcialmente, por cualquier medio mecánico o electrónico sin la correspondiente autorización escrita y citando su procedencia. **SOAINT**, se reserva el derecho de cambiar o revisar, sin previo aviso, todo o parte del presente documento.
* **SOAINT** no se responsabiliza de los daños que el uso de esta información pueda producir de forma directa o indirecta. Todas las marcas y nombres de productos citados, son propiedad de sus respectivos fabricantes.
* La información contenida en este documento es responsabilidad exclusiva de **EL CLIENTE**, y el mismo constituye la base para el desarrollo y entrega del producto final.

# Consideraciones Para El Llenado Del Documento

Para el llenado del presente documento, se requiere tomar en cuenta la información siguiente:

l

* **Por favor no elimine ninguna sección del Documento, colocar “no aplica” si es el caso**.

# Alcances de los Servicios De la Fabrica

Se definen los alcances por los Servicios de Outsourcing para Desarrollos de Sistemas bajo el Lenguaje XXXXXXXXXXXXX propuesto para **EL CLIENTE** se detalla a continuación:

**Los servicios ofertados incluyen las siguientes tareas:**

* Análisis y Desarrollo de Aplicaciones bajo Lenguaje XXXXXXXXXXXXX.
* Consultorías sobre aplicativos desarrollados bajo el lenguajeXXXXXXXXXXXXXde **EL CLIENTE**

**Esquema del Servicio**

* La bolsa de horas tiene validez de acuerdo a las condiciones de la opción seleccionada por **EL CLIENTE**, siendo regularizada al final del período.
* Las horas no consumidas se pierden automáticamente (no son acumulables).
* Posibilidad de ampliación puntual de las horas, si fuese necesario y siempre de mutuo acuerdo entre las partes.
* Los servicios se realizarán exclusivamente de forma remota a través de VPN suministrada por **EL CLIENTE.**
* Números telefónicos y dirección email de asistencia a disposición del cliente para consultorías sobre los desarrollos o extraordinaria.
* Los requerimientos sobre desarrollos y control de proyectos estarán bajo responsabilidad del cliente.
* Los ambientes de: desarrollos, calidad, producción y accesos son responsabilidad **EL CLIENTE.**
* El horario por concepto del Servicio de Outsourcing para Desarrollos de Sistemas bajo Lenguaje XXXXXXXXXXXXX es con el esquema 5x8 de 8:00 am a 5:00pm. (lunes / viernes) en el Horario de Venezuela.
* La atención de Desarrollos y/o Consultorías en horario fuera del habitual, **EL CLIENTE** deberá comunicarlo por escrito a SOAINT con antelación para poder gestionar la logística respectiva y la asignación de los recursos.

# Datos Generales

|  |  |
| --- | --- |
| **NOMBRE DEL PROYECTO O REQUERIMIENTO** | **CODIGO** |
| SONIAT CHAT BOT |  |

|  |  |
| --- | --- |
| **ELABORADO POR** | **CARGO** |
| JOSE GUERRA | COORDINADOR DE PREVENTA |

|  |  |
| --- | --- |
| **REVISADO POR** | **FECHA DE REVISIÓN** |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **CONTROL DE VERSIONAMIENTO** | | | |
| **VERSIÓN** | **FECHA** | **REVISADO POR** | **DESCRIPCIÓN** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# INTRODUCCIÓN

Este documento detalla los requerimientos de software para el **SONIAT CHAT BOT,** Esta información establece los lineamientos y las restricciones que debe considerar el equipo de desarrollo del proyecto para el desarrollo del sistema.

Se debe realizar una breve introducción del Proyecto o Requerimiento a ejecutar y los resultados deseados de manera clara y objetiva.

# Ficha Del Proyecto o Requerimiento

|  |  |
| --- | --- |
| **DESCRIPCIÓN DEL PROYECTO o REQUERIMIENTO** | |
| Explique de forma general los trabajos requeridos para lograr el objetivo, respondiendo a las siguientes interrogantes ¿Qué se quiere?, ¿Quién lo requiere?, ¿Cómo se requiere?, ¿Dónde?, ¿Hasta dónde llega?  **Que se Quiere: Se requiere una aplicación basada en un servicio web para interactuar con los posibles clientes de SOAINT en la página web de SOAINT VENEZUELA tipo chat bot.**  **Quien lo requiere: Mercadeo y Ventas de SOAINT**  **Como se Requiere: Se requiere sea desarrollado bajo un esquema de microservicio en el backend y una interfaz reactiva para el frontend usando archivos AIML para el cerebro del chat bot**  **Dónde: En los servidores de SOAINT para ser accedido desde la web.**  **Hasta donde llega: Puesta e producción del chat bot en servidores SOAINT. Una vez desplegado en producción esta aplicación estará Administrada por Ventas, Operaciones e infraestructura.** | |
|
|
|  | |
| **OBJETIVO GENERAL** | |
| Describa la meta general que se desea lograr con la implementación del proyecto o requerimiento, con su respectivo propósito. Recuerde utilizar verbos de acción en infinitivo.  **Implementar un Chat Bot para el área de venta y preventa de SOAINT Venezuela que suministre información acerca de los productos y servicios que la empresa provee.** | |
|  | |
| **OBJETIVOS ESPECÍFICOS** | |
| Describa las metas detalladas que se desean lograr con la implementación del proyecto o requerimiento, con su respectivo propósito. Recuerde que deben ser medibles, preciso, congruente con la realidad del negocio y la necesidad planteada para el desarrollo del proyecto, evite el uso de frases subjetivas. Recuerde utilizar verbos de acción en infinitivo.   * Crear un microservicio con Spring Boot que provea de las funciones de análisis sintáctico para dar respuesta a las preguntas que los usuarios realicen en la página de SOAINT. * Crear una base de conocimientos con AIML que permite respuestas estándares asociadas a los productos y servicios y que además de al usuario la sensación de estar conversando con una persona y no con un robot. * Crear una interfaz gráfica reactiva e independiente del backend que permita una débil acoplación con el microservicio. * Diseñar e implementar las medidas de seguridad para su posterior despliegue en producción. | |
|  | |
| **ÁREAS IMPACTADAS** | |
| **Área** | **¿Cómo se ve afectada por la ejecución del proyecto o requerimiento?** |
| Ventas | La información que suministre el chat bot debe ser validada por este departamento y en consecuencia deberá ser la misma que se suministra a los clientes en las visitas |
| Operaciones | Operaciones deberá tener control total de los fuentes y del conocimiento generado por la aplicación en su desarrollo para poder replicarlo a los diferentes clientes y/o pruebas de concepto. |
| Preventa | Este software será usado para potenciar las ventas a través de pruebas de concepto donde esta herramienta podrá ser usada como disparador para proyectos RPA |
|  |  |
|  |  |

# ALCANCES Y ESPECIFICACIONES FUNCIONALES

# 2.1 Requisitos Funcionales

* **Alcance del Proyecto o Requerimiento**

Detallar una definición exacta del producto, servicio o resultado, que obtendremos con la ejecución de la solución, subdividiéndolo en entregables para obtener el producto final requerido.

El Proyecto consta de una aplicación microservicio en el backend que a su vez consulta los archivos AIML junto con un frontend para consumir ese servicio, este puede ser una SPA (Single Page Application), una aplicación reactiva angular o un html5 con la lógica para consumir el servicio. Con esta implementación se obtendrá una aplicación Chat Bot que podrá ser usada como disparador para PoC’s (proof of concept) RPA y para reforzar en SOAINT Venezuela el tema de mercadeo y ventas de los productos y servicios. Los entregables a esperar serían:

1. Microservicio
2. Archivos AIML
3. Frontend reactivo con los colores y el logo SOAINT
4. Manual técnico para poner en contexto a futuros desarrolladores.

* **Descripción de la Situación Actual**

Indicar cuál es la situación actual del proceso o sistema que requiere su sustitución o solución, explicando **paso a paso la usabilidad de las pantallas y sus componentes**, con la finalidad de obtener un diagnóstico preciso para la toma de decisiones.

**Actualmente no se cuenta con un chat bot para ofrecer a nuestros clientes.**

* **Descripción de la Situación Propuesta**

Indicar las condiciones o capacidades que debe poseer o satisfacer el producto para cumplir con el objetivo y alcance del proyecto, detallando las características distintivas del producto requerido, explicando **paso a paso la usabilidad de las pantallas y sus componentes**.

**Se propone realizar un desarrollo usando ALICE BOT software libre basado en AIML, verificar** <https://www.chatbots.org/chatbot/a.l.i.c.e/> y http://www.pandorabots.com

# 2.2 Delimitaciones

* **Restricciones**

Factores que limitan el rendimiento del proyecto o requerimiento, el rendimiento de un proceso del proyecto, o las opciones de planificación del proyecto. Pueden aplicar a los objetivos del proyecto o a los recursos que se emplea en el proyecto o requerimiento.

**Un factor a considerar es que se requiere generar una aplicación para ser incrustada en un portal web.**

* **Supuestos**

Factores que para propósitos de la planificación del proyecto o requerimiento se consideran verdaderos, reales o ciertos.

# 2.3 Requisitos NO Funcionales

* **Volúmenes y Usuarios**

Indicar la cantidad de usuarios que harán uso del sistema y frecuencia de uso del sistema.

**No se conoce este dato.**

* **Seguridad y Control**

Describir los controles, niveles de seguridad y perfiles de los usuarios requeridos de acuerdo a las Políticas de Seguridad establecidas.

**Se establece una seguridad por token usando JWT o OAUTH2.**

# Diseño Lógico

# 3.1 Especificación de módulos

Identificación y detalle de los módulos. Aplica para todos los módulos de los sistemas (nuevos o existentes). Si este punto no aplica debe indicar **“No Aplica”.**

| **Nombre Módulo** | **Nombre Elementos** | **I/M/E** | **Descripción** |
| --- | --- | --- | --- |
| Seguridad | **Backend:** Seguridad del microservicio vía JWT o OAUTH2 | M | Agregar seguridad al microservicio con JWT o OAUTH2 |
| Conocimientos | **Backend:**  Archivos AIML | I/M/E | Incluir y/o modificar los archivos AIML para agregar conocimiento a el microservicio |
| Página del chat bot | **Frontend:**  Look and Feel | M | Modificar la apariencia de la página del chat bot (Look and feel de SOAINT) |

**I: Inclusión, M: Modificación, E: Eliminación (Esto se utiliza para indicar si el objeto es nuevo, si se está modificando uno ya existente o si se está eliminando)**

# 3.1 Diagramas Del Sistema

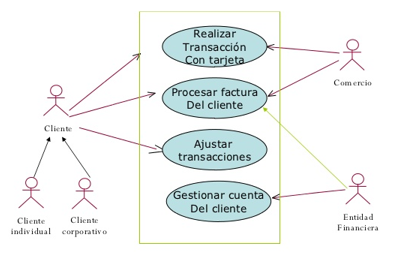
En este apartado se debe colocar los diagramas que representan la arquitectura de la solución del software propuesto, el que aplique según la naturaleza del proyecto. Los diagramas propuestos son los siguientes:

* + Diagrama de casos de uso. (Representar solo alto nivel).
  + Diagrama de Clases
  + Diagrama de flujo de procesos.
  + Diagrama de secuencia.
  + Diagrama de componentes.
  + Diagrama entidad relación. Se recomienda utilizar la notación propuesta por el Lenguaje Unificado de modelado (UML /RUP).

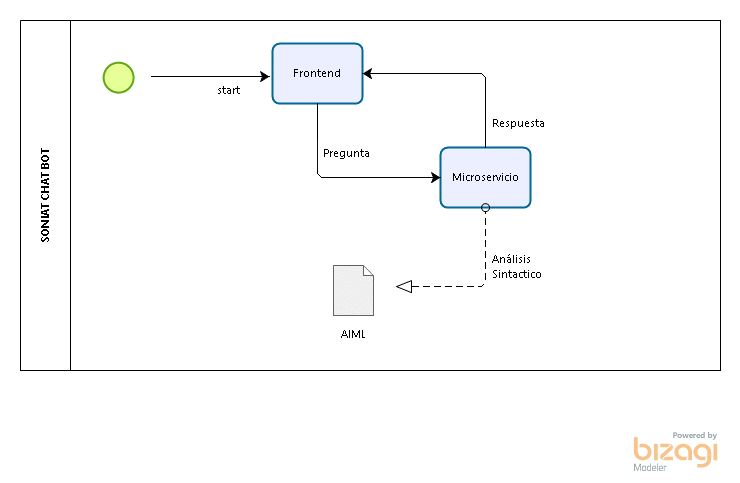
Así mismo, es necesario distinguir funciones que son realizadas automáticamente de las que se hacen en forma manual. Procesos: Descripción, Tipo de proceso, Sistema involucrado, Periodicidad, Dependencias, Disponibilidad, Parametrización.

**Diagrama de casos de uso. (General y Detallado).**

**Ejemplo Referencial:**



CASO DE USO GENERAL: Diagrama de contexto



**Descripción Detallada:**

Cada Funcionalidad expresada en el caso de uso debe ser descrita con más detalle, en términos de que debe hacer.

|  |  |  |
| --- | --- | --- |
| **Funcionalidad** | **Descripción** | **Detalle** |
| Mostrar Frontend | Debe mostrar una interfaz gráfica acorde con el look and feel de SOAINT | Logo de soaint y colores de la empresa así como el tipo de letra de la empresa |
| Mostrar conversación | La interfaz debe mostrar en un text área la conversación entre el usuario y el chat bot | Cada elemento de la conversación deberá mostrar un color diferente para identificar cada actor (Usuario y Chat Bot) |

**Diagrama de Clases.**

Es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. Está compuesto por:

Propiedades o atributos: son valores que corresponden a un objeto, como color, material, cantidad, ubicación.

Operaciones o Métodos: son aquellas actividades o verbos que se pueden realizar con/para este objeto, como por ejemplo abrir, cerrar, buscar, cancelar, acreditar, cargar.

Interfaz es un conjunto de operaciones que permiten a un objeto comportarse de cierta manera.

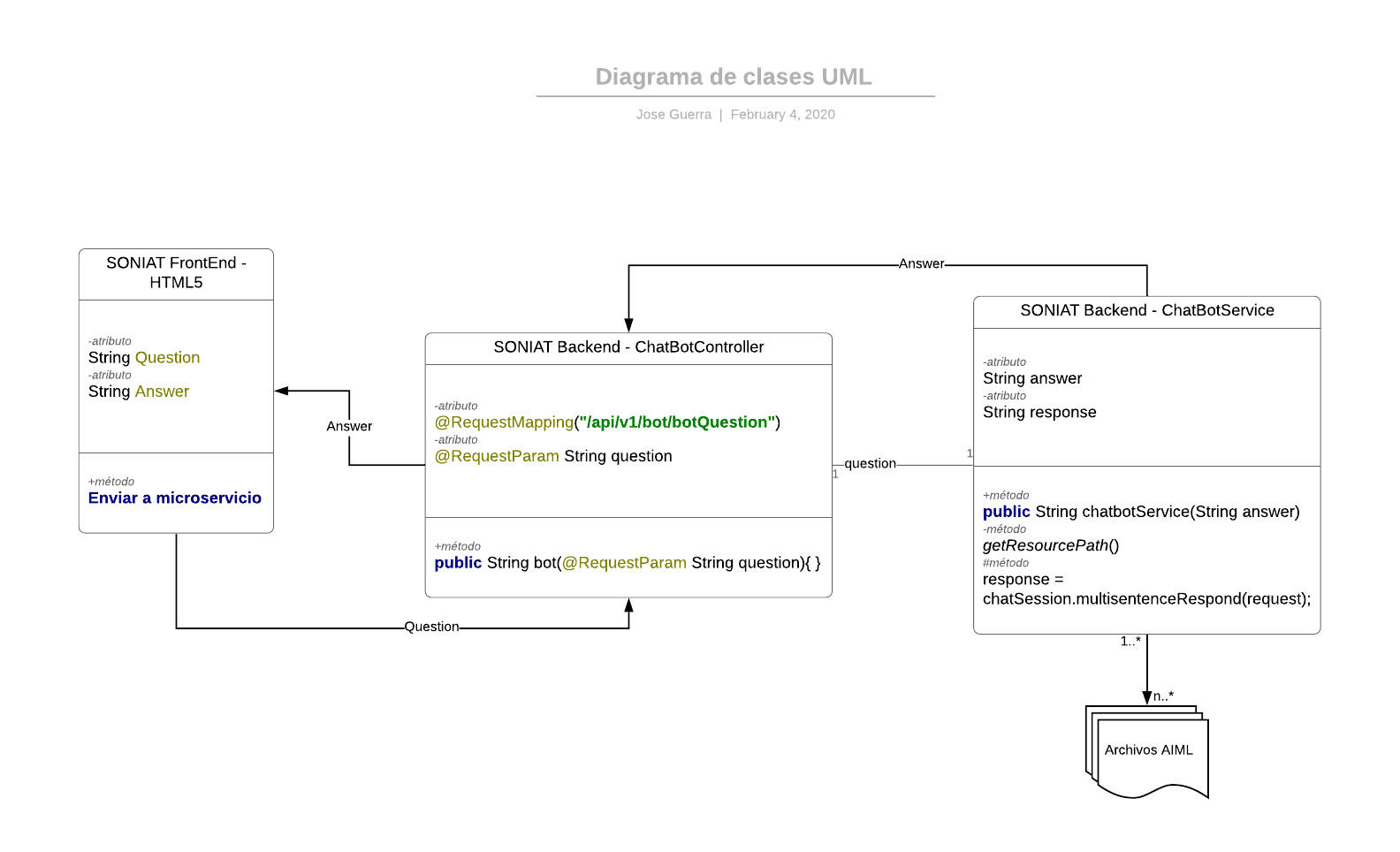
Herencia se define como la reutilización de un objeto padre ya definido para poder extender la funcionalidad en un objeto hijo. Los objetos hijos heredan todas las operaciones y/o propiedades de un objeto padre.

Solo se deben reflejar los atributos y métodos que se utilizan en la solución que se está planteando

**Ejemplo:**



Diagrama de clases:



**Descripción Detallada Atributos:** Cada Clase expresada en el diagrama de Clases debe ser descrita con más detalle, en términos de Especificaciones de tipo de campo longitud y Validaciones.

|  |  |  |  |
| --- | --- | --- | --- |
| **Clase** | **Atributo** | **Tipo y Longitud** | **Detalle Validaciones** |
| SONIAT FrontEnd - HTML5 | Question | String 255 | No puede ser null |
| SONIAT FrontEnd - HTML5 | Answer | String 255 | No puede ser null |
| SONIAT Backend - ChatBotController | @RequestMapping(**"/api/v1/bot/botQuestion"**) | String 255 | No puede ser null, No puede contener espacios |
| SONIAT Backend - ChatBotController | @RequestParam String question | String 255 | No puede ser null |
| SONIAT Backend - ChatBotService | String answer | String 255 | No puede ser null |
| SONIAT Backend - ChatBotService | String response | String 255 | No puede ser null |

**Descripción Detallada Métodos:**

|  |  |  |
| --- | --- | --- |
| **Clase** | **Método** | **Detalle** |
| SONIAT Backend - ChatBotController | **public** String bot(@RequestParam String question){ } | Método recibe un String y se lo pasa al servicio que realizará el análisis sintáctico |
| SONIAT Backend - ChatBotService | **public** String chatbotService(String answer) | Método que recibe una pregunta y procesa el texto para pasarlo al método del analizador sintáctico |
| SONIAT Backend - ChatBotService | **private static** String getResourcePath() | Método que establece la ruta de los archivos AIML |
| SONIAT Backend - ChatBotService | **public** multisentenceRespond(request) | Método que realiza el análisis sintáctico para encontrar respuesta a la pregunta |

* **Diagrama de Flujo de Procesos**

**Ejemplo Referencial:**

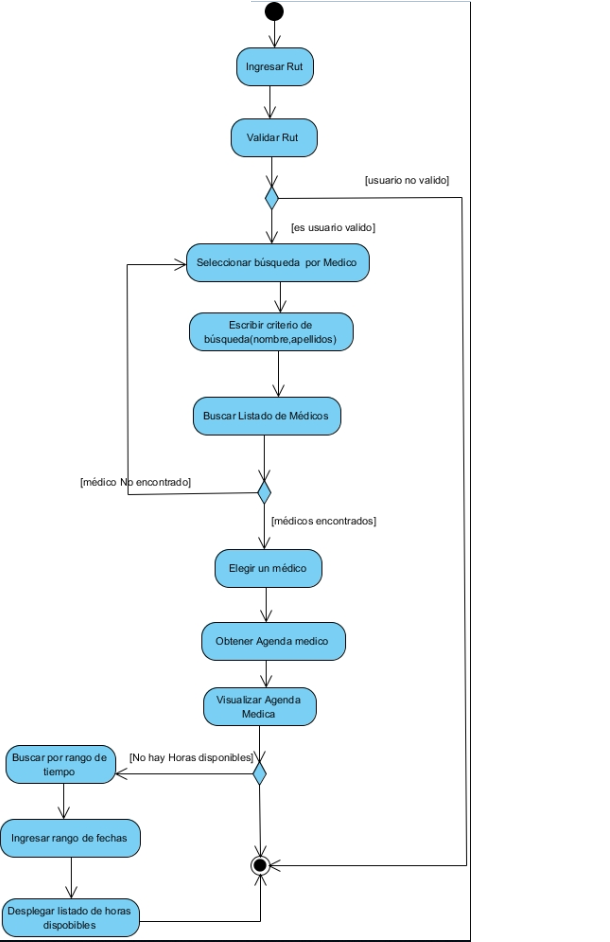
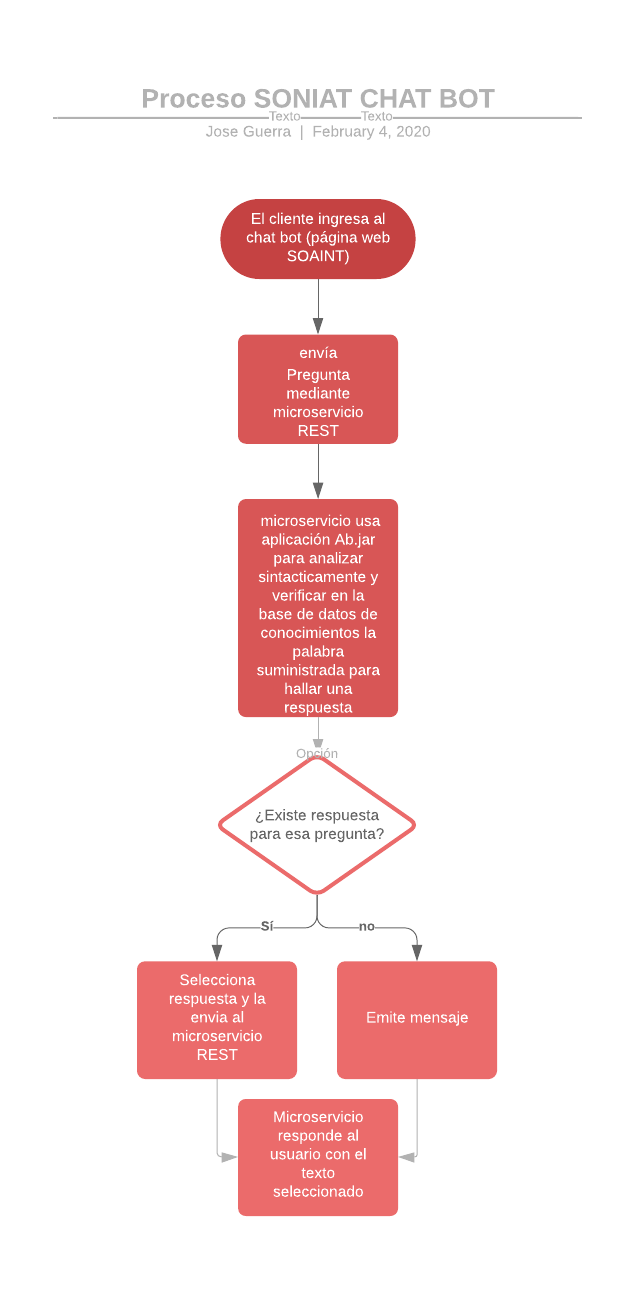


Diagrama de flujo de procesos:



**Diagrama de Secuencia.**

Este tipo de diagramas muestra una interacción ordenada según la secuencia de eventos vista a la luz de una línea de tiempo. En particular, se muestran los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado, aunque por orden lo usual es colocar los objetos de izquierda a derecha y en la parte superior. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba hacia abajo.

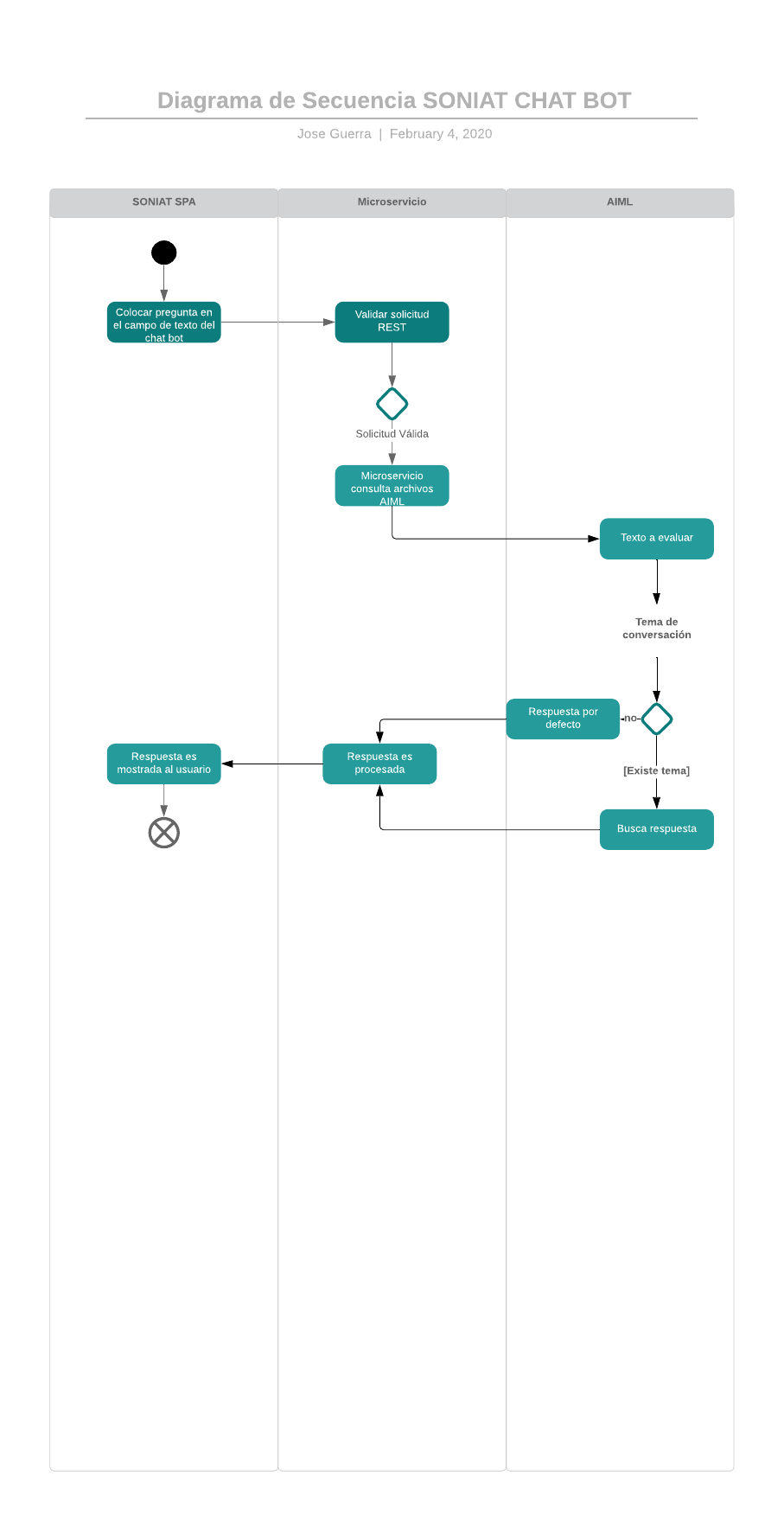
**Ejemplo Referencial:**





Si aplica, Detallar en narrativa algún aspecto del Diagrama de Secuencia

Diagrama de secuencia:



# 3.3 Pantallas, Notificaciones electrónicas (correos, mensajes, etc.) y Reportes.

En este apartado se describen las pantallas, notificaciones electrónicas (correos, mensajes), informes y reportes desde el punto de vista de su despliegue y cómo se llenan los distintos campos (variables) que utiliza. Algunos tópicos a considerar: Pantallas: Definir canal, Condiciones de Navegación entre pantallas, Layout, Detalle de campos, formato, posición significado, controles de validación, valores por defecto, componentes y/o servicios requeridos. Informes: Objetivo, Periodicidad (diaria, mensual, semanal, etc.), Forma de Visualizar (papel, on-demand, planilla Excel, archivo plano, etc), Layout, Encabezado y pies de página, Definición campo a campo, formato campo, posición, Corte de Control (Saltos de página y totales de página). Detallar los campos que van en la Pantalla / Reporte / Mensaje:

**Ejemplo Referencial:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre** | **Origen** | **Tipo** | **Validaciones** | **Formato** | **Etiqueta** |
| Pregunta | Text Element | String | Obligatorio | UTF-8 | Pregunta |
| Respuestas | Text Área Element | String | Obligatorio | UTF-8 | Conversación |

|  |  |
| --- | --- |
| **Identificación de Mensajes** | **Mensaje** |
| Código asociado | Texto del mensaje |
|  |  |

# 3.4 Interfaces.

En este apartado se describen las Interfaces que serán usadas por la solución a implementar.

**Interfaces de salida:** Periodicidad o condiciones de generación (diaria, mensual, semanal, etc), Sistemas o entidad que las recibe, Definición campo a campo, formato del campo, posición, significado, Validación del campo, Totales de control.

**Campo Text Área o similar**

**Interfaces de entrada:** Sistemas o entidad que genera, Periodicidad (diaria, mensual, semanal, etc), Controles de validación de campo a campo, Controles de validación a nivel de totales chequeo de integridad, definición campo a campo, formato del campo, posición, significado, Totales de Control.

**Campo de texto**

Ejemplo de interface:



# 3.5 Entrada / Salidas (Reglas de Negocio).

En este apartado se detallan las entradas y salidas del proceso (formato, tipo, reglas de negocio), sus relaciones con subprocesos (es decir, qué salidas de un subproceso son entradas de otros, etc.).

**Entrada:**

**Oración texto libre sin caracteres especiales**

**Salida:**

**Oración texto libre sin caracteres especiales**

**Reglas de Negocio:**

* **La base para el desarrollo de SONIAT deberá ser a partir del microservicio generado en la PoC de Cantv y se usó como estructura para crear a SONIAT (Microservicio inicial para el chat bot) el cual se suministra en el GitHub de SOAINT** [**soaintsoporte@github.com**](mailto:soaintsoporte@github.com)**.**
* **Se puede tomar como modelo el prototipo de frontend en angular 8 para consumir los servicios expuestos por el microservicio.**
* **Se tomará como base de conocimiento los archivos AIML que componen el cerebro de SONIAT, este se enfoca en aspectos de mercadeo y ventas de los productos y servicios de SOAINT.**
* **Es necesario que la aplicación pueda ser desplegada en una página web como los chats bots que se encuentran en el mercado, el prototipo suministrado en el GitHub de SOAINT incluye un frontend en angular que se presenta como un SPA sin embargo hay que adaptarlo para que pueda ser incrustado en la página web de SOAINT.**
* **En cuanto a la única pantalla que debe mostrar la app se encuentra, el logo de la empresa, los colores de la empresa, un elemento similar a un *textarea* que muestre la conversación generando un color de fondo para cada conjunto de palabras que menciones el usuario o el chat bot, por último, un elemento de *texto* para que el usuario pueda escribir las preguntas y un botón de enviar que debe aceptar el ENTER como comando por defecto.**

# 3.6 Especificaciones De Programas, Librerías, Copys, Rutinas, Clases y Servicios

En este apartado se debe documentar la funcionalidad de cada programa, copys, Librería, rutina, clase o servicio que se modifique o se agregue para el proyecto, se debe especificar la entrada, salida. Este punto es obligatorio.

Esta tabla aplica para programas, Librerías, copys y rutinas.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nombre** | **Tipo**  **Objeto** | **I,M,E** | **Descripción** | **Funcionalidad** | **Compilación** | **Afectación**  **Servicio** | **Formato** |
| Ab | Librería | I | Librería de chat bot(Analizador sintáctico) | Evaluar las frases dadas por el usuario |  |  | .jar |
| Archivos AIML | Archivos planos | I | Base de datos de conocimiento | Extender el conocimiento del chat bot |  |  | .aiml |

**I: Inclusión, M: Modificación, E: Eliminación (Esto se utiliza para indicar si el objeto es nuevo, si se está modificando uno ya existente o si se está eliminando)**

Esta tabla aplica para las clases y servicios

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Servicio /**  **Aplicativo** | **Nombre**  **Clase** | **Método o Procedimiento** | **I,M,E** | **Descripción** | **Instrucciones Especiales y/o observaciones** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**I: Inclusión, M: Modificación, E: Eliminación (Esto se utiliza para indicar si el objeto es nuevo, si se está modificando uno ya existente o si se está eliminando)**

# 3.7 Definición de archivos de datos (Solo para AS400)

En este apartado se debe detallar los archivos creados o modificados en este desarrollo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Archivo: XXXXXXX Registro: XXXXXXX** | | | | | |
| **Campo** | **Tipo** | **Longitud** | **Clave** | **Descripción** | **Valores del campo** |
| N/A |  |  |  |  |  |
| N/A |  |  |  |  |  |

# Diseño Físico Detallado

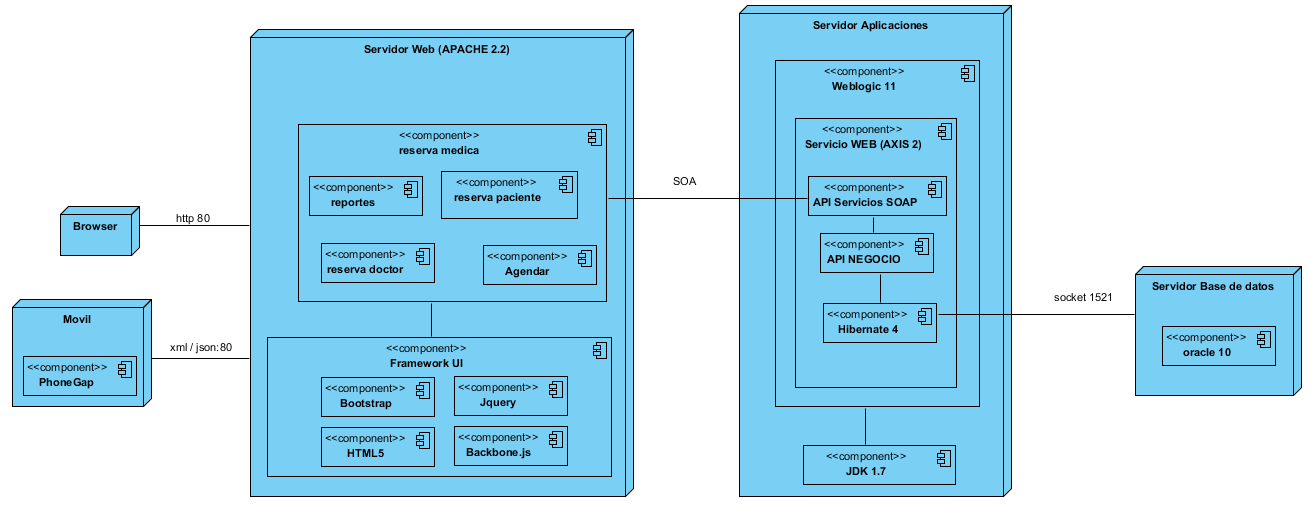
En esta sección se debe colocar el Diseño Físico de la arquitectura de Aplicación y el Diagrama de Distribución considerando las rutinas y datos a utilizar en forma de mapa del flujo transaccional.

* **Diagrama de Distribución:** este tipo de diagramas se enfoca específicamente al hardware de un sistema determinado. El elemento primordial del hardware es un nodo, que es un nombre genérico para todo tipo de recurso de cómputo. Un nodo se representa mediante un cubo, Dentro del cubo se puede introducir información sobre el nodo, que puede ser simplemente texto o inclusive componentes, usando los diagramas de componentes. Un componente de software es una parte física de un sistema, y se encuentra en la computadora en la lógica a desarrollar. Ejemplos de componentes son tablas, archivos de datos, ejecutables, bibliotecas de vínculos dinámicos, documentos cosas por el estilo. Lo que contiene un diagrama de componentes es lógicamente componentes, interfaces y relaciones, aunque también pueden aparecer otros tipos de símbolos vistos anteriormente. El símbolo principal de un diagrama de componentes es un rectángulo que tiene otros dos sobrepuestos en su lado izquierdo, con el nombre del componente dentro del rectángulo más grande, como se muestra en la siguiente figura:

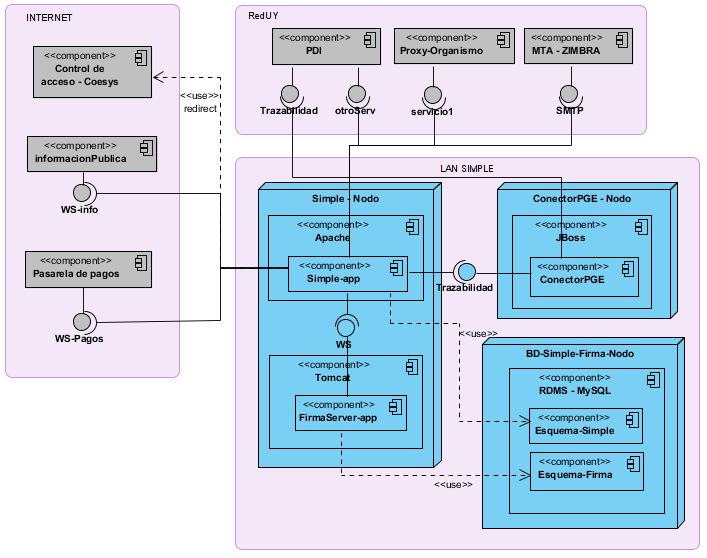


**Ejemplo Referencial:**

**Diagrama de Distribución de Componentes**



**Ejemplo Referencial:**



**Descripción de los Nodos:** Describir el Componente. / Configuración y parámetros entrada y Salida, y otros aspectos que se necesiten para su uso. Un nodo puede contener 1 o más componentes.

**Descripción de los Componentes:** Describir el Componente / Configuración y parámetros que necesita para su uso

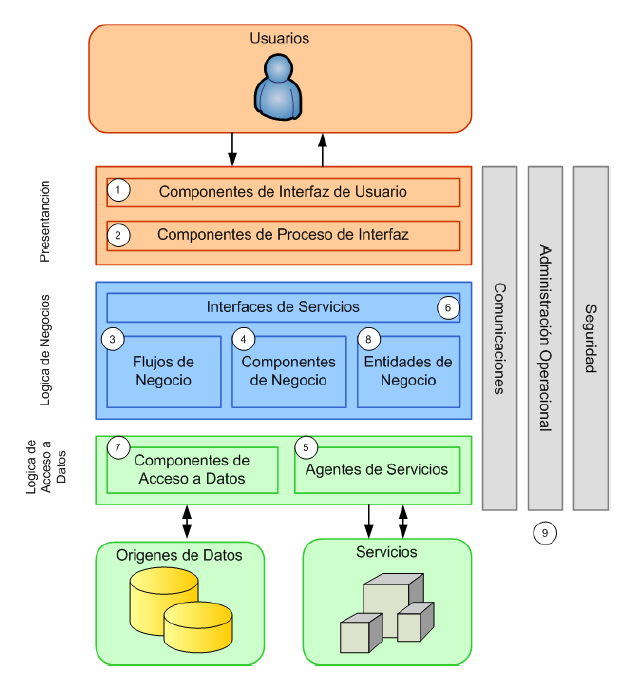
|  |  |  |  |
| --- | --- | --- | --- |
| **Nodo** | **Descripción del nodo** | **Componente** | **Detalle Componente** |
| Nodo1 |  | Componente1 |  |
| Componente2 |  |
| Nodo2 |  | Componente3 |  |
| Nodo3 |  | Componente4 |  |
| Componente5 |  |

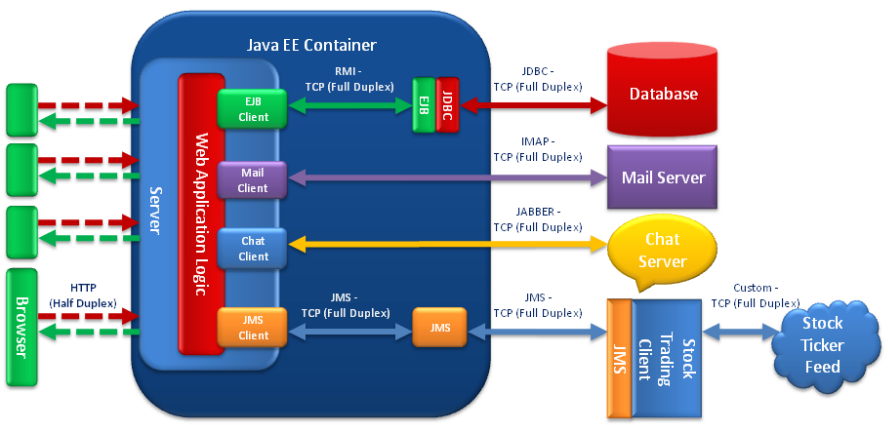
# Diseño Lógico Detallado

En esta sección se debe colocar el Diseño Detallado de la Arquitectura de Aplicación refiriéndose a:

**Arquitectura Lógica por Capas**

**Ejemplo Referencial:**

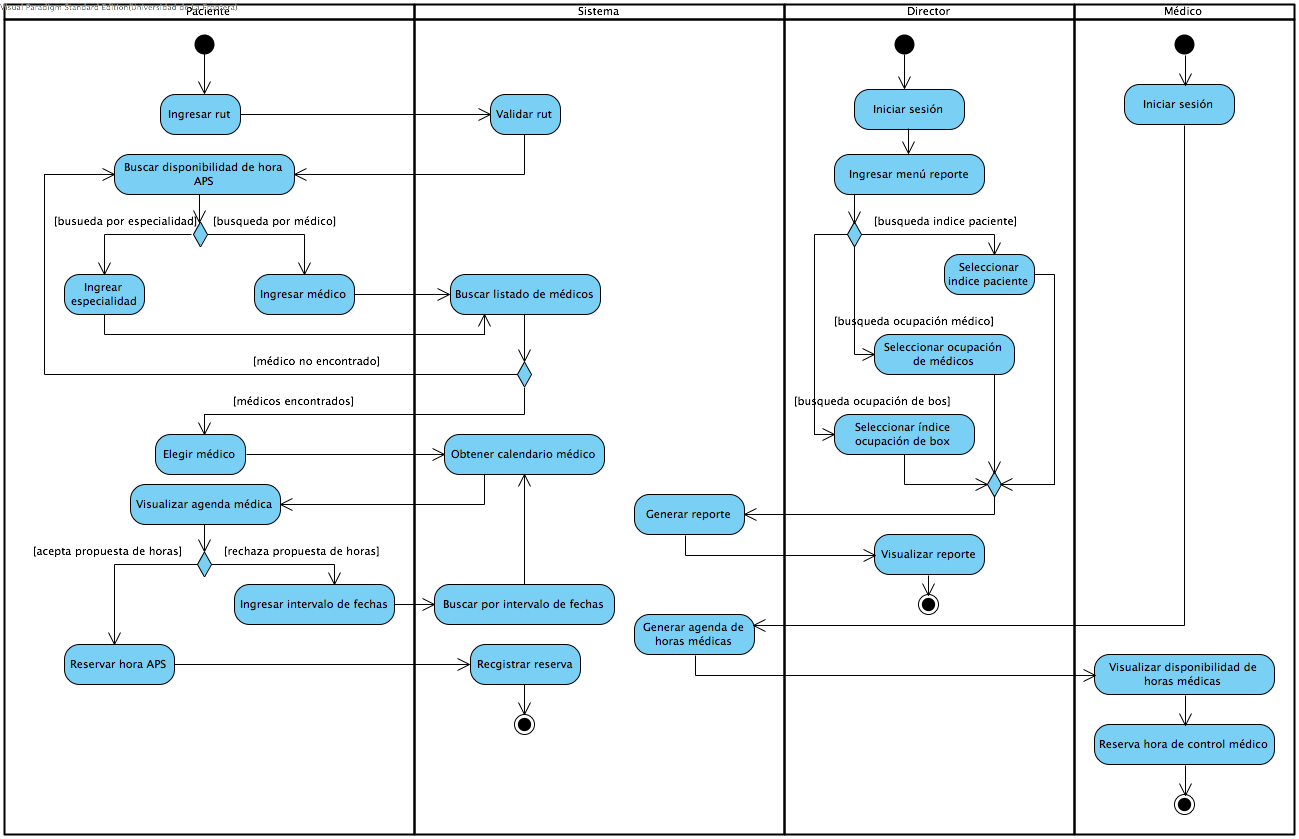




**Diagrama de Actividades y Detalle de cada proceso:** Este tipo de diagrama le resulta familiar a la mayoría de programadores, ya que el diagrama de flujo es un esquema ampliamente conocido y estos muestran una secuencia de pasos, procesos, puntos de decisión y bifurcaciones.



**Ejemplo Referencial:**



**Descripción Detallada de las Actividades**: de ser necesario alguna aclaratoria, detallar en esta sección, particularmente, si aplica, asociar las actividades a las reglas de negocio.

# Análisis Volumétrico

En este apartado se deben detallar temas tales como volúmenes de transacciones, volúmenes de datos, número de usuarios conectados al mismo tiempo, en horario pico, etc., cuando se estén modificando o agregando estructuras de datos. Se debe contar con el volumen de transacciones de negocio especificadas por el usuario para un período determinado y estimaciones a 6 meses, 1 año e incluso 3 años, por cada capa. Estas estimaciones son tanto transacciones de negocio como transacciones de las diferentes capas impactadas.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nombre Tabla** | **Largo Registro** | **N° Inicial Registros** | **N° Registros Incremento Semestral** | **N° Registros Incremento Anual** | **Largo Índice Mayor** | **N° Índices Tabla** |
|
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

# Infraestructura Requerida

En este apartado se deben detallar los requerimientos de hardware, software, personas, redes y comunicaciones. De ser necesarias estimaciones de Recursos Computacionales Críticos, éstas deben estar relacionadas a la complejidad de la petición a desarrollar y deben quedar documentadas.

**Ejemplo:**

**Software**

|  |  |  |  |
| --- | --- | --- | --- |
| Ambiente | **Desarrollo** | **Calidad** | **Producción** |
| **Aplicación** | | | |
| Lenguaje |  |  |  |
| Versión |  |  |  |
| **Aplicación** | | | |
| Manejador |  |  |  |
| Versión |  |  |  |
| **Aplicación** | | | |
| Manejador |  |  |  |
| Versión |  |  |  |
| **Aplicación** | | | |
| Manejador |  |  |  |
| Versión |  |  |  |

**Hardware**

|  |  |  |  |
| --- | --- | --- | --- |
| **Servidor de Aplicación** | | | |
| **Ambiente** | **Desarrollo** | **Calidad** | **Producción** |
| Rol (rol del servicio o aplicativo) |  |  |  |
| Dirección IP |  |  |  |
| HostName |  |  |  |
| Memoria RAM |  |  |  |
| Capacidad en disco |  |  |  |
| Modelo |  |  |  |
| Procesador |  |  |  |
| Cantidad de procesadores Físicos |  |  |  |
| Core por Socket Físico |  |  |  |
| Sistema Operativo |  |  |  |
| Service Pack |  |  |  |
| Edición del Sistema Operativo (Standard, Enterprise) |  |  |  |
| Arquitectura (x32/x64) |  |  |  |
| Dominio |  |  |  |
| Alta disponibilidad (S/N) - Tipo (Cluster o NLB Microsoft, WebSphere, etc) |  |  |  |
| Alta disponibilidad (S/N) |  |  |  |
| Físico o Virtualizado |  |  |  |
| Tipo de arreglos de disco recomendado (RAID) |  |  |  |
| Definición de tamaño de volúmenes de disco |  |  |  |
| Rutas a excluir en el chequeo del antivirus (Si aplica) |  |  |  |
| Hotfixs de Microsoft no recomendados a instalar (Si aplica) |  |  |  |
| Componentes de Sistema Operativo a instalar (IIS (WWW, FTP, etc), MS Queue, etc) |  |  |  |
| Derechos y/o privilegios de las cuentas de servicio (debe estar sustentado) |  |  |  |
| Rutas o carpetas con permisos específicos (NTFS /Share) |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Servidor de Base de Datos** | | | |
| **Ambiente** | **Desarrollo** | **Calidad** | **Producción** |
| Dirección IP |  |  |  |
| HostName |  |  |  |
| Server de BD con o sin  Instancia |  |  |  |
| Memoria RAM |  |  |  |
| Cantidad y Capacidad en disco |  |  |  |
| Modelo |  |  |  |
| Cantidad y capacidad de Procesadores |  |  |  |
| Cantidad de Núcleos del Procesador |  |  |  |
| Sistema Operativo |  |  |  |
| Service Pack |  |  |  |
| Dominio |  |  |  |
| Alta disponibilidad (S/N) |  |  |  |
| Físico o Virtualizado |  |  |  |
| Clúster (S/N) |  |  |  |

|  |  |
| --- | --- |
| **Características de las estaciones clientes** | |
| Memoria RAM |  |
| Capacidad en disco |  |
| Procesador |  |
| Sistema Operativo |  |
| Service Pack |  |
| Navegador |  |

# Seguridad y Auditoria

En este apartado se debe detallar la solución a los requerimientos de seguridad, de auditoría y perfilamiento (nombre de cada perfil, características de acceso a cada funcionalidad y condiciones de seguridad, logs y trazas).

**La seguridad deberá implementarse con JWT o OAUTH2, solo existe un rol y es el de usuario**

# Manejo De Datos Externos

En este apartado se deben detallar los requerimientos asociados a importación o exportación de datos, conversión y retención de los mismo.

**N/A**

# Respaldos, Retención Y Recuperación De Información, Logs Y Trazas.

En este apartado se deben considerar aquellos requerimientos de respaldo de información y se debe especificar los siguientes tópicos: Descripción, permanencia on-line, permanencia off-line, Descripción proceso de limpieza, Descripción proceso de recuperación.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **Nombre Base de datos** | **Crítica** | **Alta Transaccionalidad** | **Data Confidencial** | |  | SI/NO | SI/NO | SI/NO | |  | SI/NO | SI/NO | SI/NO | |  |

# Proceso de Validación de Resultados

En este apartado se deben detallar las pantallas, reportes e informes que permite al usuario validar los resultados de la funcionalidad, especificando Descripción, Tipo de Control (Integridad, Calidad, Seguridad), Modalidad (Manual o Automática), Frecuencia (Diario, Por evento, A pedido), Logs (Integridad, Condiciones de Culminación), Acciones (Descripción de las acciones a seguir en el evento de encontrar una falla).

# Tablas Maestras (Sólo para Core Bancario).

En este apartado se deben detallar las tablas maestras que han sido modificadas o agregadas por requerimiento del proyecto.

# Plan De Despliegue Y Requisitos De Implantación.

En este apartado se debe detallar las actividades que se ejecutaran para el despliegue y la implantación del proyecto o requerimiento en los ambientes de calidad y producción, aspectos relevantes para el funcionamiento del aplicativo y que es necesario documentarlo para futuros mantenimientos. Es importante contemplar igualmente el pase al centro Alterno en caso de aplicar.

# Estructura En Aldon.

En este apartado se deben detallar las rutas en donde se encuentra la aplicación para los pases entre los ambientes de Desarrollo / Calidad y Producción.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Grupo** | **Aplicativo** | **Release** | **Collections** | **Nombre Tarea** |
| Location |  |  |  |  |  |
| Collections |  |  |  |  |  |

# Roles Y Responsabilidades De La Solución.

En este apartado se deben detallar en el cuadro anexo todos los roles y las responsabilidades de los distintos departamentos de TI que deben atender la solución en sus diferentes ambientes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Aplicaciones de la Solución** | **Actividades Macro** | **Detalles Actividades** | **Responsables** |
| **Aplicación 1** | Configuración |  |  |
| Despliegues |  |  |
| Desarrollo |  |  |
| Monitoreo |  |  |
| Seguridad |  |  |
| **Aplicación 2** | Configuración |  |  |
| Despliegues |  |  |
| Desarrollo |  |  |
| Monitoreo |  |  |
| Seguridad |  |  |

# Glosario De Términos.

**Definiciones, Acrónimos y Abreviaturas**

Esta sub-sección proporciona las definiciones de todos los términos, siglas, y abreviaturas requeridas para interpretar correctamente la Realización del Requerimiento. Esta información se puede proporcionar por la referencia al Glosario del Proyecto.

**Ejemplo:**

* **DRR Padre:** Documento de Realización de Requerimiento que contiene el diseño de todas las capas distintas a la capa de aplicación.
* **Chat Bot:** Un chatbot es un programa informático con el que es posible mantener una conversación, tanto si queremos pedirle algún tipo de información o que lleve a cabo una acción.
* **AIML:** El AIML, o Artificial Intelligence Mark-up Language es un lenguaje de programación basado en XML.
* **Angular:** framework para aplicaciones web desarrollado en TypeScript, de código abierto, que se utiliza para crear y mantener aplicaciones web de una sola página.
* **HTML5:** (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML.
* **Front-end y Back-end:** son términos que se refieren a la separación de intereses entre una capa de presentación y una capa de acceso a datos, respectivamente.

# Exclusiones

**Licenciamiento**

* Las licencias de software son de responsabilidad exclusiva de **EL CLIENTE**

**Plataforma**

* No se contempla la instalación, ni configuración de ningún entorno de despliegue de productos que no estén descritos en el alcance del proyecto.

**Servidores Físicos**

* No se incluye ningún tipo de hardware requerido, estos deben ser suministrados por **EL CLIENTE**

**Servidores Virtuales**

* No se contempla la configuración de ningún tipo software para virtualización.

**Sistemas internos**

* El alcance del presente proyecto no contempla modificación de las aplicaciones de **EL CLIENTE,** salvo las que se acuerden por escrito entre las partes.

**Configuraciones adicionales**

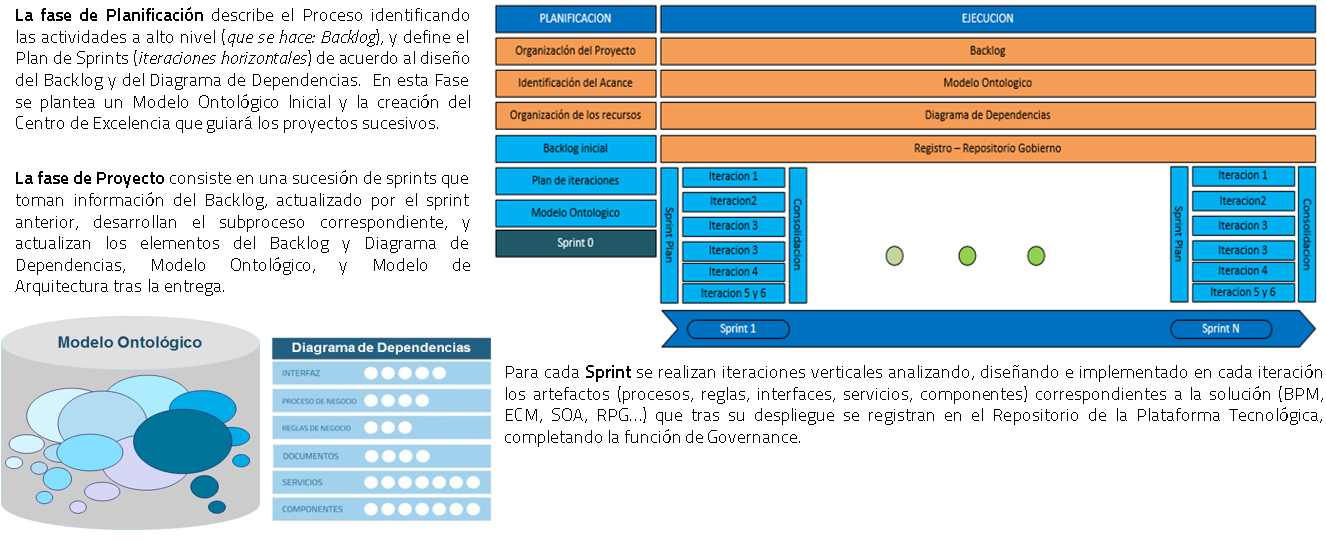
* SOAINT no se hará cargo de ningún tipo de configuración como redes, bases de datos, entre otros que no estén contemplados en el alcance.
* No están incluidas en el alcance de esta propuesta actividades relacionadas con la instalación de acometidas eléctricas, cableado de comunicaciones y/o telefónico, obras civiles, diseño o procura de aplicaciones de sistemas operativos y/o cualquier SW en general, reparación de HW y/o SW que no esté explícitamente especificada en el alcance detallado del proyecto.
* No se incluye ni garantiza ningún tipo de respaldo de datos, ya que es de responsabilidad exclusiva de **EL CLIENTE.**

# Metodología De la Fabrica: SCRUM - IterAE©

Metodología Iterativa para la construcción de Soluciones de Negocio sobre Tecnologías de la Información mediante artefactos orientados a la Arquitectura Empresarial; utilizando metodologías AGILES basados en SCRUM y con el siguiente marco:



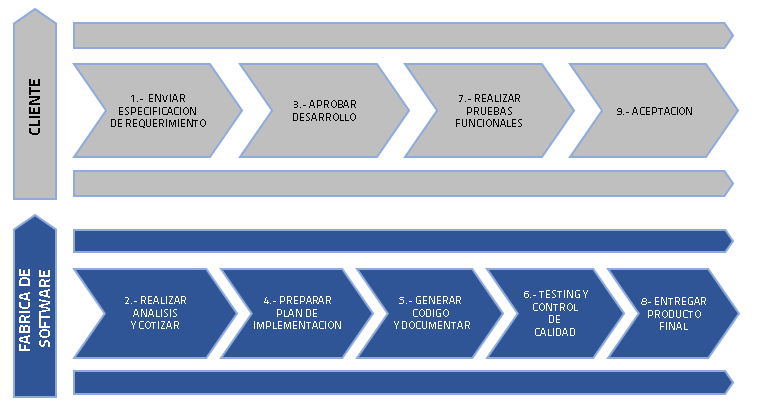
**Modelo Funcional de la Metodología SCRUM – ITERAE**



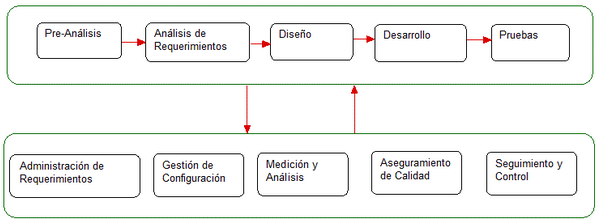
# ANEXO – Workflow Fabrica

El proceso de desarrollo de software propuesto está basado en el ciclo de vida Incremental, ya que combina elementos del modelo en cascada aplicado en forma iterativa y aplica secuencias lineales de manera escalonada conforme avanza el tiempo.

Además, se basa en el comportamiento del modelo espiral, el cual realiza el análisis de los riesgos al final de cada iteración de desarrollo, y con base en el resultado de este análisis se realiza el plan de cada incremento. Este proceso se repite después de la entrega de cada incremento (iteraciones de desarrollo) hasta que se elabore el producto completo. Los procesos definidos aplican para proyectos de desarrollo de software a la medida, de complejidad media, cuya duración oscila entre 3 y 6 meses. Para la ejecución de los procesos se elaboraron una serie documentos, diagramas, formatos, instructivos y actas que permiten el registro de las actividades de cada proceso.



**DESGLOCE DE PROCESOS**



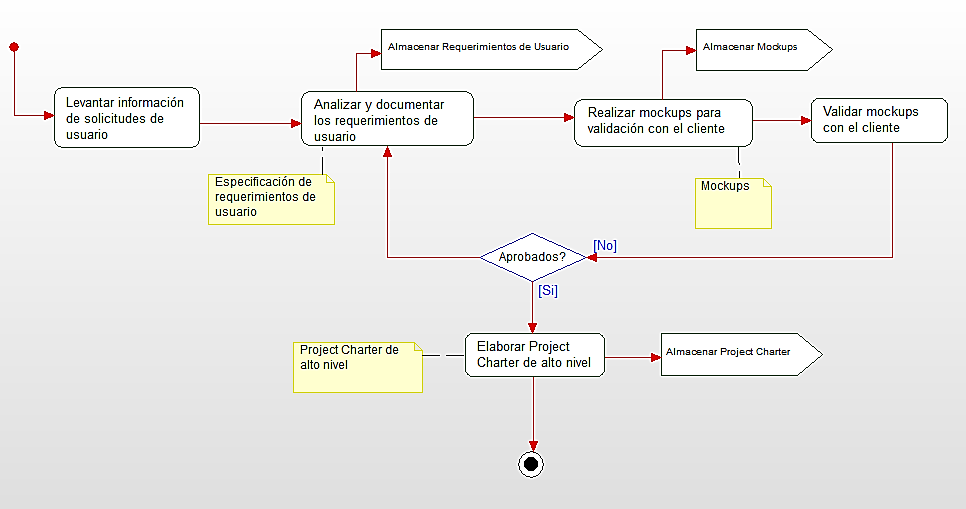
# 19.1 Procesos De Desarrollo De La Fabrica

**PRE-ANALISIS**

El proceso de **pre – análisis** define los lineamientos y pautas para levantar, analizar y documentar las solicitudes de usuario y realizar los documentos suficientes para definir y validar con el cliente el alcance del proyecto o requerimiento.

La base para la definición del proceso de pre análisis es el proceso ***Pre-Desarrollo / Exploración del concepto*** del modelo ***IEEE 1074***, que se enfoca en identificar y analizar las necesidades del cliente hasta entenderlas y refinarlas, orientando y garantizando la correcta ejecución de las actividades, para obtener la información suficiente para la ejecución de las fases posteriores del proyecto. Además, se tomaron algunas de las prácticas del área de procesos ***Requirements Development (RD)*** del modelo ***CMMI Development versión 1.3*** (Capability Maturity Model Integration), que hacen referencia de las necesidades del cliente y a la transformación de dichas necesidades en requerimientos de usuario.

El siguiente diagrama muestra el flujo de información entre las actividades del proceso de pre - análisis, incluyendo los productos de trabajo generados durante el proceso.

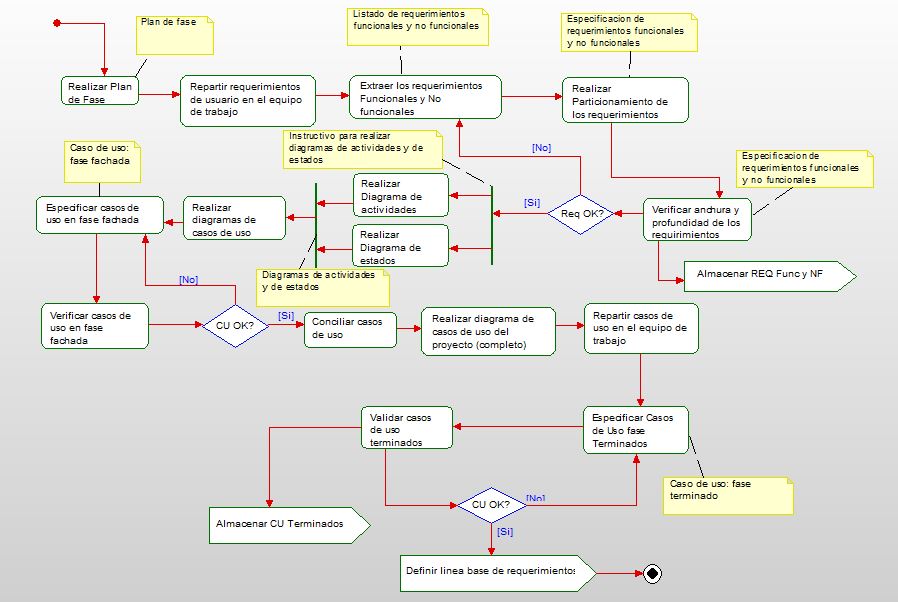


**ANÁLISIS DE REQUERIMIENTOS**

El proceso de análisis de requerimientos refina, modela, especifica y verifica las solicitudes de usuario, y con ello genera documentos base para la ejecución de los procesos siguientes.

La base para la definición del proceso de administración de requerimientos es el área de procesos Requirements Development (RD) del modelo CMMI Development versión 1.3 (Capability Maturity Model Integration), que reúne un conjunto de prácticas que orientan y garantizan el correcto análisis de los requerimientos de un sistema, en un proyecto de desarrollo de software.

El diagrama muestra el flujo de información entre las actividades del proceso de análisis de requerimientos, incluyendo los productos de trabajo generados durante el proceso.

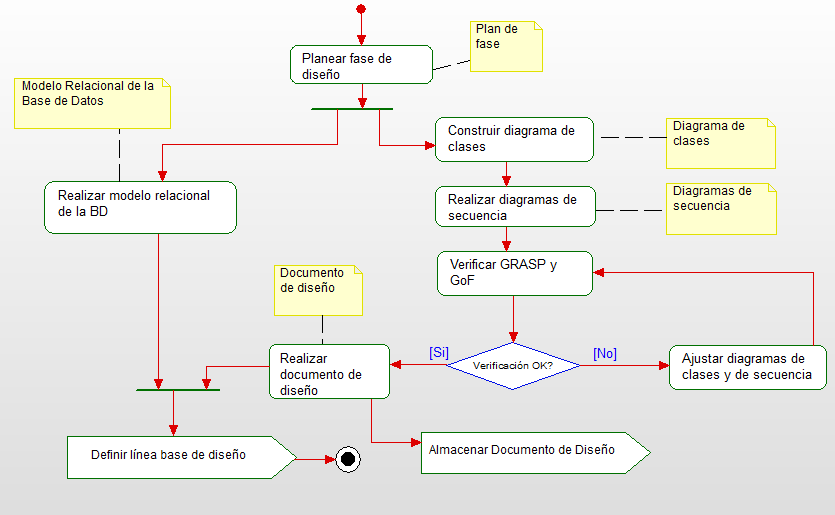


**DISEÑO**

El proceso de diseño permite la definición de los componentes de software que deben ser implementados para satisfacer las necesidades de los usuarios. Tiene en cuenta los requerimientos funcionales y no funcionales definidos y acordados con el cliente.

La base para la definición del proceso de diseño es el área de procesos Technical Solution (TS) del modelo CMMI Development versión 1.3 (Capability Maturity Model Integration), que define un conjunto de prácticas que permiten diseñar el producto y los componentes de producto adecuadamente.

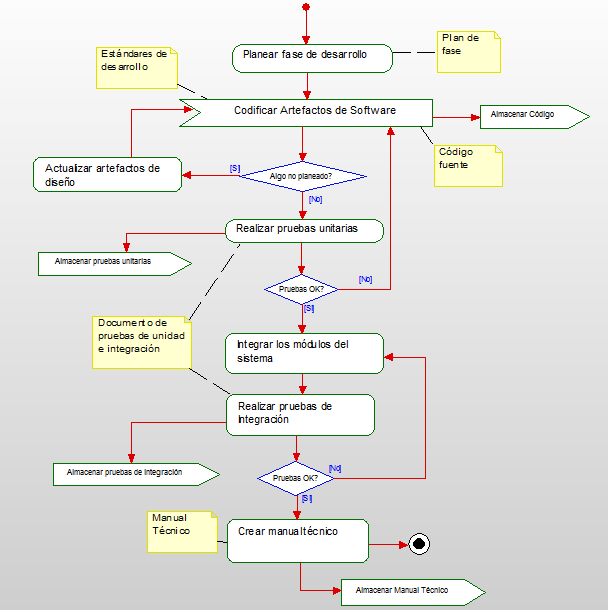
El diagrama muestra el flujo de información entre las actividades del proceso de diseño, incluyendo los productos de trabajo generados durante el proceso.



**DESARROLLO**

El proceso de desarrollo proporciona los lineamientos para la adecuada generación del código y las verificaciones necesarias para minimizar los errores técnicos inherentes a los procesos de desarrollo de software.

La base para la definición del proceso de desarrollo es el área de procesos Technical Solution (TS) del modelo CMMI Development versión 1.3 (Capability Maturity Model Integration), que define un conjunto de prácticas que permiten realizar la construcción del software, partiendo del diseño realizado. El diagrama muestra el flujo de información entre las actividades del proceso de desarrollo, incluyendo los productos de trabajo generados durante el proceso.

****

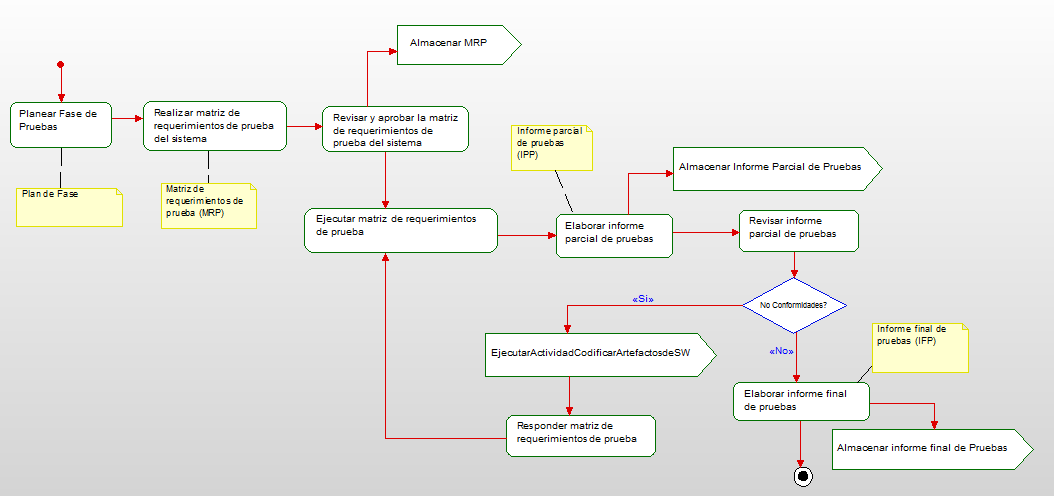
**PRUEBAS**

El proceso de Pruebas garantiza que el producto final cumpla con los requerimientos establecidos por el cliente.

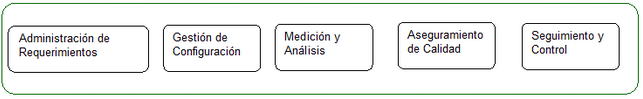
La base para la definición del proceso de Pruebas es el área de procesos Verification (VER) del modelo CMMI Development versión 1.3 (Capability Maturity Model Integration), que reúne un conjunto de prácticas que orientan y garantizan la correcta ejecución de las pruebas durante la ejecución de un proyecto de desarrollo de software.

El resultado final del proceso de Pruebas es un producto que cumple con los parámetros de calidad definidos en el proyecto.

El diagrama muestra el flujo de información entre las actividades del proceso de pruebas, incluyendo los productos de trabajo generados durante el proceso.



# 19.2 Procesos Transversales De La Fabrica

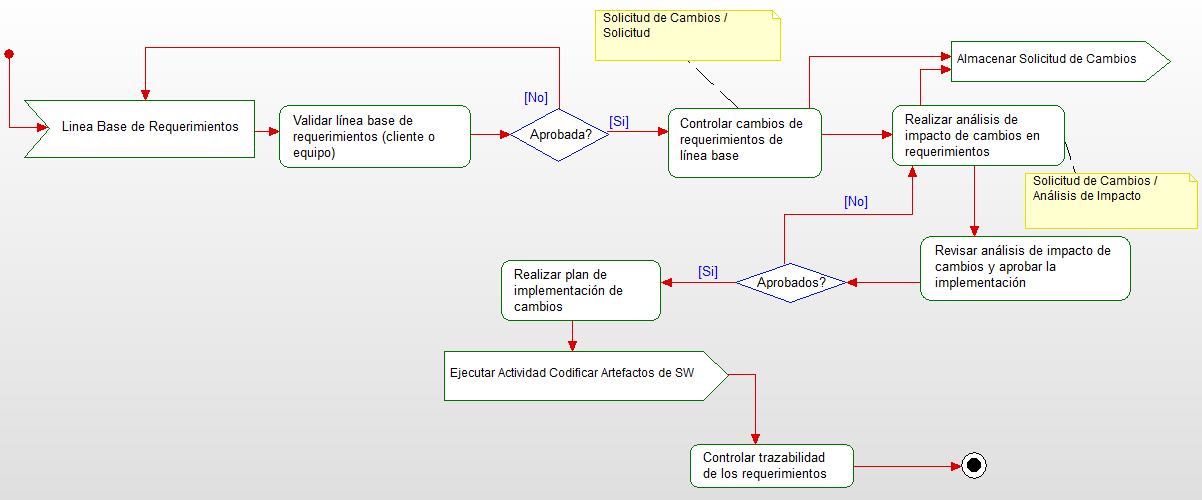


**ADMINISTRACIÓN DE REQUERIMIENTOS**

El proceso de administración de requerimientos define los lineamientos y pautas para administrar los requerimientos de un sistema de forma simple y consistente, para garantizar mantener la trazabilidad y controlar los cambios de los mismos durante la ejecución del ciclo de vida de desarrollo del software.

La base para la definición del proceso de administración de requerimientos es el área de procesos Requirements Management (REQM) del modelo CMMI Development versión 1.3 (Capability Maturity Model Integration), que reúne un conjunto de prácticas que orientan y garantizan la correcta administración de los requerimientos durante la ejecución de un proyecto de desarrollo de software.

El diagrama muestra el flujo de información entre las actividades del proceso de administración de requerimientos, incluyendo los productos de trabajo generados durante el proceso.

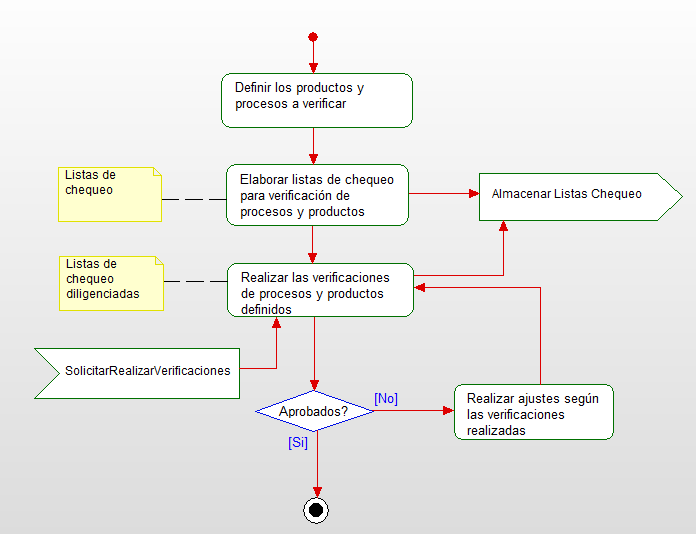


**ASEGURAMIENTO DE CALIDAD**

El proceso de aseguramiento de calidad garantiza que los procesos y los productos de trabajo del proyecto, se realicen según los estándares definidos, contribuyendo a la correcta ejecución del proyecto.

La base para la definición del proceso de aseguramiento de calidad es el área de procesos Process and Product Quality Assurance(PPQA) del modelo CMMI Development versión 1.3 (Capability Maturity Model Integration), que reúne un conjunto de prácticas que orientan y garantizan la calidad en proyectos de desarrollo de software.

El diagrama muestra el flujo de información entre las actividades del proceso de aseguramiento de calidad, incluyendo los productos de trabajo generados durante el proceso.



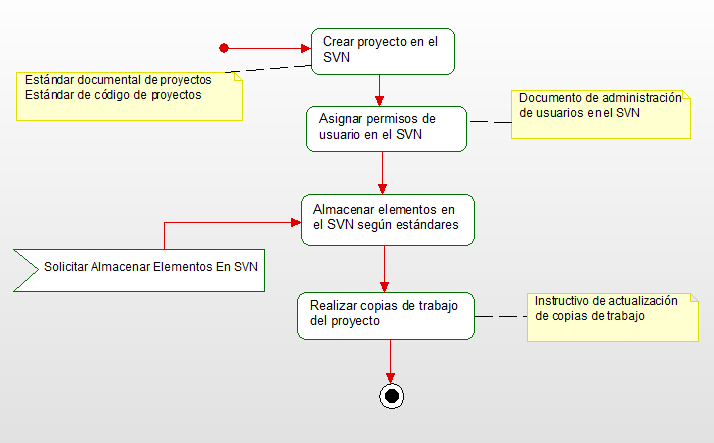
**GESTIÓN DE LA CONFIGURACIÓN**

El proceso de gestión de configuración define los lineamientos y pautas para administrar la configuración del proyecto de forma simple, de tal manera que se conserve la integridad en los productos de trabajo durante la ejecución del proyecto.

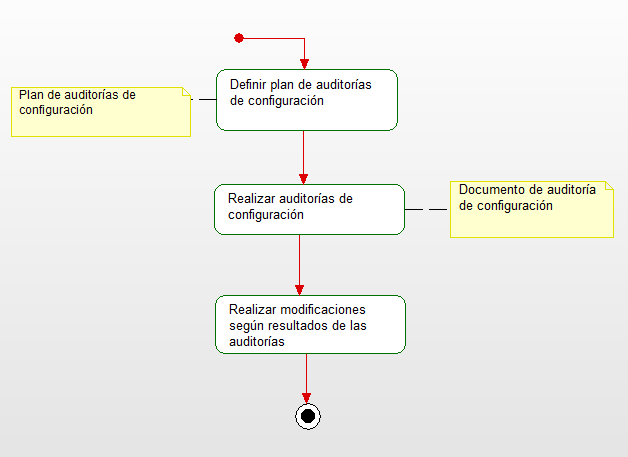
La base para la definición del proceso de gestión de configuración es el área de procesos Configuration Management (CM) del modelo CMMI Development versión 1.3 (Capability Maturity Model Integration), que reúne un conjunto de prácticas que orientan y garantizan la correcta gestión de la configuración del proyecto durante la ejecución de un proyecto de desarrollo de software.

Los siguientes diagramas muestran el flujo de información entre las actividades de los subprocesos del proceso de gestión de configuración, incluyendo los productos de trabajo generados durante los subprocesos.

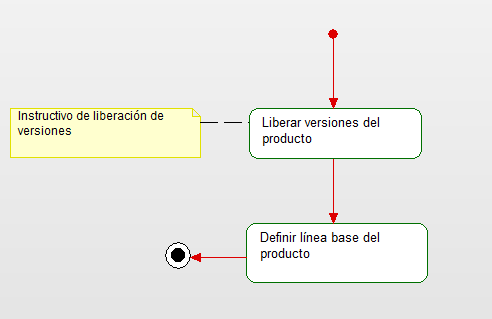
**Subproceso de Administración de la configuración del Proyecto o Requerimiento:**



Subproceso de Administración de auditorías de configuración del proyecto:



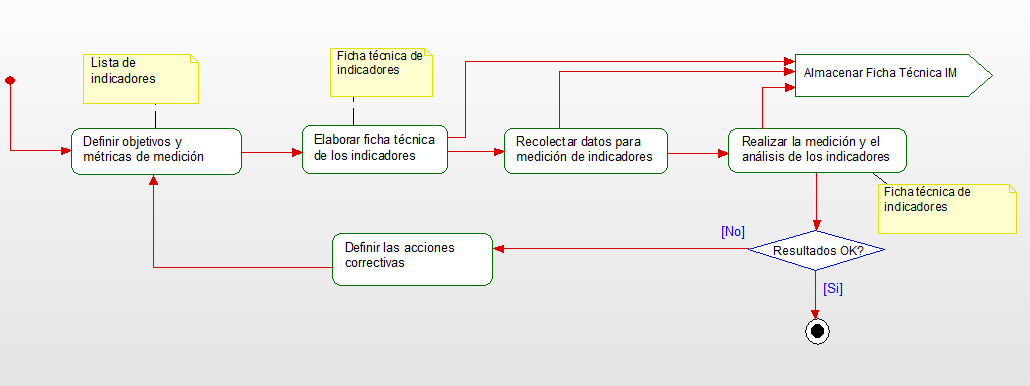
Subproceso de Liberación de versiones del producto:



**MEDICIÓN Y ANÁLISIS**

El proceso de medición y análisis define los indicadores de medición que se implementarán en los proyectos de desarrollo de software, los cuales permitirán medir el grado de adaptación del proyecto a los procesos definidos para su ejecución y tomar las acciones correctivas necesarias para minimizar los errores y garantizar la calidad del producto final.

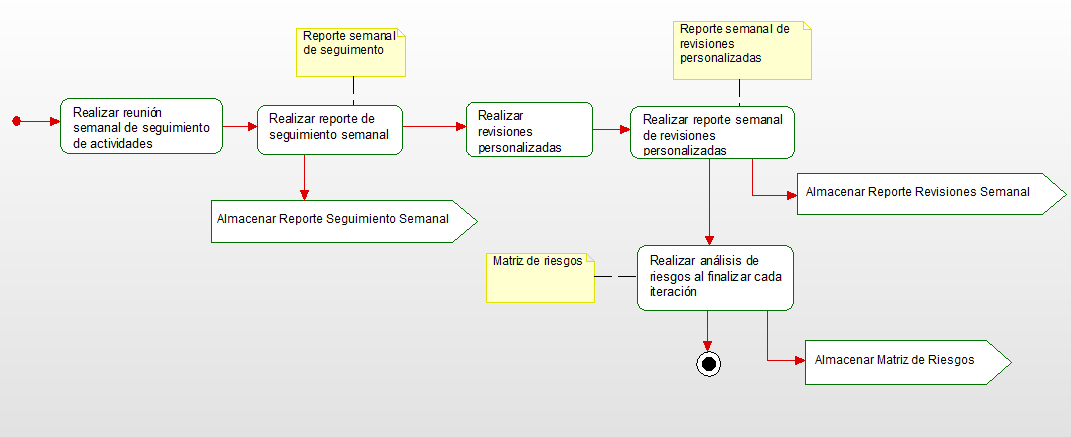
La base para la definición del proceso de medición y análisis es el área de procesos Measurement and Analysis (MA) del modelo CMMI Development versión 1.3 (Capability Maturity Model Integration), que reúne un conjunto de prácticas que orientan y garantizan la correcta definición, administración y aplicación de los indicadores de medición durante la ejecución de un proyecto de desarrollo de software.



**SEGUIMIENTO Y CONTROL**

El proceso de seguimiento y control define los mecanismos de seguimiento con los cuales se controlarán las actividades del proyecto, para garantizar su ejecución dentro del alcance, tiempo y costo definidos.

La base para la definición del proceso de seguimiento y control es el área de procesos Project Monitoring and Control (PMC) del modelo CMMI Development versión 1.3 (Capability Maturity Model Integration), que reúne un conjunto de prácticas que orientan y garantizan la correcta definición y ejecución de las actividades de seguimiento durante la ejecución de un proyecto de desarrollo de software.



ANEXOS:

PRODUCTOS Y SERVICIOS (Solicitar a mercadeo)